

# МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

## Сумський державний університет

Центр заочної, дистанційної та вечірньої форм навчання

Кафедра комп'ютерних наук

«До захисту допущено»

В.о. завідувача кафедри

Ігор ШЕЛЕХОВ

(підпис)

25 січня 2024 р.

## КВАЛІФІКАЦІЙНА РОБОТА

### на здобуття освітнього ступеня магістр

зі спеціальності 122 – Комп'ютерних наук,

освітньо-професійної програми «Інформатика»

на тему: «Інформаційна технологія проєктування інтерактивних практичних завдань по теорії числових систем»

здобувачки групи ІН.мз-22с Таранець Зої Володимирівни

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

Зоя ТАРАНЕЦЬ

(підпис)

Керівник

старший викладач,

кандидат фізико-математичних наук

Оксана ШОВКОПЛЯС

(підпис)

Суми – 2024

**Сумський державний університет**  
Центр заочної, дистанційної та вечірньої форм навчання  
Кафедра комп'ютерних наук

«Затверджую»

В.о. завідувача кафедри

Ігор ШЕЛЕХОВ

(підпис)

## ІНДИВІДУАЛЬНЕ ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

### на здобуття освітнього ступеня магістра

зі спеціальності 122 - Комп'ютерних наук, освітньо-професійної програми «Інформатика»  
здобувачки групи ІН.мз-22с Таранець Зої Володимирівни

1. Тема роботи: «Інформаційна технологія проектування інтерактивних практичних завдань по теорії числових систем»

затверджую наказом по СумДУ від «20» листопада 2023 р. № 1307-VI

2. Термін здачі здобувачем кваліфікаційної роботи до 25 січня 2024 року

3. Вхідні дані до кваліфікаційної роботи \_\_\_\_\_

4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)

1) Аналіз предметної області та актуальності розробки, постановка задачі дослідження. 2) Огляд та вибір програмних засобів для досягнення мети. 3) Розробка інформаційної моделі. 4) Розроблення інтерактивного веб-тренажеру. 5) Аналіз отриманих результатів.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) \_\_\_\_\_

6. Консультанти до проекту (роботи), із значенням розділів проекту, що стосується їх

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання «03» жовтня 2023 р.

Завдання прийняв до виконання \_\_\_\_\_

(підпис)

Керівник \_\_\_\_\_

(підпис)

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Термін виконання	Примітка
1	<i>Аналіз предметної області та актуальності розробки, постановка задачі дослідження</i>	25.09-30.10.23	Виконано
2	<i>Огляд та вибір програмних засобів для досягнення мети</i>	31.10-09.11.23	Виконано
3	<i>Розроблення інформаційної моделі</i>	10.11-25.11.23	Виконано
4	<i>Розроблення інтерактивного веб-тренажеру</i>	26.11-24.12.23	Виконано
5	<i>Аналіз отриманих результатів</i>	25.12.23-07.01.24	Виконано
6	<i>Оформлення пояснювальної записки до кваліфікаційної роботи</i>	08.01-10.01.24	Виконано

Здобувач вищої освіти \_\_\_\_\_

(підпис)

Керівник \_\_\_\_\_

(підпис)

## АНОТАЦІЯ

**Записка:** 104 стор., 39 рис., 1 табл., 1 додаток, 44 використаних джерела.

**Обґрунтування актуальності теми роботи** – тема кваліфікаційної роботи є актуальною та затребуваною, оскільки присвячена вирішенню серйозної практичної задачі, а саме впровадженню інформаційної технології проектування інтерактивних практичних завдань по теорії числових систем для удосконалення та підвищення ефективності освітнього процесу, а також спрощення отримання практичних навичок студентами у дистанційному форматі навчання.

**Об’єкт дослідження** – процес проектування та розроблення інформаційного програмного забезпечення інтерактивного тренажера по теорії числових систем.

**Предмет дослідження** – методи, підходи та технології, які використовуються під час створення інтерактивного віртуального тренажера.

**Мета роботи** – створення інформаційної технології проектування інтерактивного тренажера по темі «Системи числення. Перетворення чисел з однієї системи числення в іншу».

**Методи дослідження** – інформаційний аналіз, технологія розробки віртуальних тренажерів, мова програмування JavaScript.

**Результати** – розроблено інформаційну технологію проектування інтерактивних практичних завдань по теорії числових систем. Реалізовано інтерактивний тренажер, який відповідає поставленим завданням та проведено його тестування.

ІНТЕРАКТИВНИЙ ТРЕНАЖЕР, ДИСТАНЦІЙНА ОСВІТА, СИСТЕМА  
ЧИСЛЕННЯ, ФРЕЙМВОРК JTRAINER, JAVASCRIPT

## ЗМІСТ

ВСТУП .....	5
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ .....	7
1.1 Дослідження актуальності проблеми .....	7
1.2 Аналіз існуючих веб-додатків для інтерактивного навчання.....	12
1.3 Постановка задачі .....	18
2 МЕТОДОЛОГІЯ ВИРІШЕННЯ ЗАВДАННЯ.....	20
2.1 Середовище розробки .....	20
2.2 Інструменти графічного дизайну .....	22
2.3 Вибір локального сервера .....	24
2.4 Вибір програмних засобів.....	27
3 ПРОЄКТУВАННЯ СИСТЕМИ .....	33
3.1 Структурно-функціональна модель .....	33
3.2 UML моделювання системи.....	35
4 ПРАКТИЧНА РЕАЛІЗАЦІЯ.....	40
4.1 Існуючі обмеження системи та перелік функціональних вимог .....	40
4.2 Структура тренажера та розроблення окремих компонентів .....	41
4.3 Програмна реалізація веб-тренажера .....	45
4.4 Результати розроблення та тестування додатку.....	53
ВИСНОВКИ.....	64
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	65
ДОДАТОК А .....	71

## ВСТУП

**Актуальність.** У сучасному світі розвиток інформаційних технологій охопив всі сфери людського життя, в тому числі й освітню галузь. Події, які відбуваються в Україні останні 3-4 роки, тільки пришвидшили впровадження і бурхливий розвиток системи дистанційного навчання. Більшість закладів освіти перейшли від досліджень сучасних підходів навчання та викладання до стрімкого їх впровадження у освітній процес [1]. Такі трансформації відбуваються не тільки в нашій країні, а й у більшості країн світу.

Завдяки системі віддаленого навчання студенти мають багато переваг у процесі здобуття освіти. По-перше, вони не мають прив'язки до місця проведення занять. Це дозволяє економити час, а також навчатися з будь-якого куточка світу. По-друге, студент може довільно обирати для себе найзручніший момент для навчання, не прив'язуючись до розкладу викладача та інше. Для полегшення комунікації між викладачами та студентами у сучасному світі існує маса доступних телекомунікаційних сервісів. Наприклад: навчальні платформи, електронні журнали, чати, електронна пошта, засоби для проведення веб-конференцій, месенджери тощо.

Навчальні заклади інтенсивно впроваджують дистанційне навчання та з метою вдосконалення знань студентів пропонують великий вибір віртуальних навчальних матеріалів. До них можна віднести відео лекції, лабораторні та практичні роботи, тестові завдання, демонстрації, а також інтерактивні віртуальні тренажери, що розміщуються на онлайн ресурсах навчальних закладів. Досить великий внесок в полегшення вивчення навчальних матеріалів та опанування важливих навичок в процесі вирішення практичних завдань здійснюється за допомогою інтерактивних віртуальних тренажерів.

Дана робота присвячена процесу розробки інформаційної технології проєктування інтерактивних практичних завдань по теорії числових систем як важливого елементу навчальних матеріалів дистанційної освіти.

**Об'єктом дослідження** кваліфікаційної роботи є процес проектування та розробки інформаційного та програмного забезпечення веб-тренажера по теорії числових систем та інтегрування його до єдиної навчальної платформи Сумського державного університету.

**Предметом дослідження** є аналіз та вивчення методів, підходів, способів та технологій, які використовуються у процесі створення інтерактивного віртуального тренажера.

**Мета** кваліфікаційної роботи магістра полягає у створенні інформаційної технології проектування інтерактивного додатку для удосконалення та полегшення навчального процесу в опануванні правил перетворення чисел, а також для підвищення засвоєння навчальних матеріалів та отримання практичних навичок студентами при розв'язанні задач в процесі вивчення теми «Системи числення. Перетворення чисел з однієї системи числення в іншу».

**Зв'язок роботи з науковою темою.** Кваліфікаційна робота магістра виконана на кафедрі комп'ютерних наук Сумського державного університету та пов'язана з виконанням науково-дослідної роботи № 0120U103407 «Застосування технологій games learning, blended learning, віртуальної та доповненої реальності в навчальному процесі» (2020-2025).

**Гіпотеза.** Результатом проекту буде інтерактивний віртуальний тренажер, впроваджений в навчальний процес, який дасть змогу здобувачам освіти краще засвоювати отримані теоретичні знання, здобувати практичні навички та підвищувати якість та ефективність освітнього процесу завдяки росту рівня мотивації та зацікавленості студентів.

Дана робота має наступну **структуру**: вступ, аналіз предметної області, аналіз аналогічних рішень, постановка задачі дослідження, аналітична частина щодо вибору методики та інструментів рішення існуючої проблеми, опис програмного забезпечення інформаційної системи, результати розробки, висновки, список використаних джерел та додаток.

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

## 1.1 Дослідження актуальності проблеми

### 1.1.1 Роль інформаційно-комунікаційних технологій в освітньому процесі

Першим документально підтвердженим випадком масового дистанційного навчання у 1728 році стали заочні курси стенографії через газету Boston Gazette, організовані американським професором Калемом Філіпсом (англ. Caleb Phillips). Хоча історія дистанційного навчання налічує майже 300 років, найбільший прорив у цій області стався з моменту винайдення та масового розповсюдження електронно-обчислювальних машин.

Виключно потужний імпульс у розвитку освітніх технологій розпочався з появою комп'ютерних мереж та персональних комп'ютерів і пов'язаний з використанням глобальної мережі Інтернет. Використання спільних та розподілених ресурсів, веб-технологій, віддалений доступ до навчальних матеріалів забезпечив суттєве підвищення ефективності професійної підготовки, її доступності та масовості. Мережні технології, висока якість та підвищення апаратного забезпечення уможливили створення професійних середовищ та систем для надання освітніх послуг і реалізації різних видів формальної та неформальної освіти. Ключовими термінами цього періоду є Інтернет, веб-курси, гіпертекст, віртуальне навчання, віртуальний університет, неперервна освіта, навчання протягом усього життя, дистанційне навчання, електронне навчання та мобільне навчання [2].

Саме з даного часу університети розвинених країн світу почали активно застосовувати електронне навчання. А в останнє десятиріччя двадцятого століття дистанційне навчання стало невід'ємною частиною освітнього процесу і підвищення кваліфікації різноманітних спеціалістів [3]. Однозначно це було пов'язано з активним розвитком цифрових технологій.

Питання забезпечення належного рівня інформаційного обслуговування навчального процесу щороку стає все більш актуальним. Вплив інформаційно-комунікаційних технологій на систему освіти відбувається постійно й безперервно.

Інформаційно-комунікаційні технології (ІКТ) – це сукупність методів, засобів і прийомів, що використовуються для добору, опрацювання, зберігання, подання, передавання різноманітних даних і матеріалів, необхідних для підвищення ефективності різних видів діяльності [4].

ІКТ – це технології опрацювання інформації за допомогою комп'ютера та телекомунікаційних засобів. Впровадження ІКТ в навчальний процес стимулює інтерес до навчальної діяльності, сприяє формуванню логічного та творчого мислення, сприяє розвитку студентів та формуванню інформаційної культури.

В освіті активно використовуються ІКТ для пошуку та отримання додаткової інформації; розширення та поглиблення знань з використанням системи Інтернет, задоволення необхідних потреб викладачів та студентів; формування та закріплення навичок, прийомів, способів, умінь їх застосування[5, 6].

На думку більшості дослідників, основним педагогічним завданням використання ІКТ у навчанні є:

- Підвищення наочності навчального матеріалу та полегшення його сприйняття завдяки компактному й чіткому поданню інформації.
- Розвиток творчого потенціалу суб'єктів навчання, їх здібностей, комунікативних дій, умінь експериментально-дослідницької діяльності.
- Інтенсифікація всіх рівнів навчального процесу, підвищення його ефективності та якості.
- Розширення та поглиблення змісту навчання з дисципліни, що вивчається.



- Засвоєння повного спектру понять, вільне оперування якими передбачено змістом навчальної дисципліни.

Науковцями виділено групу найважливіших чинників активізації навчальної діяльності студентів, ефективність яких може бути підсилена за рахунок застосування в навчальному процесі ІКТ:

a) Розвиток мотивації, посилення інтересу до навчання, у тому числі до способів одержання знань.

b) Індивідуалізація та диференціація навчання.

c) Розвиток самостійності.

d) Підвищення наочності навчання.

e) Збільшення арсеналу засобів пізнавальної діяльності, опанування сучасних методів наукового пізнання, пов'язаних із застосуванням комп'ютерів.

f) Спрощення та збільшення швидкості доступу до навчальної та наукової інформації через мережу інтернет [3].

Разом з перевагами існує і низка недоліків використання ІКТ:

- Кожному викладачу та здобувачу освіти необхідний доступ до комп'ютера.

- Необхідне спеціальне обладнання для забезпечення роботи програм;

- Розробка може вимагати значних фінансових витрат та витрат часу.

- Інтернет надає доступ до величезного обсягу інформації, яка може збивати студентів.

- Системи мультимедіа представляють насичене інформацією середовище і для повноцінної експлуатації їх можливостей необхідно вміло підібрати значну кількість інформації.

- Доступ із робочого столу до технічного забезпечення може бути ускладнено, особливо в корпоративних мережах.

- Низька якість зв'язку.
- Деякі студенти важко сприймають та засвоюють інформацію з екрана.
- Присутність невеликої кількості мультимедійних програм та онлайн-курсів українською мовою.
- Відсутність чіткої методики використання ІКТ в освіті.

Проте, незважаючи на певні перепони використання інформаційно-комунікаційних технологій у навчальному процесі, вони є потужним багатофункціональним засобом здобуття освіти. Використання ІКТ дозволяє оптимізувати процес здобуття освіти, реалізувати його творчий характер, організувати гнучке управління навчальною діяльністю, інтенсифікувати й активізувати навчання в закладах освіти.

Сучасний освітній процес передбачає створення єдиного освітнього простору, що формує умови для вільного руху інформації, швидкого доступу до неї всіх учасників навчального процесу. Цього можна досягти шляхом використання інформаційно-комунікаційних технологій під час навчального процесу.

На рисунку 1.1 показано структуру напрямків застосування ІКТ у освітньому процесі.

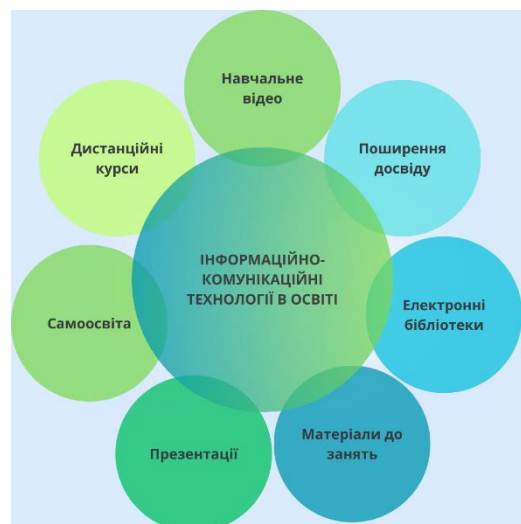


Рисунок 1.1 – Структура напрямків застосування ІКТ у освітньому процесі

До напрямків використання ІКТ в навчальному процесі відносять:

- a) Використання мережі Інтернет.
- b) Використання конструкторів навчально-методичних матеріалів.
- c) Використання платних та безоплатних цифрових сервісів та платформ.
- d) Використання комп'ютерних тестів з метою оцінювання рівня знань здобувачів освіти.
- e) Використання інтерактивних практичних завдань (тренажерів) з метою закріплення теоретичного матеріалу.
- f) Виконання творчих завдань.
- g) Використання мультимедійних можливостей комп'ютерної техніки та інше[6].

### **1.1.2 Потреба використання віртуальних тренажерів у навчанні**

Однією з важливих педагогічних вимог сучасності є удосконалення та підвищення ефективності, а також якості навчання. Створення нових методик та навчальних інструментів є актуальним питанням через те, що це збільшує рівень практичних навичок як студентів, так і трудових кадрів. Важливе місце серед різноманіття навчальних інструментів займають віртуальні тренажери, які сприяють ефективному засвоєнню теоретичного матеріалу та якісно підвищують практичні навички здобувачів освіти.

Віртуальний тренажер – інтерактивний навчально-тренувальний об'єкт, призначений для закріплення знань, набуття навичок та контролю знань у конкретній предметній галузі [7].

Використання інтерактивних віртуальних додатків може здешевити та спростити процес опанування потрібних навичок майбутнім спеціалістам багатьох професійних сфер (наприклад: майбутнім цивільним та військовим льотчикам, операторам дронів, молодим хірургам та фармацевтам, агроінженерам, а також спеціалістам ядерної фізики тощо) [8].

При використанні віртуальних тренажерів виділяють як позитивні, так і негативні сторони. До позитивного досвіду можемо віднести наступні:

- Полегшення доступу до навчальних матеріалів.
- Кожне завдання формується програмою.
- Контроль виконання завдань здійснюється покроково.
- Контроль правильності виконання завдань здійснюється інформаційною системою без втручання викладачів.
- Завдання можуть мати різний рівень складності.
- Тренажером можна користуватись без прив'язки до розкладу занять.

Недоліками вважаються відсутність живого спілкування з викладачем та колегами-студентами, а також відсутність можливості за допомогою тренажерів реалізувати креативні та творчі завдання, які передбачають використання студентами творчого підходу для їх здійснення [9-10].

При вірному підборі та використанні віртуальних тренажерів викладач має змогу зацікавити та мотивувати студентів до вивчення тієї чи іншої дисципліни. Враховуючи повсюдне впровадження дистанційної форми навчання, можна дійти висновку, що інтерактивні віртуальні додатки є досить простим та доступним, проте доволі потужним навчальним засобом сучасної освіти.

## **1.2 Аналіз існуючих веб-додатків для інтерактивного навчання**

Перетворення чисел з однієї системи числення в іншу є важливою темою в інформатиці та програмуванні. Студенти повинні вміти виконувати такі перетворення, щоб ефективно працювати з комп'ютерами.

Однак, перетворення чисел може бути складним завданням для студентів, які тільки починають вивчати інформатику. Вони можуть помилятися в розрахунках або не розуміти, як працює процес перетворення.

Ця проблема може бути вирішена за допомогою інтерактивного тренажера, який дозволить студентам практикуватися в перетвореннях чисел. Тренажер повинен бути простим у використанні та зрозумілим для студентів. Він також повинен містити пояснення того, як працює процес перетворення.

У глобальній мережі Інтернет ми маємо змогу відшукати та скористатись зручними онлайн калькуляторами для перетворення чисел з однієї системи числення в іншу. Однак їхній функціонал зводиться до безпосереднього перетворення чисел і не включає в себе освітньої складової. А от інтерактивних веб-тренажерів саме освітнього характеру є обмежена кількість.

Знайти вдалося кілька таких додатків. Перший з них – це learningapps [11]. Він має вигляд одно сторінкового сайту, як це показано на рисунку 1.2.

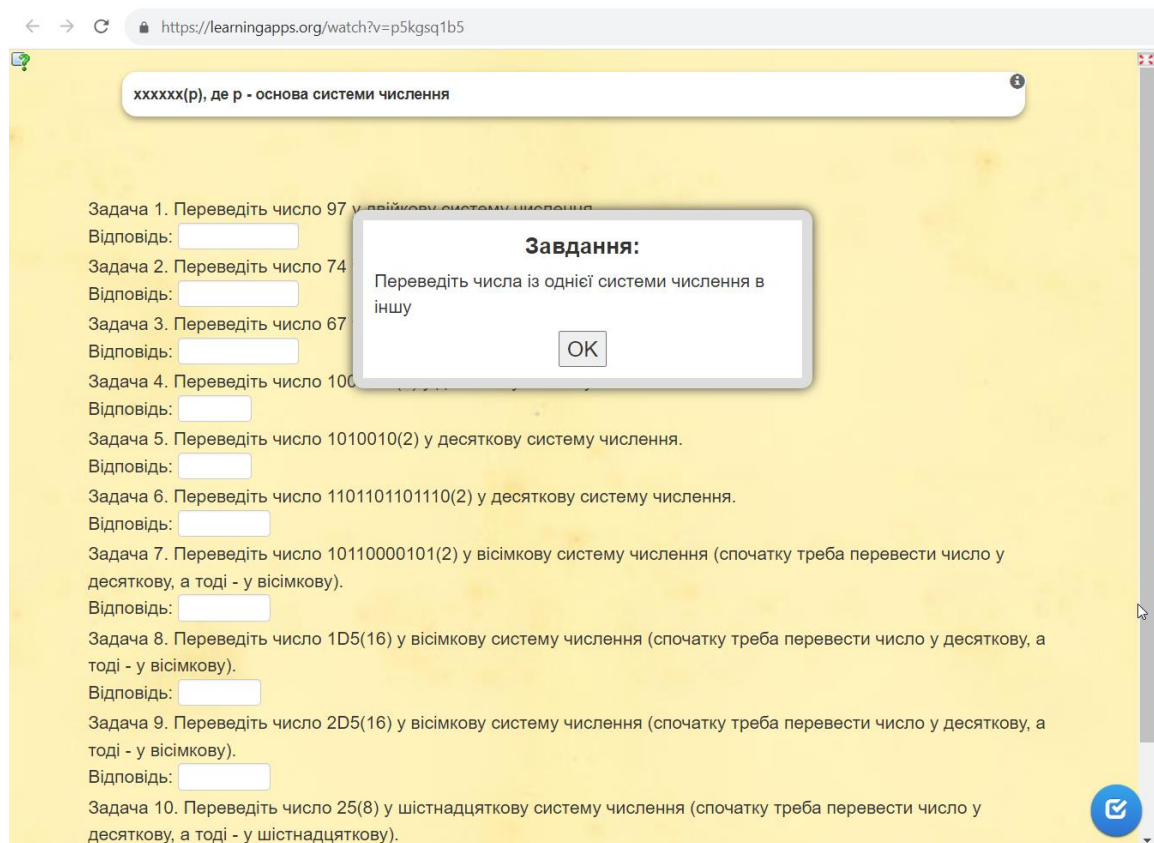


Рисунок 1.2 – Тренажер Learningapps

Основою сторінки є десять задач по перетворенню чисел з різних систем числення в інші. Для кожної задачі передбачено поле для вводу відповіді. Також у тренажері learningapps передбачено три кнопки, розміщені в різних кутах сторінки. Кнопка, що знаходиться у верхньому лівому куті, відповідає за

опис завдання. Кнопка верхнього правого кута передбачає повноекранний перегляд даної сторінки. У нижньому правому куті знаходиться кнопка, що дозволяє перевірити правильність виконання усіх десяти запропонованих задач. Верхня частина екрану має поле довідки, у якому наведено пояснення правильного запису числа в системі числення (де  $p$  – основа системи числення).

До переваг наведеного тренажера можна віднести наступні:

- а) Можливість для користувачів тренуватися у вирішенні задач, пов'язаних з переведенням чисел з однієї системи числення в іншу.
- б) Можливість перевірки правильності виконаних завдань (див. рисунок 1.3).
- с) Мінімалістичність інтерфейсу.

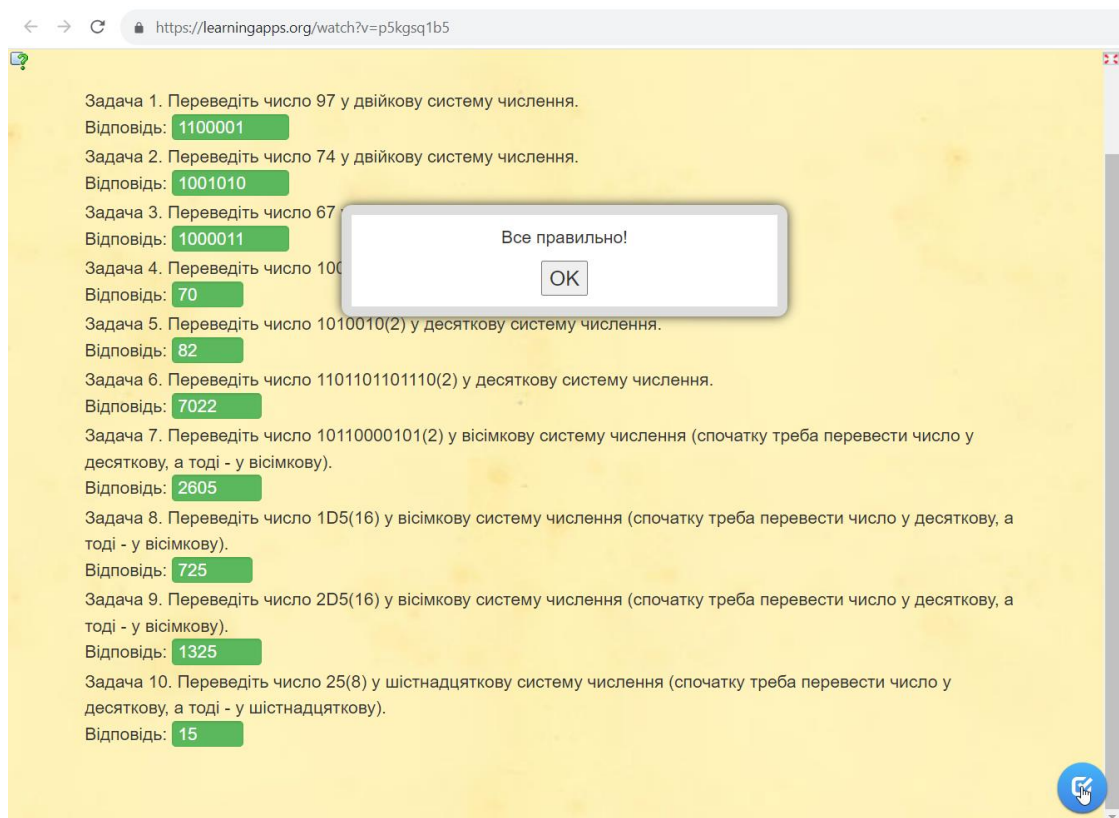


Рисунок 1.3 – Тренажер Learningapps з виконаними завданнями  
Проте дане рішення має деякі недоліки, такі як:

a) Відсутність можливості змінити вхідні дані задач. Є лише десять задач з незмінною умовою і незмінними числовими даними, що дає користувачу можливість скористатися цим тренажером усього лише один раз.

b) Повна відсутність підказок для користувача, які могли б нагадати алгоритм дій при вирішенні задач.

Також в Google Play є тренажер, розроблений для системи Android, під назвою Number System: Learn & Convert (див. рисунки 1.4-1.5). У цьому додатку можна перетворити ціле чи дробове число у двійкову, десяткову, вісімкову чи шістнадцяткову систему [12]. Це не лише навчальний інструмент, а ще й калькулятор. Ця система має простий інтерфейс користувача.

Тренажер Number System: Learn & Convert має два значні недоліки:

- Додаток розрахований тільки для мобільних пристроїв на операційній системі Android без можливості використання його на будь-яких інших пристроях.

- Тренажер генерує числа, конвертує їх в іншу систему числення і показує докладний розв'язок, що позбавляє користувачів інтерактивної взаємодії з додатком та можливості закріпити опанований теоретичний матеріал.

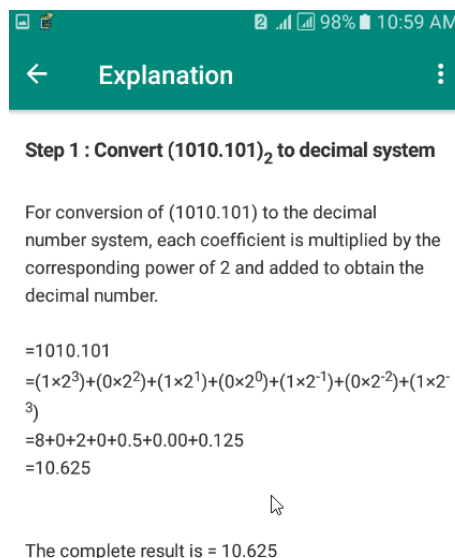


Рисунок 1.4 – Тренажер Number System: Learn & Convert

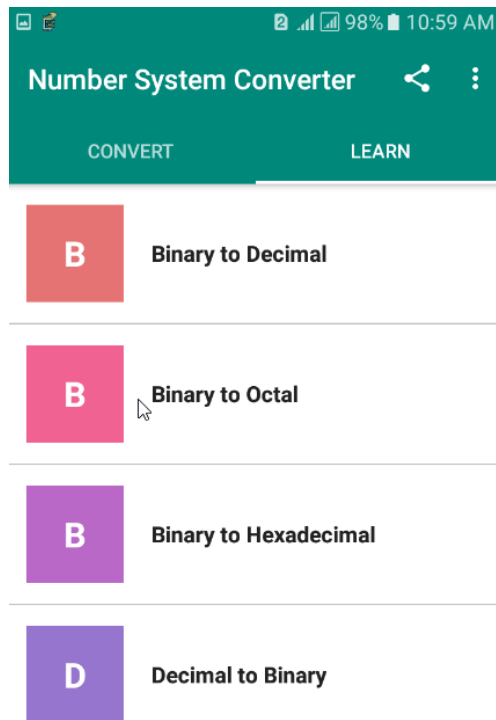


Рисунок 1.5 – Сторінка Number System: Learn & Convert з теоретичними блоками

Наступним аналогом є інтерактивна гра від компанії Cisco Binary Game [13], яка призначена для навчання основам двійкової системи числення (див. рисунок 1.6). Гра є безкоштовною та доступна для всіх користувачів.

Гра має два режими для різних рівнів підготовки (basic - в цьому режимі гравець повинен перетворювати двійкові числа в десяткові та навпаки, та advanced - в цьому режимі гравець повинен вирішувати задачі з двійкової арифметики).

Недоліком навчання за допомогою гри Binary Game можна назвати відсутність можливості отримати глибокі знання. Гра не може замінити повноцінне навчання. Вона може допомогти вивчити основи двійкової системи числення, але для отримання глибоких знань необхідно використовувати додаткові ресурси, такі як книги, статті або відео уроки.





Рисунок 1.6 – Інтерактивна гра Binary Game

Наведені вище приклади додатків з перетворення чисел є цікавими та показовими. Проте кожен з них має як свої переваги, так і свої недоліки.

На основі розглянутих прикладів для зручного порівняння всі переваги та недоліки вищеописаних продуктів було представлено у порівняльній таблиці 1.1.

Таким чином результатом проведеного аналізу являється висновок про те, що розроблення веб-тренажеру з перетворення чисел з однієї системи числення в іншу систему числення є актуальним. Він повинен містити в собі елементи інтерактивності, мати пояснення та підказки, бути інтуїтивно зрозумілим для користувачів. Продукт кваліфікаційної роботи може бути корисним під час вивчення теоретичного матеріалу та проведення практичних занять по темі «Системи числення. Перетворення чисел з однієї системи числення в іншу», а також може використовуватись для перевірки якості отриманих практичних навичок та засвоєння теорії.

Таблиця 1.1 – Порівняльна таблиця тренажерів аналогів

Назва додатку Критерій оцінювання	Learningapps	Number System: Learn and Convert	Binary Game
Безоплатність	+	+	+
Вільний доступ (відсутність реєстрації)	+	+	+
Охоплення цільової тематики	+	+	+/-
Відповідність предметній області дослідження	+/-	+/-	+/-
Наявність веб-версії додатку	+	-	+
Наявність освітньої складової	+/-	+/-	+/-
Наявність інтерактивної складової	-	-	+/-
Сучасність дизайну	-	-	+

### 1.3 Постановка задачі

За результатами проведеного вивчення аналогічних рішень можна сказати, що на даний момент на ринку не існує необхідної повноцінної інформаційної системи для самонавчання здобувачів освіти у сфері перетворення чисел з однієї системи числення в інші. Актуальність та затребуваність саме такого веб-тренажера не викликає сумнівів.

Метою даної кваліфікаційної роботи є створення інформаційної системи – тренажеру для перетворення чисел з однієї системи числення в іншу систему

числення з метою закріплення студентами вивченого теоретичного матеріалу та розвитку набутих практичних навичок.

Поставлена мета передбачає вирішення наступних завдань:

- a) Аналіз теоретичного навчального матеріалу даної тематики.
- b) Аналіз аналогічних проєктів (визначення переваг та недоліків).
- c) Розроблення дизайну додатку.
- d) Визначення методології та актуальних технологій для реалізації проєкту.
- e) Розроблення і реалізація веб-тренажеру та теоретичних блоків до завдань.
- f) Розроблення локалізації інтерфейсу інтерактивного додатку в українському та англійському варіантах.
- g) Тестування роботи тренажеру.
- h) Впровадження тренажеру до єдиної навчальної платформи СумДУ.

Продукт має відповідати всім вимогам сучасного освітнього процесу, а також давати змогу студентам оволодіти методологію перетворення чисел у дистанційному форматі навчання.

## 2 МЕТОДОЛОГІЯ ВИРІШЕННЯ ЗАВДАННЯ

### 2.1 Середовище розробки

Перед кожним програмістом (студентом чи професіоналом) на початку написання програми постає питання вибору оптимального варіанту інтегрованого середовища розробки (скорочено IDE). IDE – це комплексне програмне рішення для розробки програмного забезпечення, що як правило складається з редактора початкового коду, інструментів для автоматизації складання та відлагодження програм. Більшість сучасних середовищ розробки мають можливість автодоповнення коду. IDE завантажує проєкт (у якому може бути багато файлів), дозволяє переключатися між ними, інтегрується із системою контролю версій (наприклад, Git), надає можливість розгортання вашого проєкту на тестове середовище та багато інших функцій [14].

Існують десятки, а то і сотні різних варіантів інтегрованих середовищ розробки. Програмісти-початківці на старті своєї кар'єри, як правило, надають перевагу варіантам IDE, які мають наступні характеристики: безоплатні, кросплатформові, мають автозавершення коду та підказки при наборі коду (функції, ключові слова, змінні, оголошені раніше тощо), підсвічування синтаксису, «знання» HTML5, CSS3, JavaScript та інше. Серед таких можна зазначити NetBeans, IntelliJ Idea Community Edition, Visual Studio Code тощо. Розглянемо короткий опис деяких з них.

IntelliJ Idea Community Edition – це вільна версія в рамках ліцензії Apache 2.0. Вона призначена для JVM та Android розробки, а також програм із GUI (Graphical user interface, графічний інтерфейс користувача). Вона буде корисна як розробникам-початківцям для освітніх цілей, так і професіоналам для комерційної розробки. Community-версії поставляється для трьох платформ: Windows, macOS, Linux. Актуальну версію продукту можна завантажити з офіційного сайту компанії JetBrains [15].

NetBeans IDE – безкоштовне інтегроване середовище розробки з відкритим вихідним кодом для розробників програмного забезпечення. Середовище надає всі необхідні засоби, необхідні для створення професійних десктоп застосунків, корпоративних, мобільних та веб-застосунків на платформі Java, а також C/C++, PHP, JavaScript, Groovy та Ruby.

Для встановлення актуальної версії необхідно перейти на офіційний сайт Apache NetBeans у розділ Download [16]. Для завантаження існують версії для Linux, macOS та Windows. Обираємо, завантажуюмо та інсталуємо на свій комп'ютер, використовуючи інструкції.

До основних характеристик NetBeans IDE відносять: настроювану робочу область середовища, розширений багатомовний редактор для різних мов програмування (Java, C/C++, Ruby, Groovy, PHP, JavaScript, CSS, XML, HTML), перевірку відповідності дужок та слів, рефакторинг, виділення синтаксичних конструкцій кольором, автодоповнення мовних конструкцій на льоту, шаблони коду та інше.

Visual Studio Code (VS Code) – це легкий, але потужний редактор вихідного коду, розроблений Microsoft для Windows, macOS і Linux. Він поставляється з вбудованою підтримкою JavaScript, TypeScript, Node.js, CSS та HTML і має багату екосистему розширень для інших мов і середовищ виконання (таких як C, C++, C#, Java, Python, PHP, Go, .NET, Python) [17].

Функції включають підтримку, налагодження, підсвічування синтаксису, інтелектуальне завершення коду, фрагменти, рефакторинг коду та вбудований Git. Користувачі можуть змінювати тему, комбінації клавіш, параметри та встановлювати розширення, які додають функціональність.

Керування джерелом є вбудованою функцією Visual Studio Code. Він має спеціальну вкладку всередині панелі меню, де користувачі можуть отримати доступ до налаштувань керування версіями та переглянути зміни, внесені до поточного проекту. Щоб використовувати цю функцію, Visual Studio Code має бути пов'язано з будь-якою підтримуваною системою керування версіями (Git,

Apache Subversion, Perforce тощо). Це дозволяє користувачам створювати репозиторії, а також робити запити push і pull безпосередньо з програми Visual Studio Code.

В опитуванні розробників Stack Overflow 2023 серед 86 544 респондентів Visual Studio Code назвали найпопулярнішим інструментом середовища розробника, при цьому 73,71% повідомили, що вони ним користуються. Опитування також показало, що Visual Studio Code частіше використовують ті, хто вчиться програмувати, ніж професійні розробники (78% проти 74%) [18].

Для того, щоб встановити VS Code, необхідно завантажити з електронного ресурсу [19]. Можна обрати для завантаження версії для macOS, Linux та Windows. Обираємо потрібну, завантажуюмо на свій комп'ютер та інсталуємо, використовуючи інструкції.

Для розробки поточного проекту було обрано середовище Visual Studio Code тому, що ця IDE містить зручне автодоповнення коду, підсвічування синтаксису та інший необхідний функціонал для зручного та швидкого написання коду.

## **2.2 Інструменти графічного дизайну**

В процесі розробки додатку з'явилась потреба у створенні логотипу тренажера і постало питання вибору інструментів для створення графічного дизайну. На ринку існує велика кількість програм та онлайн платформ графічного дизайну, як от: Adobe Illustrator, Adobe Photoshop, Piktochart, Canva та багато інших [20]. Розглянемо декілька з них.

Adobe Illustrator – один із найпопулярніших векторних графічних редакторів, створений компанією Adobe Systems. Він містить широкий набір інструментів і дає можливість розробляти різноманітні дизайнерські продукти від логотипів та значків веб-сайтів до упаковок, книжкових ілюстрацій та рекламних щитів.

Adobe Illustrator має вбудовані стилі та шаблони, що допомагає пришвидшити створення дизайну. Інтеграція з Adobe Creative Cloud дає можливість користувачам ділитися своїми проектами.

До основних особливостей програми відносять: інтерактивний інтерфейс; широкий спектр функцій; вбудовані пресети та шаблони дизайну; можливість прив'язки до пікселя; ідеальний піксельний дизайн.

Проте ця програма являється платною, що робить її не дуже привабливою для молодих дизайнерів. Її можна придбати на офіційному сайті Adobe [21].

Adobe Photoshop являється лідером в галузі комерційних засобів для редагування растрових зображень і найбільш відомим продуктом фірми Adobe. Цей графічний редактор є найефективнішим і розповсюдженим програмним забезпеченням для роботи дизайнерів, починаючи від невеликих редагувань до створення складних дизайнерських рішень.

Користувач може використовувати його для ізоляції елементів від фону зображення, визначення розміру полотна, створення власних пензлів і виконання інших завдань. Photoshop має величезну кількість корисних інструментів, що зробило його однією з найкращих програм графічного дизайну для початківців і професіоналів.

Ключовими особливостями Adobe Photoshop є інструменти вибору; векторна маска; інструменти ретуші; розумні об'єкти; фотошоп 3D.

Проте, як і перше розглянуте рішення, ця програма являється платною. Хоча новим користувача надається можливість скористатися пробною 30-денною версією [20, 22].

Розглянемо ще одне популярне рішення для проектування графіки онлайн – Canva.

Canva – це хмарна платформа, що дає можливість користувачам створювати графічний дизайн професійної якості. Має простий у використанні

редактор перетягування, завдяки чому програму можуть використовувати і початківці, і професіонали, а також команди при створенні спільних проектів.

Користувач має доступ до понад мільйона шрифтів, графіки та зображень, що робить процес створення дизайну легким та швидким. Ця платформа має велику бібліотеку шаблонів для налаштування та створення дизайну, та надає можливість користувачам опублікувати власні дизайни на платформі, завантажити та роздрукувати їх.

Canva є безкоштовним рішенням для команд до 10 учасників, що надає доступ до понад восьми тисяч шаблонів і 1 ГБ пам'яті. Проте, якщо цей функціонал замалий для задач користувача, то є можливість придбати недорогий план, який розблокує більшу кількість функцій, розширені інструменти, а також необмежений обсяг пам'яті [20].

Особливостями платформи Canva є наступні зоряні фотоефекти; функція команд; іконки та рамки; дизайнерський кроппер ; фотосітки.

Зважаючи на те, що поставлена задача розробки логотипу тренажера не вимагає розширеного інструментарію від графічного редактора, було прийняте рішення використати онлайн платформу графічного дизайну Canva. Вона виявилася дуже зручною та інтуїтивно зрозумілою при створенні та редагуванні зображення. Додатковим бонусом було зекономлене місце на комп'ютері за відсутності потреби встановлення програми.

Скористатися можливостями даної онлайн платформи графічного дизайну можна за покликанням <https://www.canva.com/> [23].

### **2.3 Вибір локального сервера**

За допомогою локального сервера можна готувати сайт до публічного доступу. На етапі розробки ресурс бачать тільки розробники. Це позбавляє живих відвідувачів і пошукових роботів від взаємодії з незаповненим і недопрацьованим сайтом. А ви можете перевірити працездатність ресурсу,



освоїти адміністративну консоль і виконати інші необхідні дії в безпечних умовах. Локальний сервер допоможе легко і швидко протестувати проект.

Локальний сервер – це спеціальний набір програм, які імітують роботу реальних серверів хостинг-провайдерів, а також забезпечують правильну роботу скриптів прямо у вас на комп'ютері. Локальний сервер забезпечує доступ до ресурсу через браузер. Набір програм і скриптів перетворює інформацію баз даних MySQL і мов програмування типу Perl і PHP, HTML і CSS-код, що «розуміють» веб-оглядачі.

Локальний сервер встановлюється безпосередньо на персональний комп'ютер розробника та використовується для створення сайту без виходу в інтернет. Для реального тестування веб-застосунків використовуються популярні локальні сервери, такі як Denwer, Open Server, Live Server та інші [24]. Далі розглянемо кожен із них.

Denwer – це безкоштовний локальний сервер для тестування сайтів, веб-додатків або Інтернет сторінок. Denwer називають ще джентльменським набором розробника, оскільки він складається із усіх необхідних компонентів. Цей локальний сервер рекомендують для початківців через простоту і легкість у налаштуванні. Локальний сервер Denwer працює тільки на операційну систему Windows, розповсюджується безкоштовно і його можна завантажити з офіційного сайту розробника [denwer.ru](http://denwer.ru). За допомогою цього програмного забезпечення можна запустити практично будь-яку CMS на своєму комп'ютері.

Денвер займає кілька мегабайт і стане чудовим етапом підготовки сайту, перш ніж його заливати на хостинг. Третя версія підтримує роботу з переносного флеш-накопичувача [25].

Наступний локальний сервер – портативний Open Server. Цей дистрибутив не такий старий і популярний як Денвер.

Open Server – це безкоштовний локальний сервер з високим функціоналом по розробці і створенню сайтів. Працює в операційній системі

Windows. Завантажити можна з офіційного сайту <https://ospanel.io/download/> [26].

Open Server має багато компонентів і має потужні можливості. Також використовується для розробки та тестування веб-сайтів у локальній мережі.

Якщо порівнювати два даних локальних сервера, варто врахувати, що Open Server включає новіші версії програмного забезпечення для розробки ніж Денвер. Однак Denwer важить набагато менше пакету Open Server (8,2 Мб проти 135 Мб). До всього сказаного почати працювати з Denwer значно легше. Однак робота в Open Server комфортніша загалом [24].

Live Server – це розширення для Visual Studio Code, яке дозволяє автоматично оновлювати веб-сторінки в браузері при зміні файлів HTML, CSS або JavaScript. Це робить процес розробки веб-сайтів більш ефективним і продуктивним. Щоб встановити це розширення, необхідно перейти у вкладку "Extensions" і в пошуку знайти необхідний плагін.

Live Server дозволяє бачити зміни коду, відображені у браузері. Він запускає локальний сервер розробки з функцією перезавантаження у реальному часі як статичних, так динамічних сторінок. Щоразу, коли код зберігається, розробник миттєво може побачити зміни в реальному часі, не потрібно вручну перезавантажувати браузер. Це дозволяє набагато швидше виявляти помилки, значно скорочує час роботи і буде набагато простіше провести швидкі експерименти з кодом. Це розширення дає можливість створити свій особистий сервер. Тільки встановивши розширення можна запустити сервер для власного проекту без додаткових інструментів [27].

Оскільки середовищем розробки було обрано Visual Studio Code, то для підтримки розробки інтерактивного веб-тренажера було використано локальний сервер Live Server. Це зробить процес розробки додатку більш ефективним і продуктивним. Ще одним критерієм вибору саме цього локального серверу є зручне автоматичне оновлення веб-сторінки в браузері після збереження коду.

## 2.4 Вибір програмних засобів

Зважаючи на те, що інтерактивний додаток планується розмістити на сервері Єдиної навчальної платформи СумДУ, для якої вже існує спеціально розроблений фреймворк для тренажерів, широта вибору програмних засобів значно скорочується. Спеціалістами СумДУ розроблено фреймворк (рушій) для тренажерів мовою JavaScript.

Саме тому для реалізації функціоналу інформаційної веб-системи були обрані такі програмні засоби: мови HTML, CSS та JavaScript та фреймворки Bootstrap, JSON, jQuery, Mustache. Розглянемо кожен з них більш детально.

Як пояснює вікіпедія «HTML (HyperText Markup Language – мова розмітки гіпертексту) – стандартизована мова розмітки документів для перегляду веб-сторінок у браузері. Браузери отримують HTML документ від сервера за протоколами HTTP/HTTPS або відкривають з локального диска, далі інтерпретують код в інтерфейс, який відобразатиметься на екран монітора» [28, 29].

За допомогою HTML розробник позначає текст, вказуючи веб-переглядачу, як він повинен розуміти цей текст. HTML не є мовою програмування, тобто він не має можливості створювати динамічну функціональність.

У HTML текст позначають за допомогою тегів. Кожен HTML документ складається з групи елементів, де кожен з них визначається (починається та закінчується) певним тегом.

Популярність HTML зростає дуже швидко. Ось чому HTML сьогодні вважається офіційним веб-стандартом.

CSS (Cascading Style Sheets, або Каскадні таблиці стилів) – це спеціальна мова стилю HTML сторінок, яка використовується для опису їх зовнішнього вигляду. Разом із HTML та JavaScript CSS є основною технологією всесвітньої павутини [30, 31].

В основному CSS використовується для візуалізації HTML та XHTML сторінок. Наприклад, додавання фонових кольорів, шрифтів і розміщення вмісту в рядках або стовпцях, змінення розміру картинок. Основне завдання CSS зробити сайт привабливим та зручним для користувачів, надати проєкту завершеного вигляду. Таблиці стилів працюють з документом, а не створюють нові елементи.

Сьогодні без використання CSS-стилів неможливо створити повноцінний сучасний сайт. Після першого запуску таблиці стилів кешуються і запам'ятовуються браузером, що дає змогу швидше завантажувати сторінки в майбутньому. HTML формує структуру веб-документів, тому завантажується першим, а CSS створює розмітку, оформлення та стиль, тому завантажується після HTML.

JavaScript (JS) – динамічна, об'єктно-орієнтована прототипна мова програмування. Використовується для створення сценаріїв веб-сторінок, що дає можливість на боці клієнта взаємодіяти з користувачем, керувати браузером, обмінюватися даними з сервером в асинхронному режимі, змінювати структуру, а також зовнішній вигляд веб-сторінки.

JavaScript дозволяє впроваджувати низку складних рішень у веб-документах: допомагає зробити сторінку сайту інтерактивною, обробляє дії користувачів сайту [32]. Коли веб-сторінка не просто відображає статичну інформацію, на яку можна дивитися, а відображає своєчасні оновлення вмісту, інтерактивні карти, анімовану 2D/3D-графіку, музичні автомати з прокручуванням відео тощо. мова програмування, яка підтримується програмами, що працюють із дизайном сайту. JavaScript стала популярнішою серед програмістів, коли з'явилася AJAX-технологія, яка призвела розробку сайтів на новий рівень.

JavaScript широко використовується у фронтенд розробці. Поряд з HTML і CSS Javascript являється третім важливим блоком, на якому базується основна частина стандартних веб-інтерфейсів. Коли сторінка завантажується в браузер,

спочатку обробляються HTML і CSS, а JavaScript завантажується після цього. JavaScript обробляється у веб-додатках в браузері, тобто на стороні клієнта. Тому ця мова може працювати на будь-якій операційній системі, а веб-інтерфейси, що працюють на ній, є кросплатформними.

Зараз JavaScript є однією з найпопулярніших мов програмування. У щорічному опитуванні DOU JavaScript посіла перше місце (19,1%) серед мов програмування у 2023 році [33, 34]. Завдяки їй можна покращити досвід користування вебсайтами, похвалити їх дизайн та навіть створити повноцінні застосунки. Розробників, які володіють нею, дуже цінують на ринку.

Bootstrap – це безкоштовний та відкритий HTML, CSS та JS фреймворк для верстки адаптивних дизайнів веб-сайтів та веб-додатків. Найчастіше його використовують у фронтенд розробці. Цей фреймворк спрощує с пришвидшує процес розробки завдяки наявності шаблонів HTML і CSS для типографіки, форм, кнопок, таблиць, навігації та інших компонентів інтерфейсу, а також плагінів JavaScript.

Фреймворк Bootstrap - це набір CSS та JavaScript файлів. Щоб його використовувати, потрібно підключити ці файли до сторінки або включити аналогічний функціонал з CDN (Content Delivery Network).

До переваг Bootstrap відносять: простоту у використанні; високу швидкість створення адаптивної верстки; можливість налаштування під конкретний проект; наявність єдиного стилю усіх компонентів; наявність адаптивного CSS (адаптується до планшетів, персональних комп'ютерів і телефонів); сумісність з більшістю сучасних браузерів.

Недоліками можна назвати наступні факти: великий розмір кінцевих CSS та JS-файлів проекту; складність у використанні для верстки унікальних за дизайном сайтів [35,36].

Усіма можливостями даного фреймворку можна скористатись перейшовши за посиланням <https://getbootstrap.com/> .

Mustache — це система веб-шаблонів із реалізаціями, доступними для ActionScript, Go, Java, JavaScript, Python та інших мов програмування.

Moustache — це нелогічна система шаблонів, яка розділяє вигляд і код, так як в ній відсутні будь-які явні оператори потоку керування (if-else) та цикли (for). Проте цикли і умовні оцінювання можна описати за допомогою списків обробки тегів розділу та анонімних функцій (лямбда).

Дана система отримала назву «Moustache», тому що інтенсивно використовуються дужки {}, що нагадують косі вуса. В основному Moustache використовується для мобільних і веб-додатків.

Основними принципами системи є відсутність логіки та відокремлення проблем. [37]. Для встановлення даного шаблонизатора можна перейти за адресою <http://mustache.github.io/>.

AJAX - це поєднання двох технологій (Javascript і XML) технологій, які надають розробникам низку програмних рішень. AJAX це акронім, який означає Asynchronous Javascript and XML. Його інші компоненти включають HTML, CSS, PHP та DOM.

AJAX - це підхід до розробки веб-інтерфейсу користувача, що передбачає обмін даними між веб-браузером і ресурсами сервера без перезавантаження веб-сторінки. Асинхронне оновлення сторінки дозволяє веб-ресурсам працювати більш зручно та злагоджено. Завдяки цьому також можна мінімізувати використання трафіку в умовах обмеженого доступу в Інтернет [38, 39].

AJAX не завжди рекомендується для розробки. У деяких випадках ця технологія має свої переваги, але іноді вона може негативно вплинути на роботу вашого веб-ресурсу.

Для спілкування з сервером в AJAX можна використовувати три формати даних: HTML, XML, JSON.

JSON (JavaScript Object Notation або запис об'єктів JavaScript) - це текстовий формат обміну даними між комп'ютерами. JSON може бути

прочитаним людиною, тому що базується на тексті. За допомогою цього формату можна описати об'єкти та інші структури даних. Завдяки процесу серіалізації JSON використовується для передавання структурованої інформації через мережу.

Головне призначення JSON в написанні веб-програм при використанні технології AJAX. Однією з переваг даного формату перед XML це те, що JSON дозволяє складні структури в атрибутах, займає менше місця, а також прямо інтерпретується за допомогою JavaScript в об'єкти.

При розробці веб-застосунків JSON доречно застосовувати при обміні даними і між браузером та сервером (AJAX), і між самими серверами (програмні HTTP-інтерфейси).

JSON побудовано на двох структурах:

1. Набір пар ім'я/значення (у різних мовах це реалізовано як об'єкт, запис, структура, словник, хеш-таблиця, список із ключами або асоціативний масив).
2. Упорядкований список значень (у більшості мов це реалізовано як масив, вектор, список або послідовність) [40, 41].

jQuery – це JavaScript-бібліотека з відкритим кодом, яка допомагає спростити написання великих фрагментів коду. Бібліотека jQuery включає в себе величезну кількість готових функцій, які допомагають розробникам створювати інтерактивні елементи на сайті. Ця бібліотека працює з елементами DOM і дозволяє ними маніпулювати, а також вона допомагає реалізувати модулі API для використання AJAX [42].

Основна мета фреймворку jQuery - програмувати елементи веб-ресурсів і створювати різні веб-програми. Технологія jQuery набула великої популярності серед програмістів завдяки швидкості освоєння бібліотеки, невеликій кількості використовуваного коду та короткому часу його написання.

Фреймворк jQuery надає такі функції, як маніпуляції DOM; маніпуляції CSS; обробка та маніпулювання подіями HTML; візуальні ефекти та анімація; доповнення до AJAX; різні розширення JavaScript.

Принцип роботи jQuery полягає в використанні класу (функції), який при звертанні до нього повертає сам себе. Таким чином, це дозволяє будувати послідовний ланцюг методів.

Із багатьох існуючих бібліотек JavaScript, jQuery є найпопулярнішою та більш розширюваною. Більше 19 мільйонів веб-сайтів зараз використовують jQuery. Такі великі компанії, як Google, Microsoft, IBM, Netflix та інші використовують jQuery.

Отже, для розробки інформаційно-комунікаційної технології з перетворення чисел було обрано наступні інструменти: середовище Visual Studio Code, онлайн платформу графічного дизайну Canva, локальний сервер Live Server, мови HTML, CSS та JavaScript та фреймворки Bootstrap, JSON, jQuery, Mustache.



### 3 ПРОЄКТУВАННЯ СИСТЕМИ

#### 3.1 Структурно-функціональна модель

За допомогою методології IDEF0 описуємо функціональну модель роботи системи. IDEF0 – це методологія функціонального моделювання і графічного опису процесів, що призначається для формалізації і опису бізнес-процесів. Ця методологія передбачає побудову ієрархічної системи діаграм для опису системи в цілому, а також її взаємодії з навколишнім середовищем.

Функціонал IDEF0 складається із п'яти елементів: дія (або процес) представлена блоками; вхідні дані представлені стрілками, що перетікають у ліву частину блока; вихідні дані - це стрілки, що виходять із правої сторони блока; стрілки у верхній частині блока представляють обмеження або керуючі елементи; а стрілки знизу блока - це механізми, які виконують функції [43]. Кожен блок описується дієсловом, тому що він являється функцією (процесом).

Інструментом створення діаграм в даному випадку є СА ERwin Process Modeler. Контекстна діаграма основного процесу «Набувати та перевіряти якість практичних навичок» представлена на рисунку 3.1.

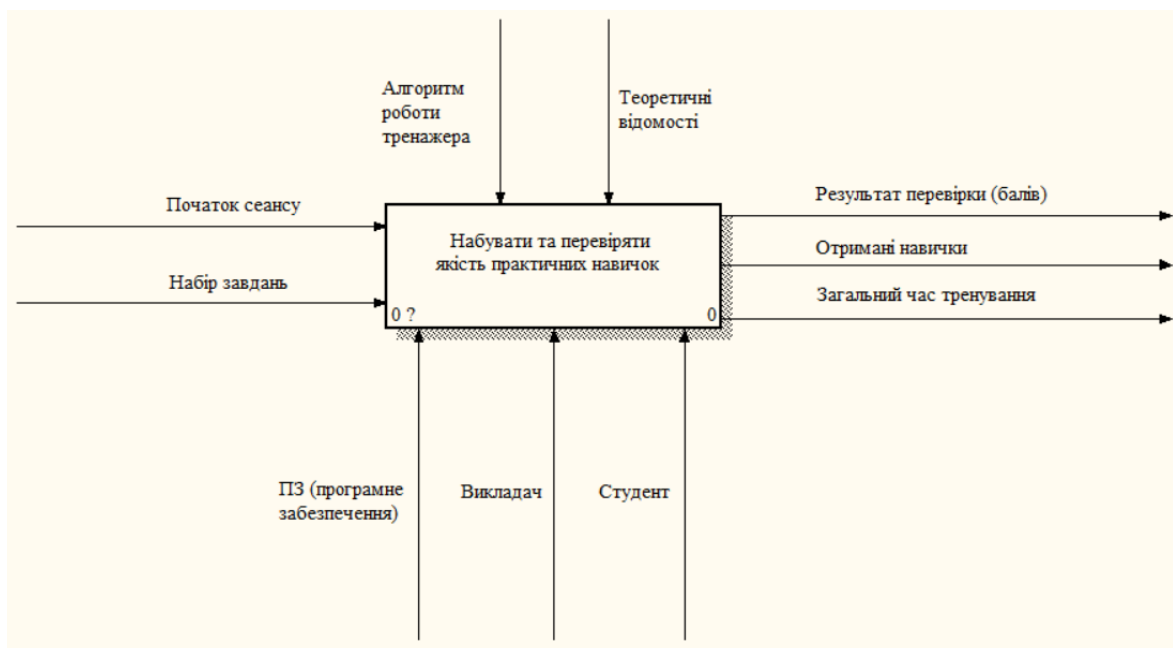


Рисунок 3.1 – Контекстна діаграма основного процесу

Дані для веб-тренажера з діаграми IDEF0 розподілені на такі типи:

- a) вхідні дані: початок сеансу, набір завдань;
- b) керуючі елементи: теоретичні відомості, алгоритм роботи тренажера;
- c) механізми процесу: студент, викладач, ПО (програмне забезпечення);
- d) вихідні дані: результат перевірки (балів), отримані навички, загальний час тренування.

Наступним кроком здійснюється декомпозиція процесу «Набувати та перевіряти якість практичних навичок» на чотири підпроцеси «Обрати мову», «Виконати завдання», «Перевірити відповіді» та «Вивести результати».

Декомпозиція процесу «Набувати та перевіряти якість практичних навичок» зображена на рисунку 3.2.

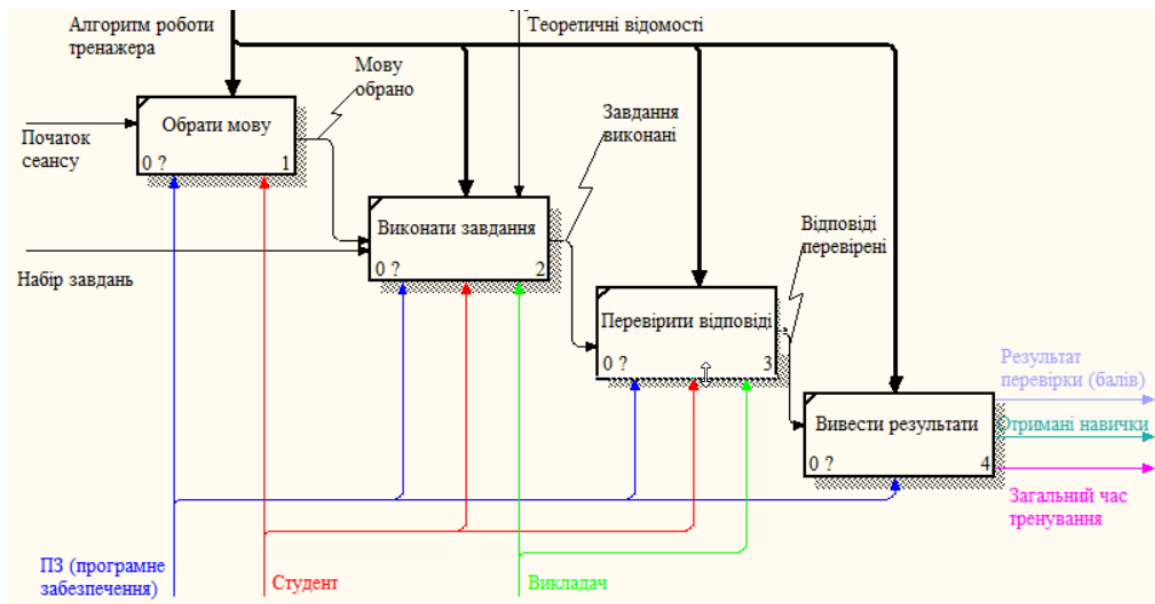


Рисунок 3.2 – Діаграма декомпозиції процесу «Набувати та перевіряти якість практичних навичок»

Даними для першого процесу «Обрати мову» являються:

- вхідні дані: початок сеансу;
- керуючі елементи: алгоритм роботи тренажера;
- механізми процесу: студент, ПО (програмне забезпечення);

- вихідні дані: мову обрано.

Другий процес «Виконати завдання» має такий перелік даних:

- вхідні дані: мову обрано, набір завдань;
- керуючі елементи: теоретичні відомості, алгоритм роботи тренажера;
- механізми процесу: студент, викладач, ПО (програмне забезпечення);
- вихідні дані: завдання виконані.

У третього процесу «Перевірити відповіді» перелік даних наступний:

- вхідні дані: завдання виконані;
- керуючі елементи: алгоритм роботи тренажера;
- механізми процесу: студент, викладач, ПО (програмне забезпечення);
- вихідні дані: відповіді перевірені.

Даними четвертого процесу «Вивести результати» являються:

- вхідні дані: відповіді перевірені;
- керуючі елементи: алгоритм роботи тренажера;
- механізми процесу: ПО (програмне забезпечення);
- вихідні дані: результат перевірки (балів), загальний час тренування, отримані навички.

### **3.2 UML моделювання системи**

Скористаємося уніфікованою мовою моделювання (UML - Unified Modeling Language) для реалізації даного проекту. UML використовується для візуалізації, проектування та складання документації бізнес-систем, програмних систем та систем, що пов'язані з програмним забезпеченням.

За допомогою UML можна розробляти різні типи діаграм, які умовно поділяють на 3 групи: діаграми взаємодії, діаграми поведінки та структурні діаграми [44]. Загалом діаграми поведінки (Behavior diagrams) відображають

динамічний аспект системи. Вони показують, що має відбуватися в даній системі, та часто використовуються, щоб описати функціональність програмних систем.

В контексті даного проєкту розглянемо дві підгрупи з групи діаграм поведінки: діаграму варіантів використання (Use Case Diagram) та діаграму послідовності (Sequence Diagram).

За допомогою діаграми варіантів використання можна описати поведінку системи з погляду користувача. Вона визначає межі системи, відношення між системою та зовнішнім середовищем. А також її використовують для візуалізації вимог до системи.

На діаграмах варіантів використання зображують варіанти використання, акторів та відношення (зв'язки) між ними.

Варіант використання (прецедент або use case) відповідає за спосіб взаємодії з системою та служить для опису сервісів, які система надає актору. Сукупність прецедентів відображає функціональність системи. Ім'я варіанту використання розміщується всередині еліпсу або під ним.

Актор (дійова особа або actor) описує роль, яку може грати персона чи інша зовнішня сутність при взаємодії з системою. Один актор може узагальнювати в собі багатьох фізичних осіб, які мають однакові функції відносно системи. Проте одна фізична особа може мати кілька ролей, а її функції будуть описуватись багатьма акторами.

Бувають наступні типи зав'язків між акторами та варіантами використання: асоціація, залежність зі стереотипом "включає" ("include"), залежність зі стереотипом "розширює" ("extend"), узагальнення.

В рамках поточного проєкту були визначені наступні актори: користувач (user), сервер (server) та рушій тренажера (trainier.engine.js). Варіанти використання виділено наступні: змінити мову, виконати завдання та отримати результати.

Діаграма варіантів використання зображена на рисунку 3.3.

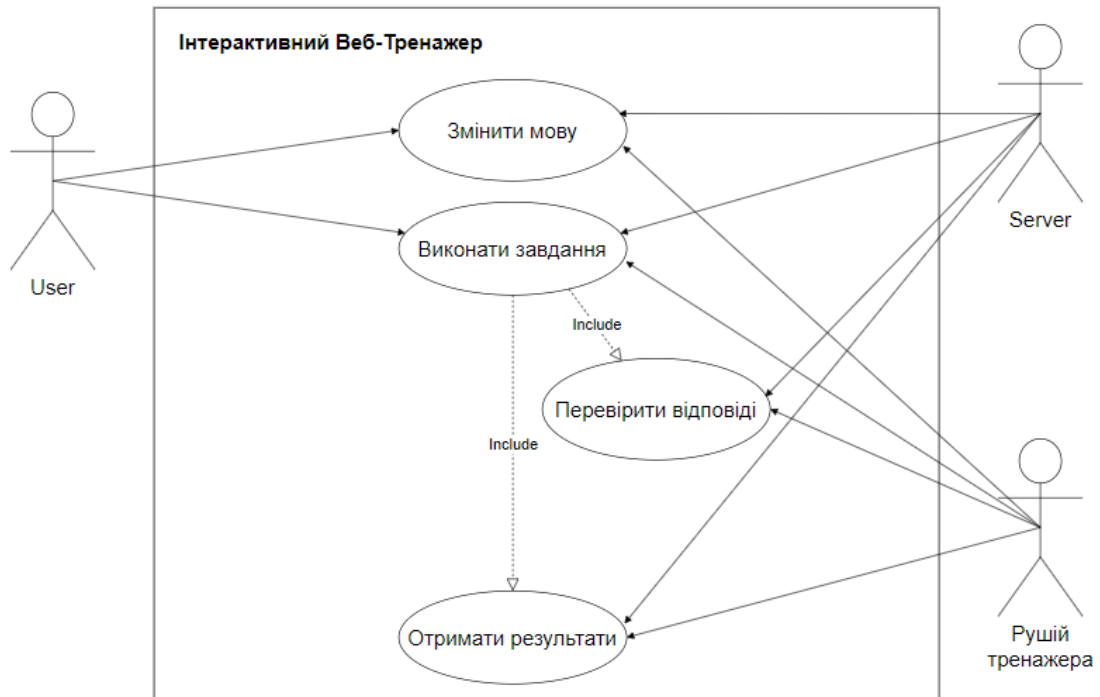


Рисунок 3.3 – Діаграма варіантів використання

Діаграму послідовності (Sequence Diagram) часто використовують для уточнення діаграми прецедентів (варіантів використання), так як вона описує поведінкові аспекти системи. Взаємодія об'єктів в динаміці та часі відображається діаграмою послідовності. А інформація отримує вигляд повідомлень, при цьому взаємодія між об'єктами передбачає обмін такими повідомленнями за сценарієм. Тобто такі діаграми зображують послідовність повідомлень, якими обмінюються між собою об'єкти. При використанні діаграм взаємодії змальовується поведінка системи, коли реалізуються різні варіанти використання.

Діаграма послідовності - це графічне зображення послідовної взаємодії об'єктів. В діаграмі послідовності по горизонталі розміщуються об'єкти, що взаємодіють між собою, а по вертикалі розміщено вимір часу. Діаграма послідовності має такі елементи: об'єкт, лінію життя, повідомлення, фокус управління, повідомлення до себе.

Діаграми послідовності розроблені на основі наступних варіантів використання: «Змінити мову», «Виконати завдання» та «Отримати

результати». На рисунках 3.4 – 3.6 зображені діаграми послідовності відповідно визначеним вище варіантам використання.

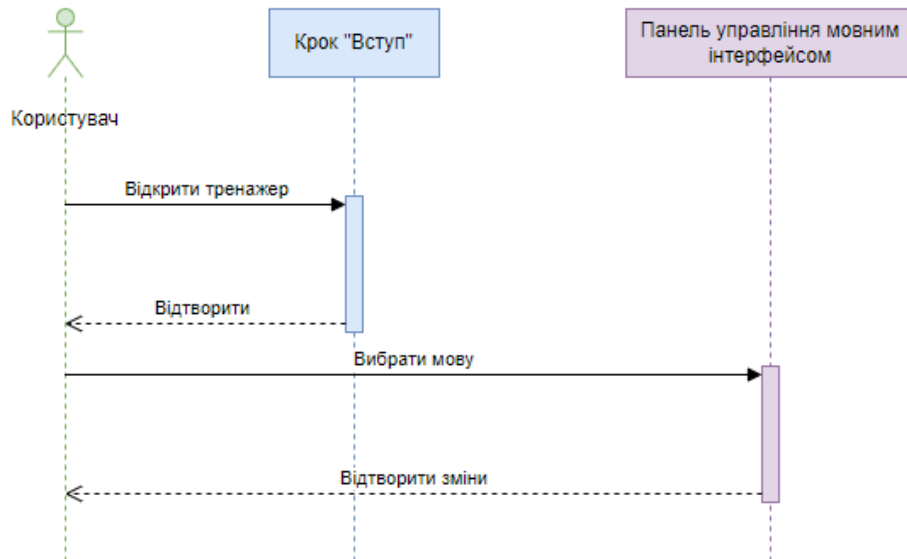


Рисунок 3.4 – Діаграма послідовності «Змінити мову»

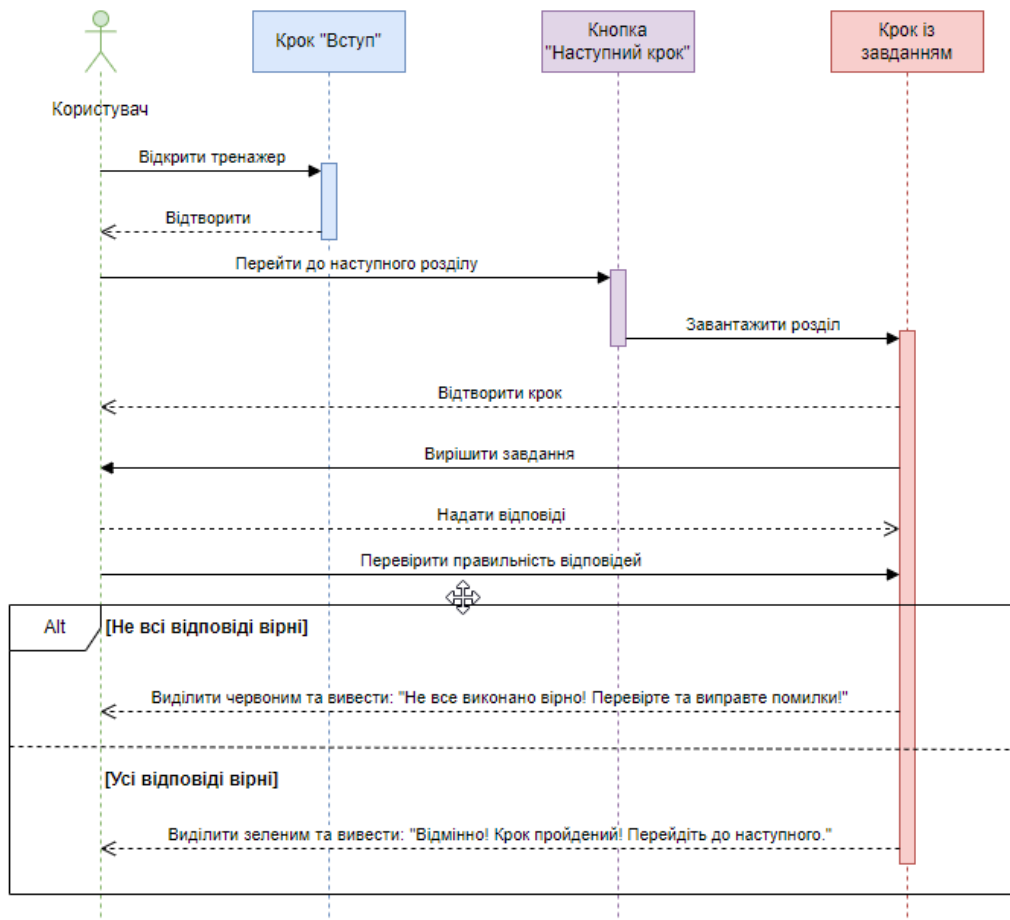


Рисунок 3.5 – Діаграма послідовності «Виконати завдання»

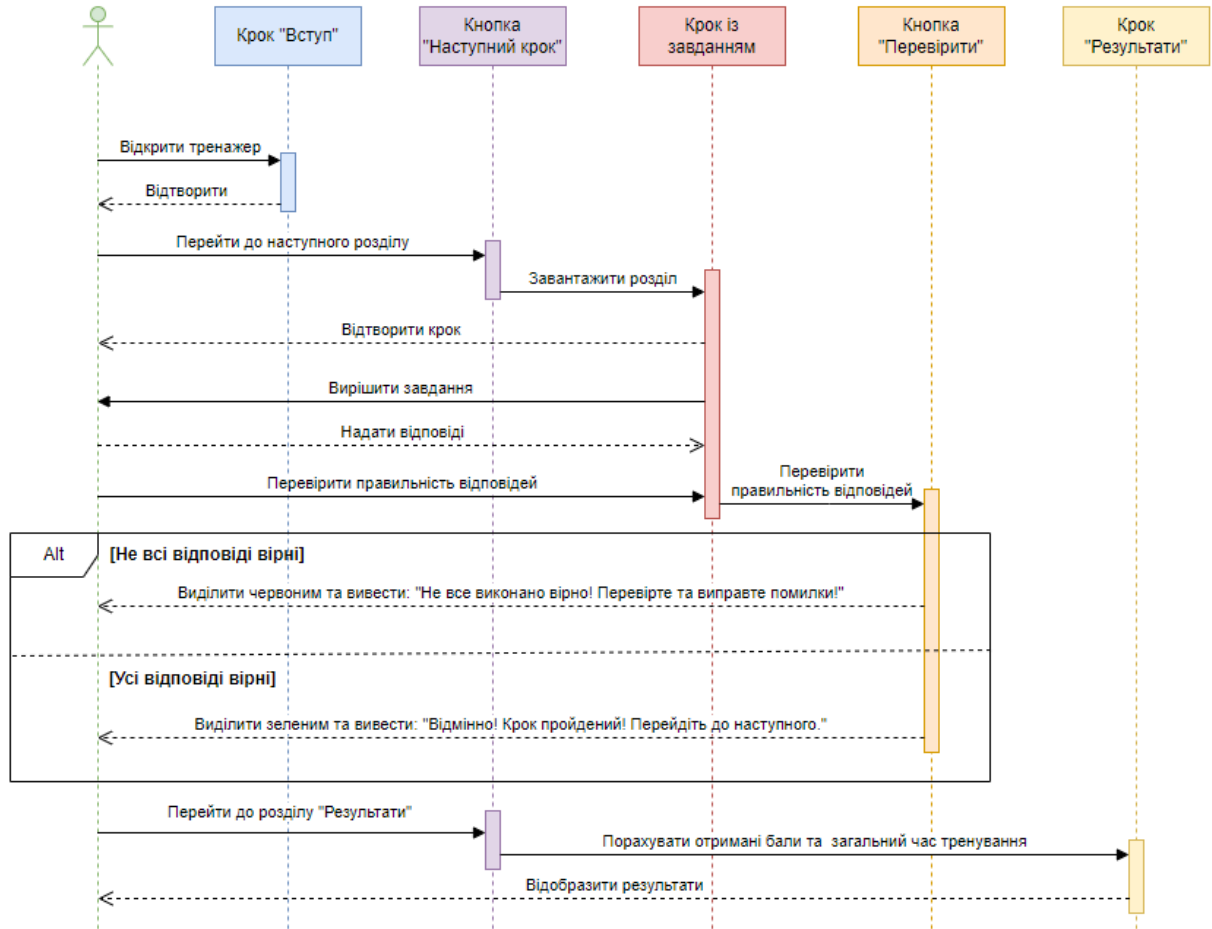


Рисунок 3.6 – Діаграма послідовності «Отримати результати»

Методології IDEF0 та UML дали можливість описати структуру інтерактивного тренажера. Скориставшись ними було візуально відображено функціонування системи, а саме: основний процес системи; його декомпозицію на підпроцеси; варіанти використання системи та послідовність взаємодії об'єктів у кожному з них.

## 4 ПРАКТИЧНА РЕАЛІЗАЦІЯ

### 4.1 Існуючі обмеження системи та перелік функціональних вимог

Інформаційна система, що розробляється в рамках даного проекту, призначена для використання викладачами та здобувачами освіти на єдиній навчальній платформі СумДУ. З її допомогою викладачі можуть надавати доступ до завдань веб-тренажеру студентам, а також перевіряти рівень засвоєння ними теоретичного матеріалу, а студенти – мають можливість проходити тренування за допомогою тренажера, удосконалювати свої практичні навички та переглядати результати тренувань, до яких також мають доступ викладачі.

Базовим фреймворком для розробки веб-тренажера було обрано фреймворк тренажерів Сумського державного університету jTrainier з метою спрощення процесу інтеграції додатку в оточення, легкої взаємодії з системою та можливістю працювати без збоїв. Проектування та реалізація здійснюються з урахуванням обмежень обраного фреймворку.

Повна версія серверного рушія записує результати тренувань студентів до бази даних та надає їм можливість отримати допомогу викладача у разі необхідності. Шаблон тренажеру складається з урізаної локальної версії серверного рушія та надає достатній функціонал для розробки.

Інтерактивний додаток матиме вступну частину; блок завдань, що складатиметься з чотирьох кроків; а також блок з результатами виконання завдань. Локалізація інтерфейсу проєкту передбачає два варіанта з використанням української та англійської мови. Кожен студент зможе обрати для себе найбільш зручний варіант на початку проходження тренажеру у вступній частині.

За виконання тренажеру передбачено оцінку в двадцять чотири бали.

Основна частина веб-тренажеру має чотири кроки. Кожен крок представляє собою окреме завдання.



На кожному кроці передбачається перевірка правильності виконання завдань. У разі надання користувачем невірних відповідей, вони виділяються червоним кольором, вірних - зеленим. За умови правильного виконання поточного кроку дозволяється перехід до наступного.

Під час виконання тренажеру не передбачаються штрафні бали за повторні спроби виконати завдання.

Тренажер повинен передбачати можливість зміни мови інтерфейсу за допомогою панелі управління мовним інтерфейсом: «Оберіть мову/Choose language».

У тренажері обов'язково повинні бути наступні кнопки:

- Перевірити/Check (для перевірки виконаного завдання).
- Інший варіант/Another variant (для зміни початкових вхідних даних).
- Допомога викладача/Teachers help (дає студенту можливість написати викладачу та отримати консультацію).
- Наступний крок /Next step.

При натисканні кнопки «Перевірити»:

- За умови, коли все виконано вірно на поточному кроці – виводиться на екран повідомлення «Відмінно! Крок пройдений! Перейдіть до наступного.»/ «Well done! The step done perfectly! Go to the next one.», активується кнопка «Наступний крок» та дозволяється перехід до виконання наступного кроку.

- За умови, коли не все виконано вірно на поточному кроці – перехід не дозволяється та виводиться на екран повідомлення «Не все виконано вірно! Перевірте та виправте помилки!» /«Oh snap! Check and fix errors! ».

## **4.2 Структура тренажера та розроблення окремих компонентів**

В поточному розділі детальніше ознайомимося зі структурою веб-тренажера «Перетворення чисел з однієї системи числення в іншу систему

числення», а також розглянемо етапи розробки основних його компонентів. Як було зазначено вище, для розробки обрано середовище Visual Studio Code.

Після відкриття папки з проектом у цьому редакторі можна ознайомитися зі структурою тренажера (як представлено на рисунку 4.1). Раніше було відзначено, що для розробки функціоналу тренажера використано локальну версію ядра, яке лежить на сервері СумДУ (папка `cdn`). Зміни в цю папку вносити не можна аби уникнути проблем та помилок у процесі перенесення веб-додатка на сервер.

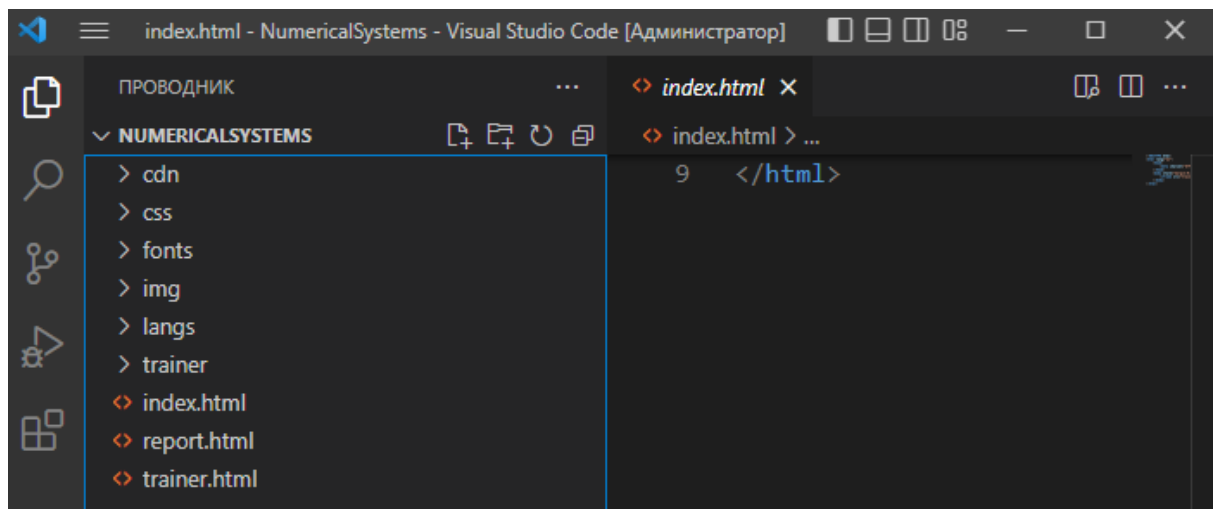


Рисунок 4.1 – Структура папки з веб-тренажером NumericalSystems

Папка `css` містить файл `trainer.custom.css` зі стилями до поточного додатку. У папці `fonts` містяться файли шрифтів. У папці для зображень `img` знаходиться логотип у файлі `NumericalSystems-logo.png`, а також малюнки для теоретичних блоків .

У каталозі `langs` знаходяться два JSON-файли, які описують всю текстову інформацію тренажера з українським та англійським варіантами локалізації інтерфейсу. Ці файли є текстовими осередками, що містять назви розділів, кнопок, тексти всіх питань, сповіщень і тексти теоретичних відомостей. За допомогою основного скрипта тренажера відбувається зчитування значень змінних із них та їх підстановка до основного файлу розмітки `trainer.html`. Приклад вмісту одного з цих файлів зображено на малюнку 4.2.

```

enjson x
langs > {} en.json > {} lang > STEP2_T8
31
32     "STEP1_T1": "<strong style='color:antiquewhite;'><u>Problem 1</u></strong>",
33     "STEP1_T2": "Convert the number",
34     "STEP1_T3": "from decimal numeral system (k=10) <br />to the binary numeral system (n=2) by",
35     "STEP1_T4": "<strong><em>Solution:</em></strong>",
36     "STEP1_T5": "Divide the number",
37     "STEP1_T6": "by 2.<br />The results of the division record in the table. In the field without",
38     "STEP1_T7": "The base<br />of the system",
39     "STEP1_T8": "The integer part<br />of the division",
40     "STEP1_T9": "Remainder<br />of division",
41     "STEP1_HINT": "<strong style='color:red;'>Hint:</strong><br />Write down<br />the number<br />.",
42     "STEP1_T10": "Enter the number you received in the binary numeral system:",
43     "STEP1_T11": "<br /><strong>Conversion of integers from the decimal numeral system to binary",
44     "STEP1_T12": "The conversion of an integer from the decimal numeral system to any other is c",
45     "STEP1_T13": "<strong>Table 1 – Conversion of a number from the decimal numeral system to bi",
46     "STEP1_T14": "<br />Write the code of the binary number in the direction of the arrow (that :
47
48     "STEP2_T1": "<strong style='color:antiquewhite;'><u>Problem 2</u></strong>",
49     "STEP2_T2": "Convert the number",
50     "STEP2_T3": "from binary numeral system (k=2) <br />to the decimal numeral system (n=10).",
51     "STEP2_T4": "<strong><em>Solution:</em></strong>",
52     "STEP2_T5": "Write down the numerical values into the corresponding gaps:",
53     "STEP2_T6": "Enter the number you received in the decimal numeral system:",
54     "STEP2_T7": "<br /><strong>Conversion of integers from binary numeral system to decimal nume",
55     "STEP2_T8": "<strong><em> The binary numeral system </em></strong>is a numeral system that u",
56     "STEP2_T9": "where a - is the alphabet of the numeral system, p - is the base of the numeral",
57     "STEP2_T10": "<strong>Table 2 – Table of Powers of Number 2</strong>",
58     "STEP2_T11": "<br /><em><u>Let's consider an example:</u></em> it is necessary to convert th",
59     "STEP2_T12": "As a result, we will get that the number 11101000<sub>2</sub> of the binary nu

```

Рисунок 4.2 – Файл локалізації інтерфейсу

У файлі `index.html` відбувається відкриття тренажеру у `iframe` (див. рисунок 4.3). Тег `<iframe>` (inline frame) – це тег-контейнер для автономної області на сторінці, в яку завантажуються самостійний документ, який визначається атрибутом `src`. У нашому випадку цим документом являється `trainer.html`, у якому підключаються усі скрипти тренажера, файли стилів, розмітка статичних частин тренажеру (хедер і футер додатку) та інше. За зв'язок із сервером відповідає файл `report.html`.

```

index.html x
index.html > html
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8" />
5   </head>
6   <body style="margin:0; padding:0; overflow:hidden;">
7     <iframe src="trainer.html" frameborder="0" style="position: absolute; height: 100%; height="100%" width="100%"></iframe>
8   </body>
9 </html>

```

Рисунок 4.3 – Файл `index.html`

Перелік файлів розміток, відповідні скрипти для всіх кроків тренажеру та файли налаштувань знаходяться у папці `trainer`. Структура цієї папки показана на рисунку 4.4. Усі файли, що пов'язані із завданнями містять у своїй назві слово «step». Кожне завдання складається із файла розмітки та відповідного файла скрипта (наприклад, файли першого завдання – `step1.html` та `step1.js`). Скрипт кожного завдання тренажера додає потрібні елементи керування у відповідний файл розмітки та містить у собі функції-обробники подій, завдяки яким генеруються випадкові числа, будуються таблиці, здійснюються перевірки та з'являються або приховуються елементи на сторінці.

Каталог `settings`, що являється структурною частиною каталогу `trainer`, містить JSON-файли налаштувань. У файлі `trainer.config.json` знаходиться інформація про можливі мови (українська та англійська), мову запуску інтерфейсу за замовчуванням (українська), максимальну кількість відсотків за тренування (100%), дані про розробника та автора курсу, статус (тестування чи готовий до завантаження у систему). А у файлі `trainer.steps.json` відбувається розподіл відсотків за виконання кожного кроку. У рамках даного проекту відсотки розподіляються порівну на кожне з чотирьох завдань (по 25%).

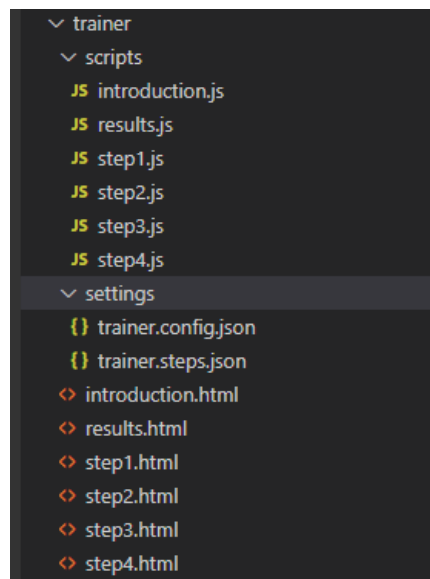


Рисунок 4.4 – Структура папки `trainer`

Лістинг файлів інтерактивного тренажера наведено у додатку А.

### 4.3 Програмна реалізація веб-тренажера

Процес створення веб-тренажера передбачає розробку логотипу для нього, реалізацію вступної частини, чотирьох розділів із завданнями та теоретичними блоками, а також розділу, в якому виводяться результати тренування.

Для вступної частини тренажеру був створений файл розмітки introduction.html, в якому прописані назва тренажеру, його логотип, кількість кроків із завданнями та панель мовного інтерфейсу (див. рисунок 4.5). Шаблонізатор Mustache замість записів у фігурних дужках підставляє значення ключів з JSON-файлів (ua.json або en.json в залежності від обраної користувачем локалізації).

```

1 <div class="row jumbotron introduction">
2   <div class="col-md-6 animated zoomIn">
3     
4   </div>
5   <div class="col-md-6 animated fadeInDown">
6     <h1>{{TRAINER_NAME}}</h1>
7
8     <p>
9       <h5><strong>{{STEP_COUNT}}:</strong> <span class="badge">{{STEPS_COUNT}}</span></h5>
10      <br/>
11      <h5>{{CHOOSE_LANG}}:</h5>
12      <div class="flag-icon flag-icon-ua" title="ua" id="ua"></div>
13      <div class="flag-icon flag-icon-en" title="en" id="en"></div>
14    </p>
15  </div>
16 </div>

```

Рисунок 4.5 – Файл introduction.html

Скриптом кроку «Вступ» (introduction.js) для всіх наступних завдань визначаються змінні, а також прописуються функції (makestep1()-makestep4()), які генерують випадкове число та динамічно будують html-елементи для виконання кожного кроку, а також функції перевірки проходження кроків. Для прикладу розглянемо детальніше код функції makestep1(), показаної на рисунку 4.6.

```

var a1 = 0, a1_bin = 0, a1_bin_back = 0, a1_even = [];
function makestep1() {
    a1 = Math.floor(Math.random() * (50-5+1)+5);

    a1_bin = a1.toString(2);
    a1_bin_back = a1_bin.split("").reverse().join("");

    let n = a1;
    a1_even = [];
    while(n > 0) {
        if(n % 2 === 0) {
            n = n / 2;
            a1_even.push(n);
        } else {
            n = n - 1;
            n = n / 2;
            a1_even.push(n);
        }
    }

    let str1 = '<tr><th>'+I18N.getConstants()
['STEP1_T7']+'</th><th>'+I18N.getConstants()
['STEP1_T8']+'</th><th>'+I18N.getConstants()['STEP1_T9']+'</th><td
id="hints1" rowspan="'+(a1_bin.length + 2)+'"><span
class="showafter">'+I18N.getConstants()
['STEP1_HINT']+'</span></td></tr><tr><td>2</td><td>'+a1+'</td><td><div
class="form-group" for="sli1"><input name="sli1" class="form-control"
type="text" placeholder="?" /></div></td></tr>';
    for(let i=0; i<(a1_bin.length * 3); i+=3) {
        str1 += '<tr><td><div class="form-group" for="sli'+(i+2)+'"><input
name="sli'+(i+2)+'" class="form-control" type="text" placeholder="?"
/></div></td><td><div class="form-group" for="sli'+(i+3)+'"><input
name="sli'+(i+3)+'" class="form-control" type="text" placeholder="?"
/></div></td><td><div class="form-group" for="sli'+(i+4)+'"><input
name="sli'+(i+4)+'" class="form-control" type="text" placeholder="?"
/></div></td></tr>';
    }
    $('#page1 table').append(str1);
    $('#page1 input[type="text"]').attr('autocomplete', 'off');
}

```

Рисунок 4.6 – Функція makestep1()

Перший рядок коду оголошує глобальні змінні:

- a1 (зберігає випадкове десяткове число в діапазоні від 5 до 50);
- a1\_bin (зберігає двійкове подання числа a1);
- a1\_bin\_back (зберігає зворотне двійкове подання числа a1);
- a1\_even (зберігає масив чисел, отриманих в процесі ділення числа a1 на 2).

Функція makestep1():

- генерує випадкове число  $a1 = \text{Math.floor}(\text{Math.random()} * (50 - 5 + 1) + 5)$ ;

- переводить число в двійкову систему `a1_bin = a1.toString(2)` та зберігає зворотнє двійкове подання числа `a1_bin_back = a1_bin.split("").reverse().join("");`
- формує масив чисел `a1_even = []`;
- генерує html-код таблиці (зберігає його у змінній `str1`);
- додає таблицю в html-сторінку першого завдання `$('.page1 table').append(str1);`
- відключає автозаповнення для текстових полів `$('.page1 input[type="text"]').attr('autocomplete', 'off');`

Наступним етапом стала розробка практичних завдань тренажера.

Для реалізації сторінки «Крок 1» були розроблені файл розмітки `step1.html` та скрипт `step1.js` (див. рисунки 4.7-4.8).

При завантаженні кроку 1 викликається і обробляється функція `makestep1()`. Випадково згенероване число `a1` за допомогою виразу `$('.page1 .a1s1').html(a1)`; підставляється в усі елементи розмітки з класом `.a1s1`.

```

1 <div class="jumbotron page1">
2   <div class="cond">
3     <h5>{{{STEP1_T1}}}</h5>
4     <p>{{{STEP1_T2}}} <span class="a1s1"></span><sub>10</sub> {{{STEP1_T3}}}</p>
5   </div>
6   <p>{{{STEP1_T4}}}</p>
7   <p>{{{STEP1_T5}}} <span class="a1s1"></span><sub>10</sub> {{{STEP1_T6}}}</p>
8   <table align="center" border="3"></table>
9
10  <div class="showafter">
11    <p>{{{STEP1_T10}}}</p>
12    <span class="text">
13      <span class="a1s1"></span><sub>10</sub> = {{{STEP1_INPUT}}} <sub>2</sub>
14    </span>
15  </div>
16
17  <div style="text-align: center; margin-top: 5px;">
18    <button class="btn btn-success check">{{{CHECK}}}</button>
19    <button class="btn btn-default shuffle">{{{SHUFFLE}}}</button>
20  </div>
21
22  <div class="container">
23    <hr>
24    <div id="hidden-theory">
25      <h5>{{{STEP1_T11}}}</h5>
26      <p class="theory">{{{STEP1_T12}}}</p>
27      <p>{{{STEP1_T13}}}</p>
28      
29      <p class="theory">{{{STEP1_T14}}}</p>
30    </div>
31
32    <div style="text-align: center; margin-top: 5px;">
33      <button class="btn btn-primary theory">{{{REFRESH_YOUR_THOUGHTS}}}</button>
34    </div>
35    <hr>
36  </div>
37 </div>

```

Рисунок 4.7 – Файл `step1.html`

Після введення користувачем відповідей і натискання кнопки «Перевірити» спрацьовує функція `checkinput1()` призначена для перевірки відповідей користувача. Вона порівнює введені користувачем відповіді з правильними відповідями і повертає результат перевірки.

```

1  var VStep1;// A variable for future validator
2  var step1 = function () {
3      this.preDispatch = function () {
4          };
5      this.postDispatch = function () {
6          $('#endTrainer').removeAttr('disabled').removeClass('disabled');
7          makestep1();
8          $('#page1 .als1').html(a1);
9          function checkinput1() {
10             let corans = [], corinp = [];
11             let inpnum = $('#page1 table input').length;
12             let rownum = $('#page1 table tr').length;
13             corans.push(a1_bin_back[0]);
14             corinp.push('s1i1');
15             for(let i=0, k=1; i<(rownum - 2), k<(rownum - 2); i+=3, k++) {
16                 corans.push('2');
17                 corinp.push('s1i'+(2+i));
18                 corans.push(a1_even[k - 1].toString());
19                 corinp.push('s1i'+(3+i));
20                 corans.push(a1_bin_back[k]);
21                 corinp.push('s1i'+(4+i));
22             }
23             corans.push('-');
24             corinp.push('s1i'+(inpnum - 2));
25             corans.push('0');
26             corinp.push('s1i'+(inpnum - 1));
27             corans.push('-');
28             corinp.push('s1i'+inpnum);
29             let corstep1 = 0;
30             for(let n=0; n<inpnum; n++) {
31                 if($('#input[name="'+corinp[n]+'"]').val() === corans[n]) {
32                     corstep1++;
33                     $('#div[for="'+corinp[n]+'"]').removeClass('has-error').addClass('has-success');
34                 } else {
35                     $('#div[for="'+corinp[n]+'"]').removeClass('has-success').addClass('has-error');
36                 }
37             }
38             if(corstep1 === inpnum) {
39                 return true;
40             } else {
41                 return false;
42             }
43         }
44     }
45 }

```

Рисунок 4.8 – Елемент файлу `step1.js`

Розберемо детальніше код функції `checkinput1()`. Спочатку відбувається ініціалізація чотирьох змінних:

- `corans` (масив для зберігання правильних відповідей);
- `corinp` (масив для зберігання імен полів введення);
- `inpnum` (кількість полів введення в таблиці);
- `rownum` (кількість рядків в таблиці).

Далі масив `corans` заповнюється правильними відповідями, які повинні бути введені в поля введення.



Потім відбувається перевірка відповідей користувача. Перебираються всі поля введення в таблиці і введені відповіді порівнюються з правильними відповідями з масиву `corans`. Якщо відповідь правильна, поле введення позначається зеленим кольором; якщо неправильна - червоним.

Наступним кроком змінна `corstep1` підраховує кількість правильних відповідей. Якщо всі відповіді правильні (`corstep1 === inprnum`), функція повертає `true`. Якщо є хоча б одна неправильна відповідь, функція повертає `false`.

Кнопка «Інший варіант» надає можливість користувачеві змінити випадково згенероване десяткове число на інше випадкове число. Скрипт, який її описує, зображено на рисунку 4.9.

```

$( '.page1 button.shuffle' ).click(function(){
    $('input[name="step1-input"]').val('');
    $(' .page1 table' ).html('');
    makestep1();

    $(' .page1 .a1s1' ).html(a1);
    $(' .page1 div.form-group' ).removeClass('has-error').removeClass('has-success');
    $(' .page1 .showafter' ).css('display', 'none');
    $(' .alert-danger' ).hide();
});

```

Рисунок 4.9 – Функціонал кнопки «Інший варіант» із `step1.js`

Розмітка вбудованого блоку теорії показана на рисунку 4.10, а функціонал кнопки «Освіжи думки» – на рисунку 4.11.

```

<div class="container">
  <hr>
  <div id="hidden-theory">
    <h5>{{{STEP1_T11}}}</h5>
    <p class="theory">{{{STEP1_T12}}}</p>
    <p>{{{STEP1_T13}}}</p>
    
    <p class="theory">{{{STEP1_T14}}}</p>
  </div>

  <div style="text-align: center; margin-top: 5px;">
    <button class="btn btn-primary theory" >{{REFRESH_YOUR_THOUGHTS}}</button>
  </div>
  <hr>
</div>

```

Рисунок 4.10 – Розмітка вбудованого блоку теорії на «Кроці 1»

```

$('.page1 button.theory').click(function(){
    if($('.page1 div#hidden-theory').css('display') == 'none') {
        $('.page1 div#hidden-theory').css('display', 'block');
        $(this).text(I18N.getConstants()['OK']);
        $('.alert-danger').hide();
        $('.alert-success').hide();
    } else {
        $('.page1 div#hidden-theory').css('display', 'none');
        $(this).text(I18N.getConstants()['REFRESH_YOUR_THOUGHTS']);
    }
});

```

Рисунок 4.11 – Функціонал кнопки «Освіжи думки» із step1.js

Функціонал кнопки «Перевірити» представлено на рисунку 4.12. Передбачено перевірку в два етапи. При першому натисканні кнопки перевіряється правильність введених значень в таблицю. Якщо результат вірний користувачеві виводиться поле для остаточного вводу відповіді і підказка. При повторному натисканні за умови правильності введених відповідей дезактивуються кнопки «Перевірити», «Інший варіант» та «Завершити сеанс» та активується кнопка «Наступний крок» для переходу до виконання наступного завдання.

```

$('.page1 button.check').click(function () {
    if($('.page1 div.showafter').css('display') == 'none') {
        if(checkinput1()) { $('.page1 .page1 td#hints1').css('display', 'block');
            $('.page1 .showafter').css('display', 'block');
            $('.page1 input[name="step1-input"]').focus();
        } else {
            $('.alert-danger').show();
        }
    } else {
        if($('.input[name="step1-input"]').val() !== a1_bin) {
            $('.alert-danger').show();
            $('div[for="step1-input"]').removeClass('has-success').addClass('has-error');
        } else {
            step1check = true;
            $('div[for="step1-input"]').removeClass('has-error').addClass('has-success');
            $('.alert-success').show();
            $('.alert-danger').hide();
            Scorer.addScore(25);
            $('.page1 .check, .page1 .shuffle, #endTrainer').attr('disabled', 'disabled').addClass('disabled');
            Rotator.enableNextButton();
        }
    }
});

```

Рисунок 4.12 – Функціонал кнопки «Перевірити» із step1.js

На другому кроці користувач повинен перевести випадково згенероване число двійкової системи числення у десяткову систему. Для цього за допомогою функції `makestep2()` генерується випадкове двійкове число та формується шаблон многочлена, який динамічно вбудовується у файл розмітки кроку 2. Код цієї функції представлено на рисунку 4.13.

```
var a2 = 0, a2_bin = 0;
function makestep2() {
  a2 = Math.floor(Math.random() * (50-5+1)+5);
  a2_bin = a2.toString(2);

  let str2 = '<noobr><span class="text">'+a2_bin+'<sub>2</sub> = </span>';
  for(let i=0; i<(a2_bin.length * 2); i+=2) {
    str2 += '<div class="form-group" for="s2i'+(i+1)+'"><input name="s2i'+(i+1)+'" class="form-control odd-input" type="text" placeholder="?" /></div><span class="text">&middot; 2</span> <div class="form-group" for="s2i'+(i+2)+'"><input name="s2i'+(i+2)+'" class="form-control even-input" type="text" placeholder="?" /></div>';
    if(i != (a2_bin.length * 2 - 2)) {
      str2 += '<span class="text">+</span> ';
    }
  }
  str2 += '</noobr>';
  $('<div>.page2 .step2_end').append(str2);
  $('<div>.page2 input[type="text"]').attr('autocomplete', 'off');
}
```

Рисунок 4.13 – Функція `makestep2()`

На цьому кроці також передбачено двоетапна перевірка. Перше натискання кнопки «Перевірити» перевіряє правильність введених значень у многочлен. При позитивному результаті виводиться поле для остаточного вводу відповіді. Повторне натискання за умови правильності введених відповідей дозволяє перехід до наступного кроку.

Завдання 3 та 4 полягають у перетворенні випадкового двійкового числа у вісімкову та шістнадцяткову системи числення відповідно. Тому реалізацію обох кроків розглянемо на прикладі «Кроку 3». Код функції `makestep3()`, що генерує випадкове двійкове число та формує шаблон тріад, наведено на рисунку 4.14.

Перевірка цього кроку працює наступним чином. Перше натискання кнопки «Перевірити» перевіряє правильність введених значень у тріади. При правильному заповненні повторно виводить заповнені тріади з комірками для введення вісімкового числа, а також поле для остаточного вводу відповіді.

Правильне заповнення усіх полів для вводу та повторне натискання кнопки дозволяє перехід до наступного кроку.

```

var a3 = 0, a3_bin = 0, a3_oct = 0, a3_bin_num = 0, a3_bin_add = 0;
function makestep3() {
  a3 = Math.floor(Math.random()*(89-10+1)+10);
  a3_bin = a3.toString(2);
  a3_oct = a3.toString(8);
  a3_bin_num = a3_bin.length;
  a3_bin_add = a3_bin;

  while(a3_bin_num % 3 !== 0) {
    a3_bin_num++;
    a3_bin_add = '0'+a3_bin_add;
  }

  let str31 = '<tr>';
  for(let i=0; i<a3_bin_num; i++) {
    if((i+1) == 4 || (i+1) == 7) {
      str31 += '<td class="space"></td>';
    }
    str31 += '<td><div class="form-group" for="s31i'+(i+1)+'"><input name="s31i'+(i+1)+'"
class="form-control" type="text" placeholder="?" /></div></td>';
  }
  str31 += '</tr>';
  $('#page3 .part31').append(str31);

  let str32 = '<tr>';
  for(let i=0; i<a3_bin_num; i++) {
    if((i+1) == 4 || (i+1) == 7) {
      str32 += '<td class="space"></td>';
    }
    str32 += '<td class="triads">'+a3_bin_add[i]+'</td>';
  }
  str32 += '</tr><tr>';
  for(let k=0; k<(a3_bin_num / 3); k++) {
    if(k == 1 || k == 2) {
      str32 += '<td class="space" style="border-left: 3px solid #d8dce0;border-right:
3px solid #d8dce0;"></td>';
    }
    str32 += '<td colspan="3"><span class="text">&#8681;</span></td>';
  }
  str32 += '</tr><tr>';
  for(let n=0; n<(a3_bin_num / 3); n++) {
    if(n == 1 || n == 2) {
      str32 += '<td class="space" style="border-left: 3px solid #d8dce0;border-right:
3px solid #d8dce0;"></td>';
    }
    str32 += '<td colspan="3"><div class="form-group" for="s32i'+(n+1)+'"><input
name="s32i'+(n+1)+'" class="form-control" type="text" placeholder="?" /></div></td>';
  }
  str32 += '</tr>';
  $('#page3 .part32').append(str32);
  $('#page3 input[type="text"]').attr('autocomplete', 'off');
}

```

Рисунок 4.14 – Функція makestep3()

У разі успішного виконання кожного кроку користувачеві нараховується 25% за кожен і він відправляється на сторінку результатів, розмітка якої зображена на рисунку 4.15.

```

<div class="row jumbotron results">
  <div class="cond">
    <h1>{{GREETING}}</h1>
  </div>

  <div class="col-md-6 animated zoomIn">
    
  </div>

  <div class="col-md-6">
    <h3>{{RESULTS_TITLE}}</h3>
    <div class="container">
      <div class="chart" id="gc"></div>
    </div>
    <p>
      <h4>{{RESULTS_START_TIME}}: {{START_TIME}}</h4>
      <h4>{{RESULTS_END_TIME}}: {{END_TIME}}</h4>
      <h4>{{RESULTS_TIME_DIFF}}: <span class="badge">{{TIME_DIFF}} {{RESULTS_TIME_SEC}}</span></h4>
    <br/>
      <h3>{{RESULTS_YOUR_SCORE}}: {{RESULTS_POINTS_IN_PERCENT}}%</h3>
    </p>
  </div>
</div>

```

Рисунок 4.15 – Файл results.html

#### 4.4 Результати розроблення та тестування додатку

В результаті створення веб-тренажера реалізовано його логотип, вступну частину, чотири розділи із завданнями та теоретичними блоками, а також розділ з результатами тренування.

Для запуску тренажера локально використаємо розширення для Visual Studio Code Live Server.

Логотип додатку створювався на онлайн платформі проєктування графіки Canva. Зображення у повній мірі пасує обраній тематиці та гармонійно виглядає на стартовій сторінці.

Зовнішній вигляд вступної частини зображено на рисунку 4.16. У даному випадку користувач може обрати зручніший для себе варіант тренування: українською або англійською мовою. Українська мова є мовою тренажера за замовчуванням. Також містяться чотири кнопки: «Попередній крок» (неактивна), «Допомога викладача», «Завершити сеанс» (неактивна) та «Наступний крок». Перша та остання кнопки забезпечують навігацію між кроками.



Рисунок 4.16 – Розділ «Вступ»

За допомогою кнопки «Допомога викладача» користувач може задати питання викладачеві. А натискання кнопки «Завершити сеанс» зупиняє тренування. При цьому зберігаються зароблені за попередні кроки бали, визначається крок, на якому сеанс переривається, а також перевіряється чи виконані умови для завершення поточного кроку. Якщо умови виконані *true* (для змінних *step1check*, *step2check*, *step3check* або *step4check*), дозволяється перехід до сторінки з результатами тренування за допомогою функції `Rotator.nextResult()`.

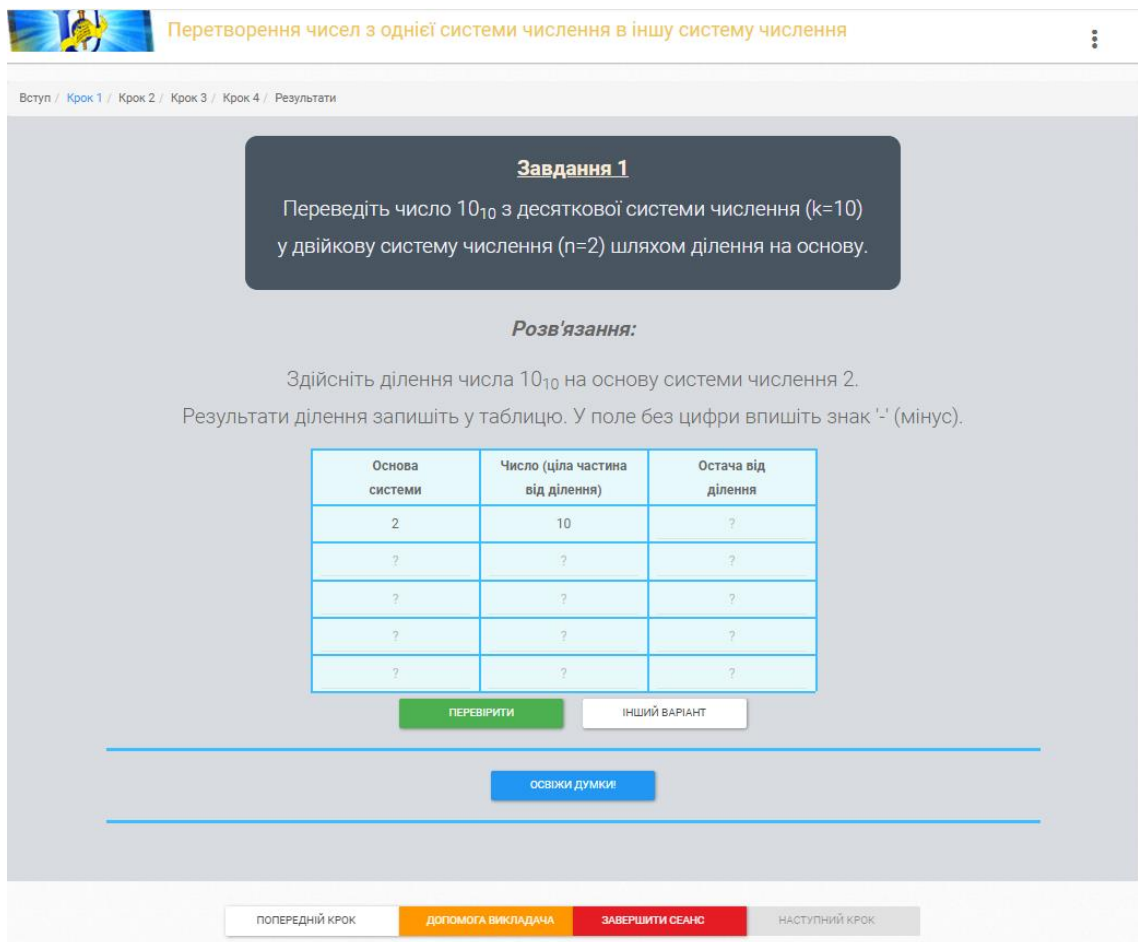
Якщо умови не виконані *false*, перевіряються відповіді користувача (імітується натискання кнопки «Перевірити»), додається 0 до рахунку за допомогою функції `Scorer.addScore(0)` та дозволяється перехід до результатів.

Кнопка «Наступний крок» направляє користувача на перше практичне завдання під назвою «Крок 1» (див. рисунок 4.17).

Суть завдання полягає у тренуванні навичок переведення випадково згенерованого десяткового числа (від 5 до 50) у двійкове шляхом ділення на основу двійкової системи числення – 2. Сторінка містить безпосередньо саме завдання, короткі інструкції для його виконання, таблицю для заповнення,

кнопки «Перевірити» та «Інший варіант», а також прихований теоретичний блок, який відкривається при натисканні кнопки «Освіжити думки!».

Таблиця має три колонки та випадкову кількість рядків в залежності від згенерованого десяткового числа. У першому рядку автоматично підставляються значення у першу комірку – цифра 2 (як основа двійкової системи) та у другу комірку – випадково згенероване десяткове число. Інші комірки студент повинен заповнити самостійно. У разі внесення студентом невірних значень до будь-якої з комірок при натисканні кнопки «Перевірити» хибні відповіді виділяються червоним кольором та виводиться сповіщення «Не все виконано вірно! Перевірте та виправте помилки!», як це показано на рисунку 4.18.



**Перетворення чисел з однієї системи числення в іншу систему числення**

Вступ / Крок 1 / Крок 2 / Крок 3 / Крок 4 / Результати

**Завдання 1**  
Переведіть число  $10_{10}$  з десяткової системи числення ( $k=10$ ) у двійкову систему числення ( $n=2$ ) шляхом ділення на основу.

**Розв'язання:**  
Здійсніть ділення числа  $10_{10}$  на основу системи числення 2. Результати ділення запишіть у таблицю. У поле без цифри впишіть знак '-' (мінус).

Основа системи	Число (ціла частина від ділення)	Остача від ділення
2	10	?
?	?	?
?	?	?
?	?	?
?	?	?

ПЕРЕВІРИТИ    ІНШИЙ ВАРІАНТ

ОСВІЖИ ДУМКИ

ПОПЕРЕДНІЙ КРОК    ДОПОМОГА ВИКЛАДАЧА    ЗАВЕРШИТИ СЕАНС    НАСТУПНИЙ КРОК

Рисунок 4.17 – Розділ «Крок 1»

Перетворення чисел з однієї системи числення в іншу систему числення

Вступ / Крок 1 / Крок 2 / Крок 3 / Крок 4 / Результати

**Завдання 1**

Переведіть число  $10_{10}$  з десяткової системи числення ( $k=10$ ) у двійкову систему числення ( $n=2$ ) шляхом ділення на основу.

**Розв'язання:**

Здійсніть ділення числа  $10_{10}$  на основу системи числення 2. Результати ділення запишіть у таблицю. У поле без цифри впишіть знак '-' (мінус).

Не все виконано вірно! Перевірте та виправте помилки!

системи	від ділення)	ділення
2	10	1
2	5	1
2	2	0
2	1	1
-	0	-

ПЕРЕВІРИТИ    ІНШИЙ ВАРІАНТ

ОСВЯЖИ ДУМКУ

ПОПЕРЕДНІЙ КРОК    ДОПОМОГА ВИКЛАДАЧА    ЗАВЕРШИТИ СЕАНС    НАСТУПНИЙ КРОК

Рисунок 4.18 – Виконання завдання 1 з неправильними відповідями

Коли користувач внесе правильні відповіді та натисне кнопку «Перевірити», з'являється додаткова підказка та поле для вводу остаточної відповіді. При її введенні та повторному натисканні кнопки «Перевірити» у випадку правильної відповіді з'являється повідомлення «Відмінно! Крок пройдений! Перейдіть до наступного», дезактивуються кнопки «Перевірити», «Інший варіант» та «Завершити сеанс», а також активується кнопка «Наступний крок», що дає можливість перейти до виконання наступного завдання (рис. 4.19).



Перетворення чисел з однієї системи числення в іншу систему числення

Вступ / Крок 1 / Крок 2 / Крок 3 / Крок 4 / Результати

**Завдання 1**

Переведіть число  $10_{10}$  з десяткової системи числення ( $k=10$ ) у двійкову систему числення ( $n=2$ ) шляхом ділення на основу.

**Розв'язання:**

Здійсніть ділення числа  $10_{10}$  на основу системи числення 2. Результати ділення запишіть у таблицю. У поле без цифри впишіть знак  $^{-1}$  (мінус).

Відмінної! Крок пройдений! Перейдіть до наступного.

системи	від ділення)	ділення
2	10	0
2	5	1
2	2	0
2	1	1
-	0	-

**Підказка:**  
Записувати число знизу-вгору

Введіть число, яке отримали в двійковій системі числення:

$10_{10} = \underline{1010}_2$

ПЕРЕВІРИТИ    ІНШИЙ ВАРІАНТ

ОСВІЖИ ДУМКИ

ПОПЕРЕДНІЙ КРОК    ДОПОМОГА ВИКЛАДАЧА    ЗАВЕРШИТИ СЕАНС    НАСТУПНИЙ КРОК

Рисунок 4.19 – Виконання завдання з правильними відповідями

Натискання користувачем кнопки «Освіжи думки!» призводить до розкриття прихованого теоретичного блоку, призначеного допомогти студенту згадати принцип виконання поточного завдання (див. рисунок 4.20). В цей момент кнопка змінює свою назву на «ОК!!!», а під час закриття блоку з теорією знову змінюється на «Освіжи думки!».

Основною метою наступного «Кроку 2» є закріплення теоретичного матеріалу та тренування практичних навичок у вправі по переведенню випадково згенерованого числа двійкової системи числення у десяткову систему.

### Перетворення цілих чисел з десятикової системи числення в двійкову методом ділення

Переведення цілого числа з десятикової системи числення у будь-яку іншу здійснюється шляхом послідовного ділення числа на основу нової системи числення. Ділення виконується доти, поки остання частка не стане менше дільника. Отримані остачі від ділення, взяті у зворотному порядку, будуть значеннями розрядів числа в новій системі числення. Остання частка є самим старшим розрядом двійкового числа.

Для розуміння, як працює метод ділення для перетворення з десятикової системи числення в двійкову *розглянемо наступний приклад*: перетворимо число  $38_{10}$ . Проведемо ділення числа  $38_{10}$  на основу двійкової системи числення 2, а результати запишемо в таблицю 1.

Таблиця 1 – Перетворення числа з десятикової системи числення у двійкову

Основа системи/ The base of the system	Число (ціла частка від ділення)/ The integer part of the division	Остача від ділення/ Remainder of division
2	38	0
2	19	1
2	9	1
2	4	0
2	2	0
2	1	1
-	0	-



Код двійкового числа запишемо за напрямком стрілки (тобто знизу вгору). Таким чином, десятиковому числу  $38_{10}$  відповідає двійковий еквівалент  $100110_2$ . Правильність результату можна перевірити переведенням двійкового числа в десятикову систему.

ОК!!

Рисунок 4.20 – Теоретичний блок Кроку 1

Сторінка даного кроку також має завдання, короткі інструкції для його виконання, кнопки «Перевірити» та «Інший варіант», а також теоретичний блок по темі другого завдання із кнопкою «Освіжити думки». Проте замість таблиці формується шаблон многочлена відповідно до згенерованого двійкового числа, що складається з творів цифр цього числа і відповідного ступеня двійки. У лівій частині рівності знаходиться рандомно згенероване двійкове число, а у правій – поля, які пропонується заповнити користувачеві (див. рисунок 4.21).

The screenshot shows a web application interface for a math task. At the top, there is a progress bar labeled 'Перетворення в іншу систему числення' with a 67% completion indicator and a 'Скинути' button. Below the progress bar, a breadcrumb trail reads 'Вступ / Крок 1 / Крок 2 / Крок 3 / Крок 4 / Результати'. The main content area is titled 'Завдання 2' and contains the text: 'Переведіть число  $11011_2$  з двійкової системи числення ( $k=2$ ) у десяткову систему числення ( $n=10$ ).'. Below this, the instruction 'Розв'язання:' is followed by 'Запишіть у пропуски відповідні числові значення:'. The equation  $11011_2 = ? \cdot 2^4 + ? \cdot 2^3 + ? \cdot 2^2 + ? \cdot 2^1 + ? \cdot 2^0$  is displayed. There are two buttons: a green 'ПЕРЕВІРИТИ' button and a white 'ІНШИЙ ВАРІАНТ' button. Below these is a blue 'ОСВІЖИ ДУМКИ!' button. At the bottom of the interface, there are four navigation buttons: 'ПОПЕРЕДНІЙ КРОК' (disabled), 'ДОПОМОГА ВИКЛАДАЧА' (highlighted in orange), 'ЗАВЕРШИТИ СЕАНС' (highlighted in red), and 'НАСТУПНИЙ КРОК' (disabled).

Рисунок 4.21 – Розділ «Крок 2»

При внесенні користувачем правильних записів у многочлен та натисканні кнопки «Перевірити» з'являється приховане поле для вводу остаточної відповіді. Введення правильної відповіді та повторне натискання кнопки «Перевірити» викликає повідомлення «Відмінно! Крок пройдений! Перейдіть до наступного», дезактивує кнопки «Перевірити», «Інший варіант», «Завершити сеанс» та активує кнопку «Наступний крок». Це дає змогу користувачеві перейти до виконання третього завдання.

Негативні і позитивні результати виконання другого завдання зображені на рисунках 4.22 – 4.23 відповідно.

Кроки 3 та 4 дуже схожі по своїй суті. В обох випадках завдання полягає у перетворенні випадкового двійкового числа у вісімкову та шістнадцяткову системи числення відповідно. Спосіб перетворення двійкового числа у вище названі системи числення шляхом розбиття на тріади та тетради відповідно пов'язаний зі ступенями двійки ( $2^3$  та  $2^4$ ).

Перетворення в іншу систему числення 80%

Вступ / Крок 1 / Крок 2 / Крок 3 / Крок 4 / Результати

**Завдання 2**

Переведіть число  $11011_2$  з двійкової системи числення ( $k=2$ ) у десяткову систему числення ( $n=10$ ).

Не все виконано вірно! Перевірте та виправте помилки!

Запишіть у пропуски відповідні числові значення:

$$11011_2 = \underline{1} \cdot 2^{\underline{4}} + \underline{0} \cdot 2^{\underline{3}} + \underline{0} \cdot 2^{\underline{2}} + \underline{1} \cdot 2^{\underline{1}} + \underline{1} \cdot 2^{\underline{0}}$$

ПЕРЕВІРИТИ      ІНШИЙ ВАРІАНТ

ОСВІЖИ ДУМКИ!

ПОПЕРЕДНІЙ КРОК    ДОПОМОГА ВИКЛАДАЧА    ЗАВЕРШИТИ СЕАНС    НАСТУПНИЙ КРОК

Рисунок 4.22 – Виконання завдання 2 з неправильними відповідями

Перетворення в іншу систему числення 67%

Вступ / Крок 1 / Крок 2 / Крок 3 / Крок 4 / Результати

**Завдання 2**

Переведіть число  $11011_2$  з двійкової системи числення ( $k=2$ ) у десяткову систему числення ( $n=10$ ).

*Розв'язання:*

Відмінно! Крок пройдений! Перейдіть до наступного.

$$11011_2 = \underline{1} \cdot 2^{\underline{4}} + \underline{1} \cdot 2^{\underline{3}} + \underline{0} \cdot 2^{\underline{2}} + \underline{1} \cdot 2^{\underline{1}} + \underline{1} \cdot 2^{\underline{0}}$$

Введіть число, яке отримали в десятковій системі числення:

$$11011_2 = \underline{27} \cdot 10$$

ПЕРЕВІРИТИ      ІНШИЙ ВАРІАНТ

ОСВІЖИ ДУМКИ!

ПОПЕРЕДНІЙ КРОК    ДОПОМОГА ВИКЛАДАЧА    ЗАВЕРШИТИ СЕАНС    НАСТУПНИЙ КРОК

Рисунок 4.23 – Виконання завдання 2 з правильними відповідями

Обидві сторінки містять завдання, короткі інструкції для їхнього виконання, вказівку, підказку, а також блоки тріад/тетрад відповідно для заповнення користувачем і вже стандартно кнопки «Перевірити», «Інший варіант» та «Освіжи думки!». При натисканні кнопки «Інший варіант» користувач може змінити випадково згенероване двійкове число на інше випадкове число. Завдання третє зображено на рисунку 4.24.

Рисунок 4.24 – Розділ «Крок 3»

У разі вірного заповнення тріад/тетрад і натискання кнопки «Перевірити» відкривається друга частина завдання, де сформовані та заповнені тріади/тетради та є поля для внесення перетворених у вісімкову/шістнадцяткову систему чисел, а також поле для введення остаточної відповіді. Подальший алгоритм перевірки правильності виконання кроків 3 та 4 аналогічний попередньому. Приклади неправильних та правильних варіантів відповідей зображені на рисунку 4.25 – 4.26 відповідно.

Перетворення ч... 67% - + Скинути у систему числення

Вступ / Крок 1 / Крок 2 / Крок 3 / Крок 4 / Результати

**Завдання 3**

Переведіть число  $1000000_2$  з двійкової системи числення ( $k=2$ ) у вісімкову систему числення ( $n=8$ ).

**Розв'язання:**

**Вказівка:** Спосіб перетворення числа з двійкової системи числення у вісімкову систему числення шляхом розбиття на триади пов'язаний зі ступенями двійки (для триади при перетворенні в вісімкову систему,  $2^3$ ).

3.1 **Не все виконано вірно! Перевірте та виправте помилки!** та:

0 0 1    0 0 0    0 0 0

**Підказка:** Якщо найстарша триада не заповнена до кінця, слід дописати до її старших розрядів нулі.

3.2 Проведіть заміну двійкових триад, починаючи з молодшої, на числа, що дорівнюють їм у вісімковій системі числення:

0 0 1    0 0 0    0 0 0

↓            ↓            ↓

1            2            0

Введіть число, яке отримали в вісімковій системі числення:

$1000000_2 = \underline{120}_8$

ПЕРЕВІРИТИ    ІНШИЙ ВАРИАНТ

ОСВІЖИ ДУМКИ

попередній крок    допомога викладача    завершити сеанс    наступний крок

Рисунок 4.25 – Виконання завдання 3 з неправильними відповідями

Перетворення ч... 67% - + Скинути у систему числення

Вступ / Крок 1 / Крок 2 / Крок 3 / Крок 4 / Результати

**Завдання 3**

Переведіть число  $1000000_2$  з двійкової системи числення ( $k=2$ ) у вісімкову систему числення ( $n=8$ ).

**Розв'язання:**

**Вказівка:** Спосіб перетворення числа з двійкової системи числення у вісімкову систему числення шляхом розбиття на триади пов'язаний зі ступенями двійки (для триади при перетворенні в вісімкову систему,  $2^3$ ).

3.1 **Відмінно! Крок пройдений! Перейдіть до наступного.** та:

0 0 1    0 0 0    0 0 0

**Підказка:** Якщо найстарша триада не заповнена до кінця, слід дописати до її старших розрядів нулі.

3.2 Проведіть заміну двійкових триад, починаючи з молодшої, на числа, що дорівнюють їм у вісімковій системі числення:

0 0 1    0 0 0    0 0 0

↓            ↓            ↓

1            0            0

Введіть число, яке отримали в вісімковій системі числення:

$1000000_2 = \underline{100}_8$

ПЕРЕВІРИТИ    ІНШИЙ ВАРИАНТ

ОСВІЖИ ДУМКИ

попередній крок    допомога викладача    завершити сеанс    наступний крок

Рисунок 4.26 – Виконання завдання 3 з правильними відповідями

По завершенню тренування усі отримані результати відображаються на сторінці «Результати». Система виводить повідомлення про успішне завершення роботи, покрокову діаграму результативності, витрачений час та одержаний результат тренування (рис.4.27).

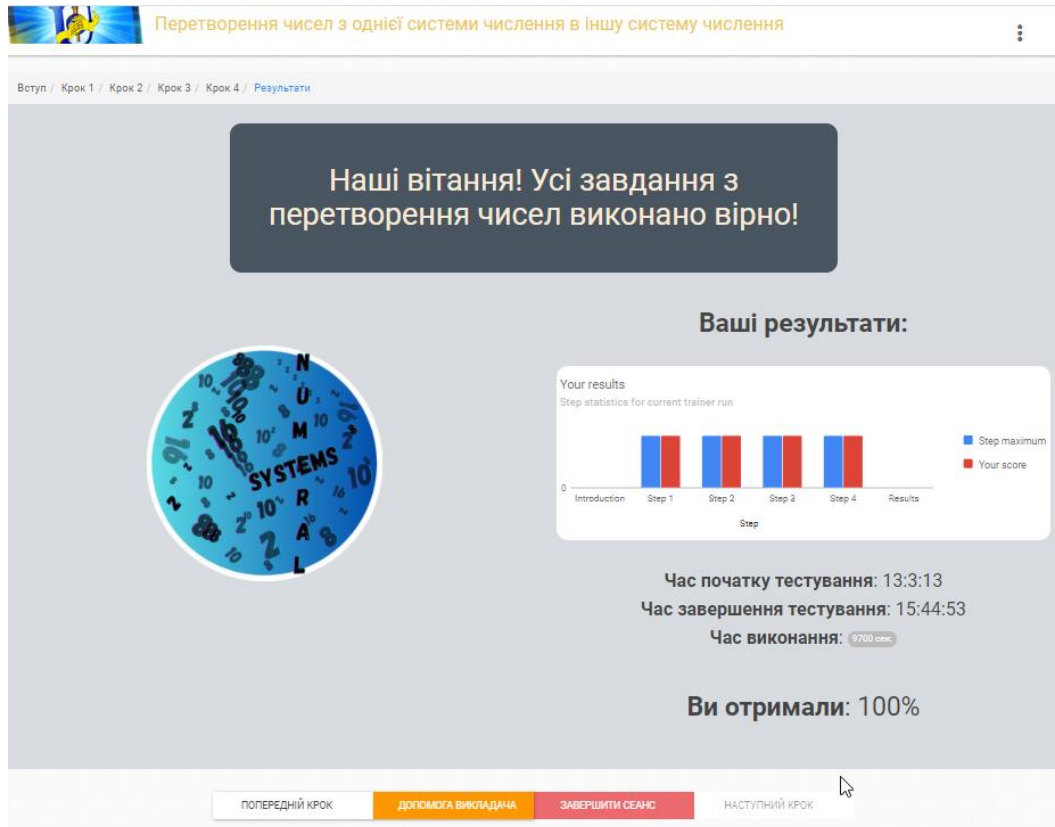


Рисунок 4.27 – Розділ «Результати»

Перевірка системи показала, що вона відповідає усім функціональним вимогам до неї. Усі компоненти працюють правильно: перемикання мови інтерфейсу відбувається без помилок, коректно формуються інтерактивні частини тренажера, а також вірно підраховуються та відображаються результати тренування.

Розроблену інформаційну систему було впроваджено до єдиної навчальної платформи Сумського державного університету. Скористатися можна за посиланням <https://elearning.sumdu.edu.ua/s/f7-1jsx>.

## ВИСНОВКИ

Результатом виконання кваліфікаційної магістерської роботи стала інформаційна технологія проєктування інтерактивних практичних завдань по теорії числових систем. Весь перелік поставлених завдань було виконано: проведено аналіз та огляд сучасної літератури, що стосується даної проблеми; проведено пошук та аналіз аналогічних проєктів, визначено їх переваги та недоліки; розроблено варіанти локалізації інтерфейсу інтерактивного додатку українською та англійською мовами; спроектовано дизайн додатку; визначено методологію та технології, необхідні для реалізації проєкту; проведено розробку інтерактивного тренажера та протестовано його; впроваджено тренажер до єдиної навчальної платформи СумДУ.

Продукт відповідає вимогам сучасного освітнього процесу та дає змогу студентам оволодіти методологією перетворення чисел у дистанційному форматі навчання, може вплинути на полегшення й удосконалення освітнього процесу та на підвищення засвоєння навчальних матеріалів і отримання практичних навичок здобувачами освіти. А також практична цінність даного продукту полягає у використанні його викладачами під час лекцій і практичних занять по системам числення. Результати вивчення аналогічних продуктів показали, що розробка і впровадження даної технології є актуальними.

Проте розроблена інформаційна технологія ще не повністю досконала та існують напрямки її розширення. Одним з них є розширення переліку завдань тренажера (наприклад, додати завдання по перетворенню чисел з десяткової системи числення в вісімкову, шістнадцяткову та інші системи). Окрім цього, якщо додаток буде використовуватися в інших країнах, можливе розширення мовного інтерфейсу додатковими мовами.



## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Шушкова Ю. В. Діяльність ЗВО в умовах війни: реалії, виклики та перспективи : матеріали круглого столу за участю порадників академічних груп та викладачів факультету управління фінансами та бізнесу. – 3 травня 2023 р. – Львів: ФУФБ, 2023. – 110 с.
2. Кіяновська Н. Поняття електронного навчання в контексті сучасної педагогічної науки. [Електронний ресурс]. – Режим доступу: [http://www.rusnauka.com/29\\_DWS\\_2012/Pedagogica/1\\_120037.doc.htm](http://www.rusnauka.com/29_DWS_2012/Pedagogica/1_120037.doc.htm) (дата звернення 28.09.2023).
3. Західна О. Р. Дистанційне навчання у ЗВО: моделі, технології, перспективи: матеріали круглого столу за участю порадників академічних груп та викладачів факультету управління фінансами та бізнесу. – 28 квітня 2021 р. – Львів: ФУФБ, 2021. С.40-44.
4. Cosker, Glynn (2023), "What Is Information Technology? A Beginner's Guide to the World of IT", Technology Blog, Rasmussen University. [Електронний ресурс]. – Режим доступу: <https://www.rasmussen.edu/degrees/technology/blog/what-is-information-technology/> (дата звернення 28.09.2023).
5. Грицюк О.М. Інформаційні технології в українській освіті: шляхи впровадження світового досвіду / О.С. Грицюк // Вісник КрНУ імені Михайла Остроградського. – 2018 (109). – Випуск 2. – Частина 2. – С. 15-20.
6. Antoniuk D. E-learning tools: definition and classification. Physical and Mathematical Education. 2019. Issue 3(21). P. 12-18.
7. Вимоги до навчально-методичних матеріалів дистанційної форми навчання та критерії їх оцінювання: наказ Міністерство освіти і науки України Сумський державний університет від 03.02.17 р. № 0048-І. 13 с.
8. Доценко Н.А. Дослідження стану використання навчальних комп'ютерних інтерактивних тренажерів в умовах інформаційно-освітнього середовища при підготовці бакалаврів з агроінженерії / А.Н. Доценко

//Освітній дискурс: збірник наукових праць, 2019.- В.15(7-8). – С. 90-104 - DOI 10.33930/ed.2019.5007.15(7-8)-7

9. Присяжнюк О. В. Особливості впровадження віртуальних тренажерів в дистанційні курси для студентів спеціальності “Комп’ютерні науки”. Priority directions of science and technology development: proceedings of the 3-th International scientific and practical conference (November 22-24, 2020, Kyiv). Kyiv, 2020. P. 362-366. – Режим доступу: <https://sci-conf.com.ua/iiimezhdunarodnaya-nauchno-prakticheskayakonferentsiyapriority-directions-of-science-andtechnology-development-22-24-noyabrya-2020-godakiev-ukraina-arhiv/>.

10. Олгодворець Л.В., Пасько О.О., Ткач О.П., Лукавенко І.М., Хижня Я.В. Віртуальні тренажери і симулятори як засоби засвоєння практичних навичок студентами природничих, інженерних та медичних спеціальностей // Шляхи вдосконалення позааудиторної роботи студентів: Матеріали ХІ Науково-методичної конференції, 19-20 травня 2022 р. – Суми: Сумський державний університет, 2022. – С.8-12.

11. Інтерактивний віртуальний тренажер learningapps. [Електронний ресурс]. – Режим доступу: <https://learningapps.org/watch?v=p5kgsq1b5> (дата звернення 05.11.2023).

12. Number System: Learn & Convert [Електронний ресурс]. – Режим доступу: [https://play.google.com/store/apps/details?id=com.rocky.numbersystemforstudent&hl=en\\_US](https://play.google.com/store/apps/details?id=com.rocky.numbersystemforstudent&hl=en_US) (дата звернення 06.11.2023)

13. Binary Game [Електронний ресурс]. – Режим доступу: <https://learningcontent.cisco.com/games/binary/index.html> (дата звернення 06.11.2023)

14. Редактори коду [Електронний ресурс]. – Режим доступу: <https://uk.javascript.info/code-editors> (дата звернення 13.11.2023).

15. IntelliJ\_IDEA [Електронний ресурс]. – Режим доступу: <https://www.jetbrains.com/idea/download/?section=windows#section=windows> (дата звернення 13.11.2023).
16. Apache NetBeans [Електронний ресурс]. – Режим доступу: <https://netbeans.apache.org/> (дата звернення 12.12.2023).
17. Visual Studio Code [Електронний ресурс]. – Режим доступу: <https://code.visualstudio.com/docs> (дата звернення 13.11.2023).
18. Stack overflow developer survey 2023 [Електронний ресурс]. – Режим доступу: <https://survey.stackoverflow.co/2023/#technology-most-popular-technologies> (дата звернення 13.11.2023).
19. Visual Studio Code [Електронний ресурс]. – Режим доступу: <https://code.visualstudio.com> (дата звернення 15.11.2023).
20. 20 найкращих інструментів і програм для графічного дизайну для початківців // Squeeze Growth LLP © 2020-2023. [Електронний ресурс]. – Режим доступу: <https://squeezegrowth.com/uk/best-graphic-design-software/> (дата звернення 12.12.2023).
21. Adobe Illustrator [Електронний ресурс]. – Режим доступу: <https://www.adobe.com/ua/> (дата звернення 12.12.2023).
22. Програми та інструменти в графічному дизайні // ТворчийOnline. [Електронний ресурс]. – Режим доступу: <https://www.creativosonline.org/uk/programas-herramientas-diseno-grafico.html> (дата звернення 12.12.2023).
23. Canva [Електронний ресурс]. – Режим доступу: <https://www.canva.com/> (дата звернення 25.11.2023)
24. Огляд популярних локальних серверів для сайту, блог Костаневіча Степана // блог Костаневіча Степана. [Електронний ресурс]. – Режим доступу: <https://jak.koshachek.com/articles/ogljad-populjarnih-lokalnih-serveriv-dlja-sajtu.html> (дата звернення 12.12.2023).

25. Денвер (набір програм) [Електронний ресурс]. – Режим доступу: [https://uk.wikipedia.org/wiki/Денвер\\_\(набір\\_програм\)](https://uk.wikipedia.org/wiki/Денвер_(набір_програм)) (дата звернення 12.12.2023).
26. Open Server Panel [Електронний ресурс]. – Режим доступу: <https://ospanel.io> (дата звернення 15.11.2023).
27. TOP 10 кращих розширень VS та корисних плагінів для Visual Studio [Електронний ресурс]. – Режим доступу: [https://cloud.itstep.org/blog\\_3/top-10-best-vs-extensions-and-useful-plugins-for-visual-studio](https://cloud.itstep.org/blog_3/top-10-best-vs-extensions-and-useful-plugins-for-visual-studio) (дата звернення 12.12.2023).
28. Connolly Randy, Hoar Ricardo. Fundamentals of Web Development 3rd Edition. - Pearson Education, 2022. - 1081 p. - ISBN13: 978-0-13-586333-6.
29. Основи HTML [Електронний ресурс]. – Режим доступу: [https://developer.mozilla.org/en-US/docs/Learn/Getting\\_started\\_with\\_the\\_web/HTML\\_basics](https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/HTML_basics) (дата звернення 13.12.2023).
30. Довідник по CSS властивостям [Електронний ресурс]. – Режим доступу: <https://css.in.ua/css/properties> (дата звернення 13.12.2023).
31. CSS basics [Електронний ресурс]. – Режим доступу: [https://developer.mozilla.org/en-US/docs/Learn/Getting\\_started\\_with\\_the\\_web/CSS\\_basics](https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/CSS_basics) (дата звернення 13.12.2023).
32. Nixon R. Learning PHP, MySQL & JavaScript: A Step-by-Step Guide to Creating Dynamic Websites (Learning PHP, MYSQL, Javascript, CSS & HTML5) / Robin Nixon., 2021. – 824 с.
33. Рейтинг мов програмування 2023 [Електронний ресурс]. – Режим доступу: <https://dou.ua/lenta/articles/language-rating-2023/> (дата звернення 15.11.2023).

34. Давиденко І. С. Аналіз мови програмування JavaScript / І. С. Давиденко, Н. П. Бабюк // Матеріали LI наукової-технічної конференції підрозділів ВНТУ, Вінниця, 31 травня 2022 р. – Електрон. текст. дані. – 2022.
35. Jensen J.Sch. The Missing Bootstrap 5 Guide Packt, 2022. - 384 p. - ISBN 180107643X, 9781801076432.
36. Benjamin Jakobus, Jason Marah. Mastering Bootstrap 4: Master the latest version of Bootstrap 4 to build highly customized responsive web apps 2nd Edition. - Packt Publishing, 2018. - 383 p. - ISBN: 978-1-78883-490-2.
37. Mustache.js - Logic-less {{mustache}} templates with JavaScript // Jan Lehnardt, GitHub. [Електронний ресурс]. – Режим доступу: <https://github.com/janl/mustache.js> (дата звернення 13.12.2023).
38. Ajax: The Definitive Guide: Interactive Applications for the Web 1st Edition by Anthony T. Holdener III. -2015-982 p. [Electronic resource] - URL: <https://pepa.holla.cz/wp-content/uploads/2015/11/Ajax-The-Definitive-Guide.pdf> (Last accessed: 14.12.2023)
39. Nixon Robin. Learning PHP, MySQL & JavaScript: A Step-by-Step Guide to Creating Dynamic Websites 7th Edition. - O'Reilly Media, Inc., 2023. - 132 p. - (Early Release). - ISBN 978-1-098-15229-1.
40. Working with JSON [Електронний ресурс]. – Режим доступу: <https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/JSON> (дата звернення 19.12.2023).
41. Introducing JSON [Електронний ресурс]. – Режим доступу: <https://www.json.org/json-en.html> (дата звернення 19.12.2023).
42. Boduch A., Chaffer J., Swedberg K. Learning jQuery 3 - Fifth Edition/ Packt Publishing, 2017. – 451 p.: ISBN 978-1-78588-298-2.
43. The Use of IDEF0 for the Design and Specification of Methodologies: [Електронний ресурс]. – Режим доступу: [https://www.researchgate.net/publication/2447898\\_The\\_Use\\_of\\_IDEF0\\_for\\_the\\_Design\\_and\\_Specification\\_of\\_Methodologies](https://www.researchgate.net/publication/2447898_The_Use_of_IDEF0_for_the_Design_and_Specification_of_Methodologies) (дата звернення: 22.10.2023).

44. UML Distilled: A Brief Guide to the Standard Object Modeling Language - Martin Fowler [Электронный ресурс]. – Режим доступа: <https://bedford-computing.co.uk/learning/wp-content/uploads/2015/10/UML-Distrilled-3nd.pdf> (дата звернення: 22.11.2023).

## ДОДАТОК А

**trainer.html**

```

<!DOCTYPE html>
<html>
<head>
  <title>jTrainer :: {{TRAINER_NAME}}</title>
  <meta charset="UTF-8"/>
  <meta name="description" content=""/>
  <meta name="viewport" content="width=device-width,
initial-scale=1">
  <link rel="shortcut icon"
href="/cdn/javascripsts/jtrainer/current/img/trainer/trainer-favicon.ico"
type="image/x-icon">
  <link rel="icon"
href="/cdn/javascripsts/jtrainer/current/img/trainer/trainer-favicon.ico"
type="image/x-icon">

  <script>
    var startTime = (new Date()).getTime();
  </script>
  <script
src="/cdn/javascripsts/jtrainer/current/js/jquery.min.js"></script>
  <script>
    $(window).load(function () {
      var endTime = (new Date()).getTime();
      var secondsLoading = ((endTime - startTime) /
1000).toFixed(4);
      $('.page-loading').append(secondsLoading + ' seconds.');
```

```

        <!-- jTrainer's engine script -->
        <script
src="/cdn/javascripts/jtrainer/current/js/trainer.engine.js"></script>
    </head>
    <body>
        <nav class="navbar navbar-default">
            <div class="navbar-header">
                <div><a class="navbar-brand" href="#"><span
class="trainer-brand"></span>{{TRAINER_NAME}}</a></div>
                </div>
                <div id="navbar" class="navbar-collapse collapse">
                    <div class="navbar-right info">
                        <div class="dropdown">
                            <button type="button" class="btn
btn-link dropdown-toggle" id="menu" data-toggle="dropdown"
                            aria-haspopup="true"
                            aria-expanded="true" aria-label="Option Vertical"><span class="glyphicon
glyphicon-option-vertical" aria-hidden="true"></span></button>
                            <ul class="dropdown-menu" aria-
labelledby="menu">
                                <li><a href="#" data-toggle="modal"
data-target="#aboutModal">About</a></li>
                                <li><a href="#" class="page-loading
text-muted"><span class="is-faded">Page generated in </span></a></li>
                                </ul>
                            </div>
                            <div class="modal fade" id="aboutModal" tabindex="-1"
role="dialog" aria-labelledby="myModalLabel">
                                <div class="modal-dialog" role="document">
                                    <div class="modal-content">
                                        <div class="modal-header">
                                            <button type="button" class="close" data-
dismiss="modal" aria-label="Close"><span aria-
hidden="true">&times;</span></button>
                                            <h4 class="modal-title"
id="aboutModalLabel">About</h4>
                                        </div>
                                        <div id="aboutBody" class="modal-body">
                                            ...
                                        </div>
                                        <div class="modal-footer">
                                            <button type="button" class="btn btn-default"
data-dismiss="modal">Close</button>
                                        </div>
                                    </div>
                                </div>
                            </div>
                        </div>
                    </div>
                </nav>
                <div class="modal modal-static fade" id="cogwheel-modal" role="dialog"
aria-hidden="true">
                    <div class="modal-dialog">
                        <div class="modal-content">
                            <div class="modal-body">
                                <div class="text-center">
                                    

```



```

                <h4><span id="cogwheel-desc">Processing</span>... </h4>
            </div>
        </div>
    </div>
</div>
<section class="stepspace" style="position: relative">
    <div class="calculator" id="calculator"></div>
    <ul class="breadcrumb bc-steps"></ul>
    <!-- Steps will be loaded here -->
</section>

<div class="text-center animated bounceInUp">
    <div class="btn-group btn-group-md rotators">
        <a id="prevController" href="javascript:void(0)" class="btn btn-
default col-xs-6 col-md-3">{{PREV_STEP}}</a>
        <a class="teacher-help btn btn-warning col-xs-6 col-md-3" data-
toggle="modal"
            data-target="#helpModal">{{TEACHER_HELP}}</a>
        <a id="endTrainer" class="btn btn-danger col-xs-6 col-md-3" data-
toggle="modal" data-target="#endModal">{{END_TRAINER}}</a>
        <a id="nextController" href="javascript:void(0)" class="btn btn-
default col-xs-6 col-md-3">{{NEXT_STEP}}</a>
    </div>
</div>

<!-- This is modal popup which shows up after click on "Teacher Help"
button -->
<div class="modal fade" id="helpModal" tabindex="-1" role="dialog"
aria-labelledby="helpModalLabel">
    <div class="modal-dialog" role="document">
        <div class="modal-content">
            <div class="modal-header">
                <button type="button" class="close" data-dismiss="modal"
aria-label="Close" id="closeButton"><span
                    aria-hidden="true">&times;</span></button>
                <h4 class="modal-title" id="helpModalLabel">
|{{TEACHER_HELP}}</h4>
            </div>
            <div class="modal-body">
                <label class="control-label">{{ENTER_HELP_TEXT}}</label>
                <input type="text" class="form-control" id="helpInput"/>
            </div>
            <div class="modal-footer">
                <button type="button" class="btn btn-default" data-
dismiss="modal">{{HELP_REQUEST_CLOSE}}</button>
                <button type="button" class="btn btn-primary"
onclick="Service.getHelpModal()">{{HELP_REQUEST_SEND}}
                </button>
            </div>
        </div>
    </div>
</div>
</div>
<!-- This is modal popup which shows up after click on "End trainer"
button -->
<div class="modal fade" id="endModal" tabindex="-1" role="dialog" aria-
labelledby="endModalLabel">

```

```

    <div class="modal-dialog" role="document">
      <div class="modal-content">
        <div class="modal-header">
          <button type="button" class="close" data-dismiss="modal"
aria-label="Close" id="closeButton"><span
          aria-hidden="true">&times;</span></button>
          <h4 class="modal-title" id="endModalLabel">
{{END_TRAINER}}</h4>
        </div>
        <div class="modal-body">
          <label class="control-label">{{END_ARE_YOU_SURE}}</label>
        </div>
        <div class="modal-footer">
          <button type="button" class="btn btn-default" data-
dismiss="modal">{{END_TRAINER_CLOSE}}</button>
          <button type="button" class="btn btn-primary"
onclick="Service.pushResultsEarly()" >
            {{END_TRAINER_SEND}}
          </button>
        </div>
      </div>
    </div>
</div>

<div class="progress trainer-progress-bar">
  <div class="progress-bar"></div>
</div>

<div class="alert alert-dismissible alert-success validation-alert
validation-alert-success">
  <button type="button" class="close" data-dismiss="alert">x</button>
  <strong>{{ALERT_WELL_DONE}}</strong>{{ALERT_STEP_SUCCESS}}
</div>
<div class="alert alert-dismissible alert-danger validation-alert
validation-alert-danger">
  <button type="button" class="close" data-dismiss="alert">x</button>
  <strong>{{ALERT_FAIL}}</strong> {{ALERT_STEP_FAIL}}
</div>

</body>
</html>
<script type="text/javascript"
src="/cdn/javascripts/jtrainer/current/js/additions/html2canvas.js"
async></script>
<script type="text/javascript">
  /* $('#calc').click(function () {
    $('div.calculator').load("/javascripts/jtrainer/current/calc.html").css
('display', 'block').draggable({
      cancel: '#result'
    });
  });*/
</script>

```

## index.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
  </head>
  <body style="margin:0; padding:0; overflow:hidden;">
    <iframe src="trainer.html" frameborder="0" style="position:
absolute; height: 100%;" height="100%" width="100%"></iframe>
  </body>
</html>
```

## introduction.html

```
<div class="row jumbotron introduction">
  <div class="col-md-6 animated zoomIn">
    
  </div>
  <div class="col-md-6 animated fadeInDown">
    <h1>{{TRAINER_NAME}}</h1>

    <p>
      <h5><strong>{{STEP_COUNT}}</strong> <span class="badge">
{{STEPS_COUNT}}</span></h5>
      <br/>
      <h5>{{CHOOSE_LANG}}</h5>
      <div class="flag-icon flag-icon-ua" title="ua" id="ua"></div>
      <div class="flag-icon flag-icon-en" title="en" id="en"></div>
    </p>
  </div>
</div>
```

## introduction.js

```
var a1 = 0, a1_bin = 0, a1_bin_back = 0, a1_even = [];
function makestep1() {
  a1 = Math.floor(Math.random() * (50-5+1)+5);

  a1_bin = a1.toString(2);
  a1_bin_back = a1_bin.split("").reverse().join("");

  let n = a1;
  a1_even = [];
  while(n > 0) {
    if(n % 2 === 0) {
      n = n / 2;
      a1_even.push(n);
    } else {
```

```

        n = n - 1;
        n = n / 2;
        a1_even.push(n);
    }
}

let str1 = '<tr><th>'+I18N.getConstants()
['STEP1_T7']+'</th><th>'+I18N.getConstants()
['STEP1_T8']+'</th><th>'+I18N.getConstants() ['STEP1_T9']+'</th><td
id="hints1" rowspan="'+(a1_bin.length + 2)+'"><span
class="showafter">'+I18N.getConstants()
['STEP1_HINT']+'</span></td></tr><tr><td>2</td><td>'+a1+'</td><td><div
class="form-group" for="sli1"><input name="sli1" class="form-control"
type="text" placeholder="?" /></div></td></tr>';
    for(let i=0; i<(a1_bin.length * 3); i+=3) {
        str1 += '<tr><td><div class="form-group" for="sli'+(i+2)+'"><input
name="sli'+(i+2)+'" class="form-control" type="text" placeholder="?"
/></div></td><td><div class="form-group" for="sli'+(i+3)+'"><input
name="sli'+(i+3)+'" class="form-control" type="text" placeholder="?"
/></div></td><td><div class="form-group" for="sli'+(i+4)+'"><input
name="sli'+(i+4)+'" class="form-control" type="text" placeholder="?"
/></div></td></tr>';
    }
    $(' .page1 table').append(str1);
    $(' .page1 input[type="text"]').attr('autocomplete', 'off');
}

var a2 = 0, a2_bin = 0;
function makestep2() {
    a2 = Math.floor(Math.random() * (50-5+1)+5);
    a2_bin = a2.toString(2);

    let str2 = '<nobr><span class="text">'+a2_bin+'<sub>2</sub> = </span>';
    for(let i=0; i<(a2_bin.length * 2); i+=2) {
        str2 += '<div class="form-group" for="s2i'+(i+1)+'"><input name="s2i'+(i
+1)+'" class="form-control odd-input" type="text" placeholder="?" /></div>
<span class="text">&middot; 2</span> <div class="form-group" for="s2i'+(i
+2)+'"><input name="s2i'+(i+2)+'" class="form-control even-input" type="text"
placeholder="?" /></div>';
        if(i != (a2_bin.length * 2 - 2)) {
            str2 += '<span class="text">+</span> ';
        }
    }
    str2 += '</nobr>';
    $(' .page2 .step2_end').append(str2);
    $(' .page2 input[type="text"]').attr('autocomplete', 'off');
}

var a3 = 0, a3_bin = 0, a3_oct = 0, a3_bin_num = 0, a3_bin_add = 0;
function makestep3() {
    a3 = Math.floor(Math.random() * (89-10+1)+10);
    a3_bin = a3.toString(2);
    a3_oct = a3.toString(8);

```

```

a3_bin_num = a3_bin.length;
a3_bin_add = a3_bin;

while(a3_bin_num % 3 !== 0) {
  a3_bin_num++;
  a3_bin_add = '0'+a3_bin_add;
}

let str31 = '<tr>';
for(let i=0; i<a3_bin_num; i++) {
  if((i+1) == 4 || (i+1) == 7) {
    str31 += '<td class="space"></td>';
  }
  str31 += '<td><div class="form-group" for="s31i'+(i+1)+'"><input
name="s31i'+(i+1)+'" class="form-control" type="text" placeholder="?"
/></div></td>';
}
str31 += '</tr>';
$('.page3 .part31').append(str31);

let str32 = '<tr>';
for(let i=0; i<a3_bin_num; i++) {
  if((i+1) == 4 || (i+1) == 7) {
    str32 += '<td class="space"></td>';
  }
  str32 += '<td class="triads">'+a3_bin_add[i]+'</td>';
}
str32 += '</tr><tr>';
for(let k=0; k<(a3_bin_num / 3); k++) {
  if(k == 1 || k == 2) {
    str32 += '<td class="space" style="border-left: 3px solid
#d8dce0;border-right: 3px solid #d8dce0;"></td>';
  }
  str32 += '<td colspan="3"><span class="text">&#8681;</span></td>';
}
str32 += '</tr><tr>';
for(let n=0; n<(a3_bin_num / 3); n++) {
  if(n == 1 || n == 2) {
    str32 += '<td class="space" style="border-left: 3px solid
#d8dce0;border-right: 3px solid #d8dce0;"></td>';
  }
  str32 += '<td colspan="3"><div class="form-group" for="s32i'+(n
+1)+'"><input name="s32i'+(n+1)+'" class="form-control" type="text"
placeholder="?" /></div></td>';
}
str32 += '</tr>';
$('.page3 .part32').append(str32);
$('.page3 input[type="text"]').attr('autocomplete', 'off');
}

```

```

var a4 = 0, a4_bin = 0, a4_hex = 0, a4_bin_num = 0, a4_bin_add = 0;
function makestep4() {
  a4 = Math.floor(Math.random()*(89-10+1)+10);
  a4_bin = a4.toString(2);
  a4_hex = a4.toString(16);

  console.log('a4 = '+a4+', a4_bin = '+a4_bin+', a4_hex = '+a4_hex);

  a4_bin_num = a4_bin.length;
  a4_bin_add = a4_bin;

  while(a4_bin_num % 4 !== 0) {
    a4_bin_num++;
    a4_bin_add = '0'+a4_bin_add;
  }
  let str41 = '<tr>';
  for(let i=0; i<a4_bin_num; i++) {
    if((i+1) == 5 || (i+1) == 9) {
      str41 += '<td class="space"></td>';
    }
    str41 += '<td><div class="form-group" for="s41i'+(i+1)+'"><input
name="s41i'+(i+1)+'" class="form-control" type="text" placeholder="?"
/></div></td>';
  }
  str41 += '</tr>';
  $('<div>.page4 .part41').append(str41);

  let str42 = '<tr>';
  for(let i=0; i<a4_bin_num; i++) {
    if((i+1) == 5 || (i+1) == 9) {
      str42 += '<td class="space"></td>';
    }
    str42 += '<td class="tetrads">'+a4_bin_add[i]+'</td>';
  }
  str42 += '</tr><tr>';
  for(let k=0; k<(a4_bin_num / 4); k++) {
    if(k == 1 || k == 2) {
      str42 += '<td class="space" style="border-left: 3px solid
#d8dce0;border-right: 3px solid #d8dce0;"></td>';
    }
    str42 += '<td colspan="4"><span class="text">&#8681;</span></td>';
  }
  str42 += '</tr><tr>';
  for(let n=0; n<(a4_bin_num / 4); n++) {
    if(n == 1 || n == 2) {
      str42 += '<td class="space" style="border-left: 3px solid
#d8dce0;border-right: 3px solid #d8dce0;"></td>';
    }
    str42 += '<td colspan="4"><div class="form-group" for="s42i'+(n
+1)+'"><input name="s42i'+(n+1)+'" class="form-control" type="text"
placeholder="?" /></div></td>';
  }
  str42 += '</tr>';
  $('<div>.page4 .part42').append(str42);
  $('<div>.page4 input[type="text"]').attr('autocomplete', 'off');
}

```

```

var step1check = false, step2check = false, step3check = false, step4check =
false;
var introduction = function () {
  this.preDispatch = function () {
    };

  this.postDispatch = function () {
    $('#endTrainer').attr('disabled', 'disabled').addClass('disabled');
    $('#div.flag-icon').each(function() {
      $(this).attr('onclick', 'window.location.href = \'?lang=\' +
$(this).attr("id")');
    });
    $('#end_end').click(function() {
      let stepId = Rotator.currentStepId();
      if(stepId === 1) {
        if(step1check) {
          Rotator.nextResult();
        } else {
          $('.page1 button.check').click();
          if(!step1check) {
            Scorer.addScore(0);
            Rotator.nextResult();
          }
        }
      } else if(stepId === 2) {
        if(step2check) {
          Rotator.nextResult();
        } else {
          $('.page2 button.check').click();
          if(!step2check) {
            Scorer.addScore(0);
            Rotator.nextResult();
          }
        }
      } else if(stepId === 3) {
        if(step3check) {
          Rotator.nextResult();
        } else {
          $('.page3 button.check').click();
          if(!step3check) {
            Scorer.addScore(0);
            Rotator.nextResult();
          }
        }
      } else if(stepId === 4) {
        if(step4check) {
          Rotator.nextResult();
        } else {
          $('.page4 button.check').click();
          if(!step4check) {
            Scorer.addScore(0);
            Rotator.nextResult();
          }
        }
      }
    }
  }
}

```

```

    $('.alert-warning').hide();|
    $('#endModal').css('display', 'none');
    $('.modal-dackdrop').remove();
  });
};

this.mustache = function () {
  return {
    STEPS_COUNT: Rotator.getStepsCount()-2
  }
};

```

## step1.html

```

<div class="jumbotron page1">
  <div class="cond">
    <h5>{{STEP1_T1}}</h5>
    <p>{{STEP1_T2}} <span class="als1"></span><sub>10</sub>
    {{{STEP1_T3}}}</p>
  </div>
  <p>{{STEP1_T4}}</p>
  <p>{{STEP1_T5}} <span class="als1"></span><sub>10</sub>
  {{{STEP1_T6}}}</p>
  <table align="center" border="3"></table>

  <div class="showafter">
    <p>{{STEP1_T10}}</p>
    <span class="text">
      <span class="als1"></span><sub>10</sub> =
    {{{STEP1_INPUT}}} <sub>2</sub>
    </span>
  </div>

  <div style="text-align: center; margin-top: 5px;">
    <button class="btn btn-success check">{{CHECK}}</button>
    <button class="btn btn-default shuffle">{{SHUFFLE}}</button>
  </div>

  <div class="container">
    <hr>
    <div id="hidden-theory">
      <h5>{{STEP1_T11}}</h5>
      <p class="theory">{{STEP1_T12}}</p>
      <p>{{STEP1_T13}}</p>
      
      <p class="theory">{{STEP1_T14}}</p>
    </div>

    <div style="text-align: center; margin-top: 5px;">
      <button class="btn btn-primary theory" >
    {{{REFRESH_YOUR_THOUGHTS}}}</button>
    </div>
    <hr>
  </div>
</div>

```



**step1.js**

```

var VStep1;// A variable for future validator
var step1 = function () {
  this.preDispatch = function () {

  };

  this.postDispatch = function () {
    $('#endTrainer').removeAttr('disabled').removeClass('disabled');
    makestep1();
    $('#page1 .als1').html(a1);
    function checkinput1() {
      let corans = [], corinp = [];
      let inpnum = $('#page1 table input').length;
      let rownum = $('#page1 table tr').length;
      corans.push(al_bin_back[0]);
      corinp.push('sli1');
      for(let i=0, k=1; i<(rownum - 2), k<(rownum - 2); i+=3, k++) {
        corans.push('2');
        corinp.push('sli'+(2+i));
        corans.push(al_even[k - 1].toString());
        corinp.push('sli'+(3+i));
        corans.push(al_bin_back[k]);
        corinp.push('sli'+(4+i));
      }
      corans.push('-');
      corinp.push('sli'+(inpnum - 2));
      corans.push('0');
      corinp.push('sli'+(inpnum - 1));
      corans.push('-');
      corinp.push('sli'+inpnum);

      let corstep1 = 0;
      for(let n=0; n<inpnum; n++) {
        if($('#input[name="'+corinp[n]+'"]').val() === corans[n]) {
          corstep1++;
          $('#div[for="'+corinp[n]+'"]').removeClass('has-
error').addClass('has-success');
        } else {
          $('#div[for="'+corinp[n]+'"]').removeClass('has-
success').addClass('has-error');
        }
      }

      if(corstep1 === inpnum) {
        return true;
      } else {
        return false;
      }
    }

    $('#page1 button.check').click(function () {
      if($('#page1 div.showafter').css('display') == 'none') {
        if(checkinput1()) { $('#page1 .page1 td#hints1').css('display',
'block');
                                $('#page1 .showafter').css('display',
'block');

```

```

                                $('.page1 input[name="step1-input"]').focus
());
        } else {
            $('.alert-danger').show();
        }
        } else {
            if($('.input[name="step1-input"]').val() !== al_bin) {
                $('.alert-danger').show();
                $('.div[for="step1-input"]').removeClass('has-
success').addClass('has-error');
            } else {
                step1check = true;
                $('.div[for="step1-input"]').removeClass('has-error').addClass
('has-success');
                $('.alert-success').show();
                $('.alert-danger').hide();
                Scorer.addScore(25);
                $('.page1 .check, .page1 .shuffle, #endTrainer').attr
('disabled', 'disabled').addClass('disabled');
                Rotator.enableNextButton();
            }
        }
    });

    $('.page1 button.shuffle').click(function(){
        $('.input[name="step1-input"]').val('');
        $('.page1 table').html('');
        makestep1();

        $('.page1 .als1').html(al);
        $('.page1 div.form-group').removeClass('has-error').removeClass
('has-success');
        $('.page1 .showafter').css('display', 'none');
        $('.alert-danger').hide();
    });

    $('.page1 button.theory').click(function(){
        if($('.page1 div#hidden-theory').css('display') == 'none') {
            $('.page1 div#hidden-theory').css('display', 'block');
            $(this).text(I18N.getConstants()['OK']);
            $('.alert-danger').hide();
            $('.alert-success').hide();
        } else {
            $('.page1 div#hidden-theory').css('display', 'none');

            $(this).text(I18N.getConstants()['REFRESH_YOUR_THOUGHTS']);
        }
    });
};

this.mustache = function () {
    return {
        STEP1_INPUT: new TextInput('step1-input')
            .render()
    }
}
};

```

## step2.html

```

<div class="jumbotron page2">
  <div class="cond">
    <h5>{{{STEP2_T1}}}</h5>
    <p>{{{STEP1_T2}}} <span class="a2s2"></span><sub>2</sub>
{{{STEP2_T3}}}</p>
  </div>
  <p>{{{STEP2_T4}}}</p>
  <p>{{{STEP2_T5}}}</p>
  <div class="step2_end"></div>

  <div class="showafter"><p>{{{STEP2_T6}}}</p>
    <span class="text">
      <span class="a2s2"></span><sub>2</sub> =
{{{STEP2_INPUT}}} <sub>10</sub>
    </span>
  </div>

  <div style="text-align: center; margin-top: 5px;">
    <button class="btn btn-success check">{{CHECK}}</button>
    <button class="btn btn-default shuffle">{{SHUFFLE}}</button>
  </div>

  <div class="container">
    <hr>
    <div id="hidden-theory">
      <h5>{{{STEP2_T7}}}</h5>
      <p class="theory">{{{STEP2_T8}}}</p>
      
      <p class="theory">{{{STEP2_T9}}}</p>
      <p>{{{STEP2_T10}}}</p>
      
      <p class="theory">{{{STEP2_T11}}}</p>
      
      <p class="theory">{{{STEP2_T12}}}</p>
    </div>

    <div style="text-align: center; margin-top: 5px;">
      <button class="btn btn-primary theory" >
{{{REFRESH_YOUR_THOUGHTS}}}</button>
    </div>
  </div>
</div>

```

**step2.js**

```

var VStep2;
var step2 = function () {

  this.preDispatch = function () {

  };

  this.postDispatch = function () {
    $('#endTrainer').removeAttr('disabled').removeClass('disabled');
    makestep2();
    $('.page2 .a2s2').html(a2_bin);
    function checkinput2() {
      let corans = [], corinp = [];
      let inpnum = $('.page2 .step2_end input').length;
      for(let i=0, k=(inpnum / 2 - 1), n=0; i<inpnum, k>=0, n<(inpnum /
2); i+=2, k--, n++) {
        corans.push(a2_bin[n]);
        corinp.push('s2i'+(i+1));
        corans.push(k.toString());
        corinp.push('s2i'+(i+2));
      }
      corans.push(a2.toString());
      corinp.push('step2-input');

      let corstep2 = 0;
      for(let n=0; n<=inpnum; n++) {
        if($('.input[name="'+corinp[n]+'"]').val() === corans[n]) {
          corstep2++;
          $('.div[for="'+corinp[n]+'"]').removeClass('has-
error').addClass('has-success');
        } else {
          $('.div[for="'+corinp[n]+'"]').removeClass('has-
success').addClass('has-error');
        }
      }
      if(corstep2 === inpnum) {
        return true;
      } else {
        return false;
      }
    }

    $('.page2 button.check').click(function () {
      if($('.page2 div.showafter').css('display') == 'none') {
        if(checkinput2()) { $('.page2 .showafter').css('display',
'block');
          $('.page2 input[name="step2-
input"]').focus();
        } else {
          $('.alert-danger').show();
        }
      } else {
        if($('.input[name="step2-input"]').val() !== a2.toString()) {
          $('.alert-danger').show();
        } else {
          step2check = true;
        }
      }
    });
  };
};

```

```

        $('#div[for="step2-input"]').removeClass('has-
error').addClass('has-success');
        $('.alert-success').show();
        $('.alert-danger').hide();
        Scorer.addScore(25);
        $('#page2 .check, .page2 .shuffle, #endTrainer').attr
('disabled', 'disabled').addClass('disabled');
        Rotator.enableNextButton();
    }
});

$('#page2 button.shuffle').click(function(){
    $('#input[name="step2-input"]').val('');
    $('#page2 .step2_end').html('');

    makestep2();
    $('#page2 .a2s2').html(a2_bin);
    $('#page2 div.form-group').removeClass('has-error').removeClass
('has-success');
    $('#page2 .showafter').css('display', 'none');
    $('.alert-danger').hide();
});
$('#page2 button.theory').click(function(){
    if($('#page2 div#hidden-theory').css('display') == 'none') {
        $('#page2 div#hidden-theory').css('display', 'block');
        $(this).text(I18N.getConstants()['OK']);
        $('.alert-danger').hide();
        $('.alert-success').hide();
    } else {
        $('#page2 div#hidden-theory').css('display', 'none');
        $(this).text(I18N.getConstants()['REFRESH_YOUR_THOUGHTS']);
    }
});
};

this.mustache = function () {
    return {
        STEP2_INPUT: new TextInput('step2-input')
        .render()
    }
}
};

```

### step3.html

```

<div class="jumbotron page3">
  <div class="cond">
    <h5>{{{STEP3_T1}}}</h5>
    <p>{{{STEP3_T2}}} <span class="a3s3"></span><sub>2</sub>
{{{STEP3_T3}}}</p>
  </div>
  <p>{{{STEP3_T4}}}</p>
  <p>{{{STEP3_T5}}}</p>

```

```

    <p>{{STEP3_T6}} <span class="a3s3"></span><sub>2</sub>
    {{{STEP3_T7}}}</p>
    <table align="center" border="3" class="part31"></table>
    <p>{{STEP3_T8}}</p>

    <div class="showafter">
        <p>{{STEP3_T9}}</p>
        <table align="center" border="3" class="part32"></table>
        <p>{{STEP3_T10}}</p>
        <span class="text">
            <span class="a3s3"></span><sub>2</sub> =
    {{{STEP3_INPUT}}} <sub>8</sub>
        </span>
    </div>

    <div style="text-align: center; margin-top: 5px;">
        <button class="btn btn-success check">{{CHECK}}</button>
        <button class="btn btn-default shuffle">{{SHUFFLE}}</button>
    </div>

    <div class="container">
        <hr>
        <div id="hidden-theory">|
            <h5>{{STEP3_T11}}</h5>
            <p class="theory">{{STEP3_T12}}</p>
            <p>    {{STEP3_T13}}</p>
            
            <p class="theory">{{STEP3_T14}}</p>
            
            <p class="theory">{{STEP3_T15}}</p>
        </div>

        <div style="text-align: center; margin-top: 5px;">
            <button class="btn btn-primary theory" >
    {{{REFRESH_YOUR_THOUGHTS}}}</button>
        </div>
        <hr>
    </div>
</div>

```

### step3.js

```

var VStep3;// A variable for future validator
var step3 = function () {
    this.preDispatch = function () {

    };

    this.postDispatch = function () {
        $('#endTrainer').removeAttr('disabled').removeClass('disabled');
        makestep3();
        $('.page3 .a3s3').html(a3_bin);
        function checkinput31() {
            let corans = [], corinp = [];

```

```

for(let i=0; i<a3_bin_num; i++) {
    corans.push(a3_bin_add[i]);
    corinp.push('s31i'+(i+1));
}

let corstep31 = 0;
for(let n=0; n<a3_bin_num; n++) {
    if($('input[name="'+corinp[n]+'"]').val() === corans[n]) {
        corstep31++;
        $('div[for="'+corinp[n]+'"]').removeClass('has-
error').addClass('has-success');
    } else {
        $('div[for="'+corinp[n]+'"]').removeClass('has-
success').addClass('has-error');
    }
}
if(corstep31 === a3_bin_num) {
    return true;
} else {
    return false;
}
}

function checkinput32() {
    let corans = [], corinp = [];
    for(let i=0; i<a3_oct.length; i++) {
        corans.push(a3_oct[i]);
        corinp.push('s32i'+(i+1));
    }
    corans.push(a3_oct);
    corinp.push('step3-input');

    let corstep32 = 0;
    for(let n=0; n<=a3_oct.length; n++) {
        if($('input[name="'+corinp[n]+'"]').val() === corans[n]) {
            corstep32++;
            $('div[for="'+corinp[n]+'"]').removeClass('has-
error').addClass('has-success');
        } else {
            $('div[for="'+corinp[n]+'"]').removeClass('has-
success').addClass('has-error');
        }
    }

    if(corstep32 === (a3_oct.length + 1)) {
        return true;
    } else {
        return false;
    }
}

$('.page3 button.check').click(function () {
    if($('.page3 div.showafter').css('display') == 'none') {
        if(checkinput31()) {
            $('.page3 .showafter').css('display', 'block');
        }
    }
}

```

```

        } else {
            $('.alert-danger').show();
        }
    } else {
        if(!checkinput32()) {
            $('.alert-danger').show();
            $('#div[for="step3-input"]').removeClass('has-
success').addClass('has-error');
        } else {
            step3check = true;
            $('#div[for="step3-input"]').removeClass('has-
error').addClass('has-success');
            $('.alert-success').show();
            $('.alert-danger').hide();
            Scorer.addScore(25);
            $('#page3 .check, .page3 .shuffle,
#endTrainer').attr('disabled', 'disabled').addClass('disabled');
            Rotator.enableNextButton();
        }
    }
});

$('#page3 button.shuffle').click(function(){
    $('#input[name="step3-input"]').val('');
    $('#page3 table').html('');
    makestep3();

    $('#page3 .a3s3').html(a3_bin);
    $('#page3 div.form-group').removeClass('has-error').removeClass
('has-success');
    $('#page3 .showafter').css('display', 'none');
    $('.alert-danger').hide();
});
$('#page3 button.theory').click(function(){
    if($('#page3 div#hidden-theory').css('display') == 'none') {
        $('#page3 div#hidden-theory').css('display', 'block');
        $(this).text(I18N.getConstants()['OK']);
        $('.alert-danger').hide();
        $('.alert-success').hide();
    } else {
        $('#page3 div#hidden-theory').css('display', 'none');
        $(this).text(I18N.getConstants()['REFRESH_YOUR_THOUGHTS']);
    }
});

};

this.mustache = function () {
    return {
        STEP3_INPUT: new TextInput('step3-input')
            .render()
    }
}
};

```



## step4.html

```

<div class="jumbotron page4">
  <div class="cond">
    <h5>{{STEP4_T1}}</h5>
    <p>{{STEP4_T2}} <span class="a4s4"></span><sub>2</sub>
    {{{STEP4_T3}}}</p>
  </div>
  <p>{{STEP4_T4}}</p>
  <p>{{STEP4_T5}}</p>
  <p>{{STEP4_T6}} <span class="a4s4"></span><sub>2</sub>
  {{{STEP4_T7}}}</p>
  <table align="center" border="3" class="part41"></table>
  <p>{{STEP4_T8}}</p>

  <div class="showafter">
    <p>{{STEP4_T9}}</p>
    <table align="center" border="3" class="part42"></table>
    <p>{{STEP4_T10}}</p>
    <span class="text">
      <span class="a4s4"></span><sub>2</sub> =
    {{{STEP4_INPUT}}}} <sub>16</sub>
    </span>
  </div>

  <div style="text-align: center; margin-top: 5px;">
    <button class="btn btn-success check">{{CHECK}}</button>
    <button class="btn btn-default shuffle">{{SHUFFLE}}</button>
  </div>

  <div class="container">
    <hr>
    <div id="hidden-theory">
      <h5>{{STEP4_T11}}</h5>
      <p class="theory">{{{STEP4_T12}}}</p>
      <p>{{{STEP4_T13}}}</p>
      
      <p class="theory">{{{STEP4_T14}}}</p>
      
      <p class="theory">{{{STEP4_T15}}}</p>
    </div>

    <div style="text-align: center; margin-top: 5px;">
      <button class="btn btn-primary theory" >
    {{{REFRESH_YOUR_THOUGHTS}}}</button>
    </div>
  </div>
</div>

```

**step4.js**

```

var VStep4;// A variable for future validator
var step4 = function () {
  this.preDispatch = function () {
  };
  this.postDispatch = function () {
    $('#endTrainer').removeAttr('disabled').removeClass('disabled');
    makestep4();
    $('.page4 .a4s4').html(a4_bin);
    function checkinput41() {
      let corans = [], corinp = [];
      for(let i=0; i<a4_bin_num; i++) {
        corans.push(a4_bin_add[i]);
        corinp.push('s41i'+(i+1));
      }
      let corstep41 = 0;
      for(let n=0; n<a4_bin_num; n++) {
        if($('#input[name="'+corinp[n]+'"]').val() === corans[n]) {
          corstep41++;
          $('#div[for="'+corinp[n]+'"]').removeClass('has-
error').addClass('has-success');
        } else {
          $('#div[for="'+corinp[n]+'"]').removeClass('has-
success').addClass('has-error');
        }
      }
      if(corstep41 === a4_bin_num) {
        return true;
      } else {
        return false;
      }
    }
    function checkinput42() {
      let corans = [], corinp = [];
      for(let i=0; i<a4_hex.length; i++) {
        corans.push(a4_hex[i].toUpperCase());
        corinp.push('s42i'+(i+1));
      }
      corans.push(a4_hex.toUpperCase());
      corinp.push('step4-input');

      let corstep42 = 0;
      for(let n=0; n<=a4_hex.length; n++) {
        if($('#input[name="'+corinp[n]+'"]').val() === corans[n]) {
          corstep42++;

          $('#div[for="'+corinp[n]+'"]').removeClass('has-
error').addClass('has-success');
        } else {
          $('#div[for="'+corinp[n]+'"]').removeClass('has-
success').addClass('has-error');
        }
      }
      if(corstep42 === (a4_hex.length + 1)) {
        return true;
      } else {
        return false;
      }
    }
  }
}

```

```

    $('#page4 button.check').click(function () {
        if($('#page4 div.showafter').css('display') == 'none') {
            if(checkinput41()) {
                $('#page4 .showafter').css('display', 'block');
            } else {
                $('#alert-danger').show();
            }
        } else {
            if(!checkinput42()) {
                $('#alert-danger').show();
                $('#div[for="step4-input"]').removeClass('has-
success').addClass('has-error');
            } else {
                step4check = true;
                $('#div[for="step4-input"]').removeClass('has-
error').addClass('has-success');
                $('#alert-success').show();
                $('#alert-danger').hide();
                Scorer.addScore(25);
                $('#page4 .check, .page4 .shuffle,
#endTrainer').attr('disabled', 'disabled').addClass('disabled');
                Rotator.enableNextButton();
            }
        }
    });
    $('#page4 button.shuffle').click(function(){
        $('#input[name="step4-input"]').val('');
        $('#page4 table').html('');
        makestep4();
        $('#page4 .a4s4').html(a4_bin);
        $('#page4 div.form-group').removeClass('has-error').removeClass
('has-success');
        $('#page4 .showafter').css('display', 'none');
        $('#alert-danger').hide();
    });
    $('#page4 button.theory').click(function(){
        if($('#page4 div#hidden-theory').css('display') == 'none') {
            $('#page4 div#hidden-theory').css('display', 'block');
            $(this).text(I18N.getConstants()['OK']);
            $('#alert-danger').hide();
            $('#alert-success').hide();
        } else {
            $('#page4 div#hidden-theory').css('display', 'none');

            $(this).text(I18N.getConstants()['REFRESH_YOUR_THOUGHTS']);
        }
    });
};
this.mustache = function () {
    return {
        STEP4_INPUT: new TextInput('step4-input')
        .render()
    }
}
};

```

## results.html

```

<div class="row jumbotron results">
  <div class="cond">
    <h1>{{GREETING}}</h1>
  </div>

  <div class="col-md-6 animated zoomIn">
    
  </div>

  <div class="col-md-6">
    <h3>{{RESULTS_TITLE}}</h3>
    <div class="container">
      <div class="chart" id="gc"></div>
    </div>
    <p>
      <h4>{{RESULTS_START_TIME}}: {{START_TIME}}</h4>
      <h4>{{RESULTS_END_TIME}}: {{END_TIME}}</h4>
      <h4>{{RESULTS_TIME_DIFF}}: <span class="badge">{{TIME_DIFF}}
{{RESULTS_TIME_SEC}}</span></h4>
      <br/>
      <h3>{{RESULTS_YOUR_SCORE}}: {{RESULTS_POINTS_IN_PERCENT}}%</h3>
    </p>
  </div>
</div>

```

## results.js

```

var results = function () {

  this.preDispatch = function () {
    Scorer.end();
  };

  this.postDispatch = function () {
    var g = new GoogleCharts(); // Making a call to google charts api to
    build a column chart

    var scores = Rotator.getAllStepScores();
    var userScores = Scorer.getUserStepScores();

    var data = [
      ['Step', 'Step maximum', 'Your score'],
      ['Introduction', scores[0], userScores[0]],
      ['Step 1', scores[1], userScores[1]],
      ['Step 2', scores[2], userScores[2]],
      ['Step 3', scores[3], userScores[3]],
      ['Step 4', scores[4], userScores[4]],
      ['Results', scores[5], userScores[5]]
    ];

```

```

var options = {
  chart: {
    title: 'Your results',
    subtitle: 'Step statistics for current trainer run'
  }
};

g.setType("bar");
g.setLibrary("google.charts.Bar");
g.setData(data);
g.setOptions(options);
g.doQuery($('div#gc'));

if (PRODUCTION)
  Service.pushResults();
};

this.mustache = function () {
  return {
    START_TIME: Scorer.getFormattedStartTime(),
    END_TIME: Scorer.getFormattedEndTime(),
    TIME_DIFF: Scorer.getTimeDifference(),
    SCORE: Scorer.getScore(),
    RESULTS_POINTS_IN_PERCENT: Scorer.getScoreInPercent()
  }
};
};

```

### **trainer.config.json**

```

{
  "LANGUAGES": {
    "ua": "Українська",
    "en": "English"
  },
  "DEFAULT_LANG": "ua",
  "TRAINER_SCORE": 100,
  "TRAINER_AUTHOR": "Zoia Taranets",
  "COURSE_AUTHOR": "Oksana Shovkoplyas",
  "PRODUCTION": true
}

```

**trainer.steps.json**

```
[
  {
    "filename"      : "introduction",
    "score"         : 0,
    "hasScript"    : true
  },
  {
    "filename"      : "step1",
    "score"         : 25,
    "hasScript"    : true
  },
  {
    "filename"      : "step2",
    "score"         : 25,
    "hasScript"    : true
  },
  {
    "filename"      : "step3",
    "score"         : 25,
    "hasScript"    : true
  },
  {
    "filename"      : "step4",
    "score"         : 25,
    "hasScript"    : true
  },
  {
    "filename"      : "results",
    "score"         : 0,
    "hasScript"    : true
  }
]
```

**en.json**

```
{
  "local": "English",
  "lang": {
    "TRAINER_NAME": "Conversion of numbers from one numeral system to
another",
    "MAX_SCORE": "Max score",
    "NEXT_STEP": "Next step",
    "PREV_STEP": "Previous step",
    "ENTER_TEXT": "Enter number",
    "CHOOSE_LANG": "Choose language",
    "CHOOSE_SELECT": "Choose element from the list",
    "TEACHER_HELP": "Teachers help",
    "ENTER_HELP_TEXT": "Describe what you are having problem with...",
    "HELP_REQUEST_SEND": "Send",
    "HELP_REQUEST_CLOSE": "Cancel",
  }
}
```

```

"ALERT_WELL_DONE": "Well done!",
"ALERT_FAIL": "Oh snap!",
"ALERT_STEP_SUCCESS": " The step done perfectly! Go to the next
one.",
"ALERT_STEP_FAIL": "Check and fix errors!",
"ALERT_STEP_WARNING": "Badly! Step is failed. Go to the next one.",
"STEP_COUNT": "Number of steps",
"END_TRAINER": "Finish session",
"END_ARE_YOU_SURE": "Are you sure you want to end trainer
execution?",
"END_TRAINER_CLOSE": "Continue current session",
"END_TRAINER_SEND": "Finish current session",
"CALCULATOR" : "Calculator",
"CHECK": "Check",
"SHUFFLE": "Another variant",
"REFRESH_YOUR_THOUGHTS": "Refresh your thoughts!",
"OK": "OK!!!",
"ATTEMPTS_LEFT": "Attempts left - ",

"STEP1_T1": "<strong style='color:antiquewhite;'><u>Problem
1</u></strong>",
"STEP1_T2": "Convert the number",
"STEP1_T3": "from decimal numeral system (k=10) <br />to the binary
numeral system (n=2) by dividing by the base.",
"STEP1_T4": "<strong><em>Solution:</em></strong>",
"STEP1_T5": "Divide the number",
"STEP1_T6": "by 2.<br />The results of the division record in the
table. In the field without a number, enter the sign '-' (minus).",
"STEP1_T7": "The base<br />of the system",
"STEP1_T8": "The integer part<br />of the division",
"STEP1_T9": "Remainder<br />of division",
"STEP1_HINT": "<strong style='color:red;'>Hint:</strong><br />Write
down<br />the number<br />of bottom-up",
"STEP1_T10": "Enter the number you received in the binary numeral
system:",
"STEP1_T11": "<br /><strong>Conversion of integers from the decimal
numeral system to binary by the division method</strong><br /> <br />",

"STEP1_T12": "The conversion of an integer from the decimal numeral
system to any other is carried out by successive division of the number on
the basis of the new numeral system. Division is performed until the last
quotient is less than the divisor. The resulting remainders from the
division, taken in the reverse order, will be the values of the digits of the
number in the new numeral system. The last bit is the most significant digit
of the binary number. <br />To understand how the division method works to
convert from a decimal numeral system to a binary <em><u>consider the
following example:</u></em> convert the number 38<sub>10</sub>. Let's divide
the number 38<sub>10</sub> on the basis of the binary numeral system 2, and
write the results in table 1. <br /><br />",
"STEP1_T13": "<strong>Table 1 - Conversion of a number from the
decimal numeral system to binary</strong>",
"STEP1_T14": "<br />Write the code of the binary number in the
direction of the arrow (that is, from bottom to top). Thus, the decimal
number 38<sub>10</sub> corresponds to the binary equivalent of
100110<sub>2</sub>. The correctness of the result can be checked by
converting the binary number to the decimal system.",

```

"STEP2\_T1": "<strong style='color:antiquewhite;'><u>Problem  
2</u></strong>",  
 "STEP2\_T2": "Convert the number",  
 "STEP2\_T3": "from binary numeral system (k=2) <br />to the decimal  
numeral system (n=10).",  
 "STEP2\_T4": "<strong><em>Solution:</em></strong>",  
 "STEP2\_T5": "Write down the numerical values into the corresponding  
gaps:",  
 "STEP2\_T6": "Enter the number you received in the decimal numeral  
system:",  
 "STEP2\_T7": "<br /><strong>Conversion of integers from binary numeral  
system to decimal numeral system</strong><br /> <br />",  
 "STEP2\_T8": "<strong><em> The binary numeral system </em></strong>is  
a numeral system that uses only two digits: 0 and 1. Each number in the  
binary numeral system represents a power of two. When converting from the  
binary numeral system to the decimal numeral system, the initial number must  
be divided by base 10. <br />However, division is difficult to perform in the  
binary system. Therefore, in practice, if it is necessary to convert numbers  
from a system with a small base to a system with a larger base, it is  
convenient to use the general notation of numbers in the form of a polynomial  
according to the formula: <br />",  
 "STEP2\_T9": "where a - is the alphabet of the numeral system, p - is  
the base of the numeral system, n - is the weight of the digit. <br /><br  
>To convert a binary number into a decimal number, it must be written in the  
form of a polynomial consisting of products of the digits of the number with  
the corresponding powers of the number 2, and calculate according to the  
rules of decimal arithmetic. When converting, it is convenient to use the  
table of powers of two (see table 2).",  
 "STEP2\_T10": "<strong>Table 2 - Table of Powers of Number  
2</strong>",  
 "STEP2\_T11": "<br /><em><u>Let's consider an example:</u></em> it is  
necessary to convert the number 11101000<sub>2</sub> from the binary numeral  
system to the decimal one. Using a polynomial, we will get the following  
entry:",  
 "STEP2\_T12": "As a result, we will get that the number  
11101000<sub>2</sub> of the binary numeral system is equal to  
232<sub>10</sub> of the decimal numeral system.",  
 "STEP3\_T1": "<strong style='color:antiquewhite;'><u>Problem  
3</u></strong>",  
 "STEP3\_T2": "Convert the number",  
 "STEP3\_T3": "from binary numeral system (k=2) <br />to the octal  
system (n=8).",  
 "STEP3\_T4": "<strong><em>Solution:</em></strong>",  
 "STEP3\_T5": "<em><u>Instruction:</u></em> The method of converting a  
number from the binary numeral system to the octal numeral system by dividing  
it into triads is related to the degrees of two (for the triad when  
converting to the octal system, <strong>2<sup>3</sup></strong>).",  
 "STEP3\_T6": "3.1. Divide the number ",  
 "STEP3\_T7": "by triad, start with the lower bits:",  
 "STEP3\_T8": "<strong style='color:red;'>Hint:</strong> If the older  
triad is not filled until the end, you should add in its most significant  
bits of zeros.",  
 "STEP3\_T9": "3.2. Replace your binary triads, starting with the lower  
bits, by a number equal to them in octal numeral system:",  
 "STEP3\_T10": "Enter the number received in the octal system:",  
 "STEP3\_T11": "<br /><strong>Conversion of integers from the binary  
numeral system to the octal by dividing into triads </strong><br /> <br />",



"STEP3\_T12": "The conversion of a number from the binary numeral system to the octal number can be done with the help of some rules. First, it is necessary to divide the binary number into groups of three digits (executive triads) starting from the least significant bit (that is, from right to left). The group that is in the leftmost position, can contain less than three binary digits. If the senior triad is not filled to the end, it is necessary to add zeros to its senior digits. Then match each group with its octal equivalent, that is, use the table of correspondences between the binary and octal numeral systems (table 3).<br />",

"STEP3\_T13": "<strong>Table 3 - Table of correspondence of numbers in different numeral systems</strong>",

"STEP3\_T14": "<br /><em><u>Let's consider an example:</u></em> it is necessary to convert the number 100001<sub>2</sub> from the binary numeral system to the octal numeral system. First, we divide the binary number into triads: ",

"STEP3\_T15": "Then we will replace the binary triads, starting from the youngest, with the corresponding numbers in the octal numeral system. As a result, we will get that the number 100001<sub>2</sub> of the binary numeral system is equal to 101<sub>8</sub> of the octal numeral system.",

"STEP4\_T1": "<strong style='color:antiquewhite;'><u>Problem 4</u></strong>",

"STEP4\_T2": "Convert the number",

"STEP4\_T3": " from binary numeral system (k=2) <br />to the hexadecimal numeral system (n=16).",

"STEP4\_T4": "<strong><em>Solution:</em></strong>",

"STEP4\_T5": "<em><u>Instruction:</u></em> The method of converting a number from the binary numeral system to hexadecimal by dividing it into tetrads is related to the degrees of two (for a tetrad when converting to the hexadecimal system, <strong>2<sup>4</sup></strong>).",

"STEP4\_T6": "4.1. Divide the number ",

"STEP4\_T7": "by tetrads, start with the lower bits:",

"STEP4\_T8": "<strong style='color:red;'>Hint:</strong> If the older tetrad is not filled until the end, you should add in its most significant bits of zeros.",

"STEP4\_T9": "4.2. Replace your binary tetrads, starting with the lower bits, by a number equal to them in hexadecimal system <u><em>(numeric values in the input fields must be filled in UPPER CASE)</em></u>:",

"STEP4\_T10": "Enter the number received in the hexadecimal system:",

"STEP4\_T11": "<br /><strong>Conversion of integers from the binary numeral system to hexadecimal by the method of division into tetrads</strong><br /> <br />",

"STEP4\_T12": "The conversion of a number from the binary numeral system to hexadecimal is carried out using certain rules. First, it is necessary to divide the binary number into groups of four digits (tetrads) starting from the least significant bit (that is, from right to left). The group located in the leftmost position can contain less than four binary digits. If the senior tetrad is not completely filled, it is necessary to add zeros to its senior digits. Then match each group with its hexadecimal equivalent (table 3).<br />",

"STEP4\_T13": "<strong>Table 3 - Table of correspondence of numbers in different number systems</strong>",

"STEP4\_T14": "<br /><em><u>Let's consider an example:</u></em> it is necessary to convert the number 1000100<sub>2</sub> from the binary numeral system to the hexadecimal numeral system. First, we divide the binary number into tetrads: ",

```
"STEP4_T15": "Then we will replace the binary tetrads, starting from the youngest, with the corresponding numbers in the hexadecimal numeral system. As a result, we will get that the number 1000100<sub>2</sub> of the binary numeral system is equal to 44<sub>8</sub> of the hexadecimal numeral system.",
```

```
"GREETING": "Our congratulations! All number conversion problems are correct!",
```

```
"RESULTS_TITLE": "<strong>Your scores:</strong>",
"RESULTS_START_TIME": "<strong>You started at</strong>",
"RESULTS_END_TIME": "<strong>Finished at</strong>",
"RESULTS_TIME_DIFF": "<strong>Time to complete</strong>",
"RESULTS_TIME_SEC": "sec.",
"RESULTS_YOUR_SCORE": "<strong>Your score is</strong>",
```

```
"STEP1_NAME": "Introduction",
"STEP2_NAME": "Step 1",
"STEP3_NAME": "Step 2",
"STEP4_NAME": "Step 3",
"STEP5_NAME": "Step 4",
"STEP6_NAME": "Results"
```

```
}
```

```
}
```

## ua.json

```
{
  "local": "Українська",
  "lang": {
    "TRAINER_NAME": "Перетворення чисел з однієї системи числення в іншу систему числення",
    "MAX_SCORE": "Кількість балів",
    "NEXT_STEP": "Наступний крок",
    "PREV_STEP": "Попередній крок",
    "ENTER_TEXT": "Введіть число",
    "CHOOSE_LANG": "Оберіть мову",
    "CHOOSE_SELECT": "Оберіть елемент зі списку",
    "TEACHER_HELP": "Допомога викладача",
    "ENTER_HELP_TEXT": "Опишіть з чим Вам потрібна допомога...",
    "HELP_REQUEST_SEND": "Відправити",
    "HELP_REQUEST_CLOSE": "Відмінити",
    "CALCULATOR": "Калькулятор",
    "ALERT_WELL_DONE": "Відмінно!",
    "ALERT_FAIL": "Не все виконано вірно!",
    "ALERT_STEP_SUCCESS": "Крок пройдений! Перейдіть до наступного.",
    "ALERT_STEP_FAIL": "Перевірте та виправте помилки!",
    "ALERT_STEP_WARNING": "Погано! Крок провалений. Перейдіть до наступного.",
    "STEP_COUNT": "Кількість кроків",
    "END_TRAINER": "Завершити сеанс",
    "END_ARE_YOU_SURE": "Ви впевнені, що хочете завершити виконання тренажера?",
    "END_TRAINER_CLOSE": "Продовжити поточний сеанс",
    "END_TRAINER_SEND": "Завершити поточний сеанс",
```

```

"CHECK": "Перевірити",
"SHUFFLE": "Інший варіант",
"REFRESH_YOUR_THOUGHTS": "Освіжи думки!",
"OK": "OK!!!",
"ATTEMPTS_LEFT": "Ще спроб - ",

"STEP1_T1": "<strong style='color:antiquewhite;'><u>Завдання 1</u>
</strong>",
"STEP1_T2": "Переведіть число",
"STEP1_T3": "з десяткової системи числення (k=10) <br />у двійкову
систему числення (n=2) шляхом ділення на основу.",
"STEP1_T4": "<strong><em>Розв'язання:</em></strong>",
"STEP1_T5": "Здійсніть ділення числа",
"STEP1_T6": "на основу системи числення 2.<br />Результати ділення
запишіть у таблицю. У поле без цифри впишіть знак '-' (мінус).",
"STEP1_T7": "Основа <br />системи",
"STEP1_T8": "Число (ціла частина<br /> від ділення)",
"STEP1_T9": "Остача від<br />ділення",
"STEP1_HINT": "<strong style='color:red;'>Підказка:</strong><br />
Записувати<br />число знизу-<br />вгору",
"STEP1_T10": "Введіть число, яке отримали в двійковій системі
числення:",
"STEP1_T11": "<br /><strong>Перетворення цілих чисел з десяткової
системи числення в двійкову методом ділення</strong><br /> <br />",
"STEP1_T12": "Переведення цілого числа з десяткової системи числення
у будь-яку іншу здійснюється шляхом послідовного ділення числа на основу
нової системи числення. Ділення виконується доти, поки остання частка не
стане менше дільника. Отримані остачі від ділення, взяті у зворотному
порядку, будуть значеннями розрядів числа в новій системі числення. Остання
частка є самим старшим розрядом двійкового числа. <br />Для розуміння, як
працює метод ділення для перетворення з десяткової системи числення в
двійкову <em><u>розглянемо наступний приклад:</u></em> перетворимо число
38<sub>10</sub>. Проведемо ділення числа 38<sub>10</sub> на основу двійкової
системи числення 2, а результати запишемо в таблицю 1. <br /><br />",
"STEP1_T13": "<strong>Таблиця 1 - Перетворення числа з десяткової
системи числення у двійкову</strong>",
"STEP1_T14": "<br />Код двійкового числа запишемо за напрямком
стрілки (тобто знизу вгору). Таким чином, десятковому числу 38<sub>10</sub>
відповідає двійковий еквівалент 100110<sub>2</sub>. Правильність результату
можна перевірити переведенням двійкового числа в десяткову систему.",

"STEP2_T1": "<strong style='color:antiquewhite;'><u>Завдання 2</u>
</strong>",
"STEP2_T2": "Переведіть число",
"STEP2_T3": "з двійкової системи числення (k=2) <br />у десяткову
систему числення (n=10).",
"STEP2_T4": "<strong><em>Розв'язання:</em></strong>",
"STEP2_T5": "Запишіть у пропуски відповідні числові значення:",
"STEP2_T6": "Введіть число, яке отримали в десятковій системі
числення:",
"STEP2_T7": "<br /><strong>Перетворення цілих чисел з двійкової
системи числення в десяткову </strong><br /> <br />",
"STEP2_T8": "<strong><em>Двійкова система числення</em></strong> -
це система числення, в якій використовуються лише дві цифри: 0 та 1. Кожне
число у двійковій системі числення представляє ступінь двійки. При
переведенні з двійкової системи числення в десяткову початкове число
необхідно ділити на основу 10. <br />Проте ділення виконати в двійковій
системі важко. Тому на практиці за необхідності переведення чисел із системи
з малою основою в систему з більшою основою зручно користуватися загальним
записом чисел у вигляді поліному за формулою: <br />",

```

"STEP2\_T9": "де  $a$  – алфавіт системи числення,  $p$  – основа системи числення,  $n$  – вага розряду. <br /><br />Для переведення двійкового числа в десяткове необхідно його записати у вигляді многочлена, що складається з творів цифр числа з відповідними ступенями числа 2, та обчислити за правилами десяткової арифметики. При переведенні зручно користуватися таблицею ступенів двійки (див. таблиця 2).",

"STEP2\_T10": "<strong>Таблиця 2 – Таблиця ступенів числа 2</strong>",

"STEP2\_T11": "<br /><em><u>Розглянемо приклад:</u></em> необхідно перевести число  $11101000_2$  з двійкової системи числення в десяткову. Використовуючи поліном, отримаємо наступний запис:",

"STEP2\_T12": "В результаті отримаємо, що число  $11101000_2$  двійкової системи числення дорівнює  $232_{10}$  десяткової системи числення.",

"STEP3\_T1": "<strong style='color:antiquewhite;'><u>Завдання 3</u></strong>",

"STEP3\_T2": "Переведіть число",

"STEP3\_T3": "з двійкової системи числення ( $k=2$ ) <br />у вісімкову систему числення ( $n=8$ ).",

"STEP3\_T4": "<strong><em>Розв'язання:</em></strong>",

"STEP3\_T5": "<em><u>Вказівка:</u></em> Спосіб перетворення числа з двійкової системи числення у вісімкову систему числення шляхом розбиття на триади пов'язаний зі ступенями двійки (для триади при перетворенні в вісімкову систему, <strong>2<sup>3</sup></strong>).",

"STEP3\_T6": "3.1 Розділіть число",

"STEP3\_T7": "на триади, починаючи з молодшого біта:",

"STEP3\_T8": "<strong style='color:red;'>Підказка:</strong> Якщо найстарша триада не заповнена до кінця, слід дописати до її старших розрядів нулі.",

"STEP3\_T9": "3.2 Проведіть заміну двійкових триад, починаючи з молодшої, на числа, що дорівнюють їм у вісімковій системі числення:",

"STEP3\_T10": "Введіть число, яке отримали в вісімковій системі числення:",

"STEP3\_T11": "<br /><strong>Перетворення цілих чисел з двійкової системи числення у вісімкову методом розбиття на триади </strong><br /><br />",

"STEP3\_T12": "Переведення числа з двійкової системи числення у вісімкову можна виконати за допомогою деяких правил. Спочатку необхідно розділити двійкове число на групи по три цифри (виконавчі триади) починаючи з молодшого біту (тобто з права наліво). Група, що знаходиться в крайньому лівому положенні, може містити менше трьох двійкових цифр. Якщо старша триада заповнена не до кінця, необхідно дописати в її старші розряди нулі. Потім кожній групі поставити у відповідність її восьмеричний еквівалент, тобто скористатися таблицею відповідностей між двійковою та вісімковою системами числення (таблиця 3).<br />",

"STEP3\_T13": "<strong>Таблиця 3 – Таблиця відповідності чисел у різних системах числення</strong>",

"STEP3\_T14": "<br /><em><u>Розглянемо приклад:</u></em> необхідно перевести число  $1000001_2$  з двійкової системи числення у вісімкову систему числення. Спочатку двійкове число розбиваємо на триади:",

"STEP3\_T15": "Потім проведемо заміну двійкових триад, починаючи з молодшої, на відповідні числа у вісімковій системі числення. В результаті отримаємо, що число  $1000001_2$  двійкової системи числення дорівнює  $101_8$  вісімкової системи числення.",

"STEP4\_T1": "<strong style='color:antiquewhite;'><u>Завдання 4</u></strong>",

"STEP4\_T2": "Переведіть число",

```

"STEP4_T3": "з двійкової системи числення (k=2) <br />у шістнадцяткову систему числення (n=16).",
"STEP4_T4": "<strong><em>Розв'язання:</em></strong>",
"STEP4_T5": "<em><u>Вказівка:</u></em> Спосіб перетворення числа з двійкової системи числення у шістнадцяткову систему числення шляхом розбиття на тетради пов'язаний зі ступенями двійки (для тетради при перетворенні в шістнадцяткову систему, <strong>2<sup>4</sup></strong>).",
"STEP4_T6": "3.1 Розділіть число ",
"STEP4_T7": "на тетради, починаючи з молодшого біта:",
"STEP4_T8": "<strong style='color:red;'>Підказка:</strong> Якщо найстарша тетрада не заповнена до кінця, слід дописати до її старших розрядів нулі.",
"STEP4_T9": "3.2 Проведіть заміну двійкових тетрад, починаючи з молодшої, на числа, що дорівнюють їм у шістнадцятковій системі числення <u><em>(числові значення в полях введення повинні бути заповнені у ВЕРХНЬОМУ РЕГІСТРІ)</em></u>:",
"STEP4_T10": "Введіть число, яке отримали в шістнадцятковій системі числення:",
"STEP4_T11": "<br /><strong>Перетворення цілих чисел з двійкової системи числення у шістнадцяткову методом розбиття на тетради</strong><br />",
"STEP4_T12": "Переведення числа з двійкової системи числення у шістнадцяткову здійснюється за допомогою певних правил. Спочатку необхідно розділити двійкове число на групи по чотири цифри (тетради) починаючи з молодшого біту (тобто з права наліво). Група, що знаходиться в крайньому лівому положенні, може містити менше чотирьох двійкових цифр. Якщо старша тетрада заповнена не до кінця, необхідно дописати в її старші розряди нулі. Потім кожній групі поставити у відповідність її шістнадцяткочисельний еквівалент (таблиця 3).<br />",
"STEP4_T13": "<strong>Таблиця 3 - Таблиця відповідності чисел у різних системах числення</strong>",
"STEP4_T14": "<br /><em><u>Розглянемо приклад:</u></em> необхідно перевести число 1000100<sub>2</sub> з двійкової системи числення у шістнадцяткову систему числення. Спочатку двійкове число розбиваємо на тетради:",
"STEP4_T15": "Потім проведемо заміну двійкових тетрад, починаючи з молодшої, на відповідні числа у шістнадцятковій системі числення. В результаті отримаємо, що число 1000100<sub>2</sub> двійкової системи числення дорівнює 44<sub>8</sub> шістнадцятковій системі числення.",

"GREETING": "Наші вітання! Усі завдання з перетворення чисел виконано вірно!",
"RESULTS_TITLE": "<strong>Ваші результати:</strong>",
"RESULTS_START_TIME": "<strong>Час початку тестування</strong>",
"RESULTS_END_TIME": "<strong>Час завершення тестування</strong>",
"RESULTS_TIME_DIFF": "<strong>Час виконання</strong>",

"RESULTS_TIME_SEC": "сек.",
"RESULTS_YOUR_SCORE": "<strong>Ви отримали</strong>",

"STEP1_NAME": "Вступ",
"STEP2_NAME": "Крок 1",
"STEP3_NAME": "Крок 2",
"STEP4_NAME": "Крок 3",
"STEP5_NAME": "Крок 4",
"STEP6_NAME": "Результати"
}
}

```

**trainer.custom.css**

```

.navbar-brand {
    height: auto !important;
}
.foto {
    margin: 40px auto;
    padding: 30px 15px;
}
.fadeInDown h1, .results h1 {
    background-color: #495561;
    color: antiquewhite;
    margin: 25px auto;
    padding: 15px 5px;
    border-radius: 15px;
    font-size: 40px;
}
.jumbotron p, h5 {
    font-size: 25px;
}
.cond {
    text-align: center;
    max-width: 800px;
    background-color: #495561;
    color: ghostwhite;
    margin: 25px auto;
    padding: 15px 5px;
    border-radius: 15px;
}
.stepspace {
    background-color: #d8dce0;
}
.page1, .page2, .page3, .page4, .results, .introduction {
    text-align: center;
    background-color: #d8dce0;
}
.text {
    font-size: 25px;
    font-weight: 250;
}

.page1 table th, .page1 table td, .page3 table td, .page4 table td {
    border: 3px solid #3EBDFE;
    background-color: #e8f9fc;
    margin-bottom: 15px;
    margin-top: 10px;
    padding: 3px 10px;
    font-size: 16px !important;
    text-align: center;
}

```

```

.page1 td#hints1 {
    background-color: #d8dce0;
    border-top: 3px solid #d8dce0;
    border-right: 3px solid #d8dce0;
    border-bottom: 3px solid #d8dce0;
    border-left: 3px solid #3EBDFF;
    text-align: center;
}

.page1 input[name="step1-input"], .page2 input[name="step2-input"],
.page3 input[name="step3-input"], .page4 input[name="step4-input"] {
    width: 110px !important;
    font-size: 25px;
}

.page1 input[name="step1-input"]::placeholder,
.page2 input[name="step2-input"]::placeholder,
.page3 input[name="step3-input"]::placeholder,
.page4 input[name="step4-input"]::placeholder {
    font-size: 16px;
}

.showafter {
    display: none;
}

.page2 input.even-input {
    font-size: 14px;
    vertical-align: 50%;
    width: 20px !important;
    height: 15px !important;
}

.page2 input.odd-input {
    width: 40px !important;
    font-size: 25px !important;
}

.page2 .step2_end {
    margin-bottom: 15px;
}

.page3 td.space, .page4 td.space {
    border-bottom: 3px solid #d8dce0 !important;
    border-top: 3px solid #d8dce0 !important;
    width: 60px !important;
    background-color: #d8dce0;
}

.page3 table input[type="text"], .page4 table input[type="text"] {
    width: 40px !important;
}

.page3 table.part32 td.triads, .page4 table.part42 td.tetrads {
    width: 60px;
    height: 40px;
}

.page3 table.part32 td[colspan="3"], .page4 table.part42 td[colspan="4"] {
    border-left: 3px solid #d8dce0 !important;
    border-right: 3px solid #d8dce0 !important;
    border-bottom: 3px solid #d8dce0 !important;
    border-top: 3px solid #d8dce0 !important;
    background-color: #d8dce0;
}

```

```
.results #gc {
    text-align: center;
    background-color: white;
    margin: 25px auto;
    padding: 15px 5px;
    border-radius: 15px;
}
.chart {
    left: 0;
    top: 0;
    margin: 0 auto;
    width: 100%;
    height: 100%;
    position: relative;
}
.check, .shuffle, .theory {
    margin: 1px 10px;
    min-width: 200px;
}
p.theory {
    text-align: justify;
}
hr {
border-top: 4px solid #3EBDFF;
}

#hidden-theory {
    display: none;
}
```