

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ**  
**ЦЕНТР ЗАОЧНОЇ, ДИСТАНЦІЙНОЇ ТА ВЕЧІРНЬОЇ ФОРМ НАВЧАННЯ**  
**КАФЕДРА КОМП'ЮТЕРНИХ НАУК**

# **КВАЛІФІКАЦІЙНА МАГІСТЕРСЬКА РОБОТА**

**на тему:**

**«Інформаційна технологія забезпечення  
інтерактивності навчального процесу в  
загальноосвітніх школах»**

**Завідувач  
випускаючої кафедри**

**Довбиш А.С.**

**Керівник роботи**

**Проценко О.Б.**

**Студентка групи ІН.мз-01с**

**Онічко В.А.**

**СУМИ 2021**

**Сумський державний університет**

(назва вузу)

Факультет \_\_\_\_\_ ЦЗДФН \_\_\_\_\_ Кафедра \_\_\_\_\_ Комп'ютерних наук \_\_\_\_\_  
 Спеціальність \_\_\_\_\_ 122 «Комп'ютерні науки» («Інформатика») \_\_\_\_\_

Затверджую:

зав.кафедрою \_\_\_\_\_

“ \_\_\_\_\_ ” \_\_\_\_\_ 20\_\_ р.

**ЗАВДАННЯ  
 НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ) СТУДЕНТОВІ**

**Онічко Вікторії Анатоліївни**

(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) Інформаційна технологія забезпечення інтерактивності навчального процесу в загальноосвітніх школах

затверджую наказом по інституту від “ \_\_\_\_\_ ” \_\_\_\_\_ 20\_\_ р. № \_\_\_\_\_

2. Термін здачі студентом закінченого проекту (роботи)

\_\_\_\_\_

3. Вхідні дані до проекту (роботи)

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

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

1) Інформаційний огляд; 2) Вибір методів вирішення задачі; 3) Проектування та програмна реалізація.

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

\_\_\_\_\_

Керівник

\_\_\_\_\_

(підпис)

Завдання прийняв до виконання

\_\_\_\_\_

(підпис)

## КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів дипломного проекту (роботи)	Термін виконання проекту (роботи)	Примітка
1.	Інформаційний огляд	20.09.2021 – 12.10.2021	
2.	Вибір методів вирішення задачі	13.10.2021 – 01.11.2021	
3.	Проектування та програмна реалізація	02.11.2021 – 24.11.2021	
4.	Оформлення кваліфікаційної магістерської роботи	25.11.2021 – 02.12.2021	

Студент – дипломник

\_\_\_\_\_

(підпис)

Керівник проекту

\_\_\_\_\_

(підпис)

## РЕФЕРАТ

**Записка:** 106 стор., 45 рис., 6 додатків, 85 джерел.

**Об'єкт дослідження** — інформаційна технологія з інтерактивними завданнями для учнів загальноосвітніх шкіл.

**Мета роботи** — спроектувати та створити базу даних із використанням СУБД MySQL; розробити інформаційну веб-систему на основі фреймворка Yii; реалізувати мовою JavaScript інтерактивні завдання для учнів шкіл, забезпечити можливість перевірки правильності виконання завдань; протестувати інформаційну веб-систему.

**Методи дослідження** — для розробки інформаційної веб-системи застосовуються мови PHP, JavaScript, а також фреймворк Yii, для створення бази даних – СУБД MySQL.

**Результати** — виконаний інформаційний огляд, аналіз актуальності розробки інформаційної технології, вибрані засоби вирішення задачі. Спроектвана та створена база даних із використанням СУБД MySQL. Інформаційна веб-система з інтерактивними завданнями та їх перевірка реалізовані за допомогою засобів мови JavaScript на основі фреймворку Yii. Функціонал інформаційної веб-системи протестований.

ВЕБ-СИСТЕМА, ФРЕЙМВОРК, БАЗА ДАНИХ, HTML, CSS, JAVASCRIPT,  
PHP, YII, BOOTSTRAP, MYSQL

## ЗМІСТ

ВСТУП .....	6
1 ІНФОРМАЦІЙНИЙ ОГЛЯД.....	8
1.1 Інформаційно-комунікаційні технології в освіті.....	8
1.2 Принципи роботи веб-додатків та проектування їх структури .....	9
1.3 Огляд існуючих веб-додатків для інтерактивного навчання .....	14
1.4 Постановка задачі .....	21
2 ВИБІР МЕТОДІВ ВИРІШЕННЯ ЗАДАЧІ.....	22
2.1 Вибір методів розробки .....	22
2.2 Вибір засобів розробки .....	24
3 ПРОЕКТУВАННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ.....	33
3.1 Проектування структури сайту .....	33
3.2 Проектування інформаційної системи .....	36
3.3 Реалізації основних компонентів .....	39
3.4 Тестування інформаційної веб-системи.....	60
ВИСНОВОК.....	68
СПИСОК ЛІТЕРАТУРИ.....	69
ДОДАТКИ.....	76
Додаток А .....	76
Додаток Б.....	78
Додаток В .....	83
Додаток Г.....	92
Додаток Д.....	96
Додаток Е .....	100

## ВСТУП

Повсякденна діяльність сучасної людини тісно пов'язана з використанням інформаційно-комунікаційних технологій. Вони застосовуються майже в усіх сферах та продовжують стрімко поширюватися з кожним днем. Медицина, наука, промисловість, бізнес – неможливо уявити їх подальший розвиток без комп'ютерних технологій.

Серед основних пріоритетів інформатизації в Україні можна виділити освітню галузь, а саме впровадження ІКТ у навчальний процес у середніх навчальних закладах.

На сьогоднішній день більшість українських шкіл мають комп'ютерні класи, проте рідко використовують переваги інформаційно-комунікаційних технологій у повній мірі, найчастіше обмежуючись застосуванням комп'ютера на уроках інформатики, для демонстрації презентацій або перегляду відео, інколи – для тестування.

Сучасний світ з його величезними обсягами інформації та різноманіттю її представлення вимагає від людини не тільки вміння її сприймати, а ще й критично мислити. Формуванню цих навичок сприяє інтерактивне навчання, яке виводить навчальний процес за рамки пасивного сприйняття розповіді вчителя учнями. Інтерактивні методи покликані підвищити ефективність навчання шляхом формування вмінь шукати, аналізувати інформацію та виділяти головне, аргументовано висловлювати свої думки, творчо підходити до вирішення поставлених завдань. Комп'ютерні технології безперечно можуть значно спростити організацію такого виду діяльності для вчителя, а для учнів – зробити навчання цікавим, пізнавальним та різнобічним.

Однією з головних переваг використання ІКТ є наочність представлення інформації, що особливо актуально в контексті дисциплін природничого циклу – хімії, географії, біології. Використання комп'ютера

дозволить не лише переглядати надруковані рисунки, схеми, таблиці, карти, а й взаємодіяти з ними під час практичних робіт та тестування. Наявні наразі в Україні програми є примітивними та не відповідають високому рівню розвитку комп'ютерних технологій. Удосконалення існуючих та розробка нових програм, незважаючи на трудомісткість, може стати потужним поштовхом до викорінення застарілих методик навчання та впровадження нових, орієнтованих на сучасний інформаційний світ.

# 1 ІНФОРМАЦІЙНИЙ ОГЛЯД

## 1.1 Інформаційно-комунікаційні технології в освіті

Модернізація навчального процесу шляхом впровадження інформаційно-комунікаційних технологій уже давно є головною тенденцією розвитку освіти як за кордоном, так і в Україні. В процесі інформатизації середніх навчальних закладів можна умовно виділити три етапи:

- оснащення шкіл комп'ютерною технікою;
- інтеграція комп'ютерів у навчальний процес;
- підключення комп'ютерів до мережі Інтернет.

Українські школи у своїй більшості оснащені мультимедійним та комп'ютерним обладнанням з доступом до мережі Інтернет, але ефективність його використання знаходиться на низькому рівні порівняно з можливостями сучасних інформаційно-комунікаційних технологій у нашій країні. Причинами цього часто є відсутність у школах кваліфікованих фахівців з підтримки комп'ютерів та їх ПЗ, недостатня обізнаність учителів у сфері ІКТ, застаріле апаратне та програмне забезпечення.

Серед найбільш поширених застосувань комп'ютерних технологій у вітчизняній освіті можна виділити: знайомство з базовими комп'ютерними додатками та програмування на уроках інформатики, демонстрація презентацій, перегляд відеофрагментів, іноді – тестування. В іншому вчителі зазвичай дотримуються традиційних методик та прийомів, з тих чи інших причин ігноруючи необхідність взаємодії учня з комп'ютером.

Використання спеціалізованого програмного забезпечення зможе значно покращити якість викладання предмету, вмотивувати учнів та оптимізувати засвоєння ними матеріалу. Примітивність та вузька спеціалізація більшості існуючих програм не здатні викликати в учня інтерес до поглиблення знань з дисципліни. Школам бракує сучасних додатків, які були б цікавими та корисними як для вчителів, так і для учнів.



З огляду на те, що не всі школи обладнані сучасним апаратним забезпеченням, використання веб-технологій є найбільш оптимальним варіантом при розробці інформатизованих методик. Веб-додатки можуть стабільно працювати практично на будь-якій моделі комп'ютера з будь-якою версією операційної системи, адже не потребують установки додаткового ПЗ, достатньо лише наявності браузера. Функціонал таких додатків може поступово розширюватися, починаючи від вузькоспеціалізованих програм для тестування з певної дисципліни та закінчуючи цілісною системою для навчання, перевірки знань і моніторингу оцінок, використовуваною вчителями, учнями та їх батьками.

## **1.2 Принципи роботи веб-додатків та проектування їх структури**

Веб-додаток являє собою клієнт-серверну прикладну програму, що працює на стороні веб-сервера й використовує веб-браузер у якості клієнта. Серверна частина, отримавши запит від клієнта, виконує необхідні операції та формує веб-сторінку, яка за допомогою протоколу HTTP відправляється клієнту для відображення інформації користувачеві. Веб-додатки є інтерактивними, розробленими для мережі інтернет, програмами. [2] Сторінки веб-додатку зазвичай є динамічними – їх контент повністю або частково залежить від запиту, сформованого на основі дій користувача. Вигляд статичних сторінок завжди залишається однаковим, тобто сервер відправляє такі сторінки веб-браузеру без внесення будь-яких змін. [4]

Для веб-додатків створюються набори даних, що зберігаються у вигляді таблиць – бази даних, які надають необхідну інформацію у відповідь на запит сервера. Запити до бази даних містять критерії пошуку та формуються за допомогою мови структурованих запитів SQL. [4]

Веб-додатки зазвичай виконують такі функції, як отримання даних від користувача та їх зберігання на сервері; виконання різноманітних дій на запит користувача: додавання, видалення, редагування даних в базі даних,

виконання складних обчислень; аутентифікація користувача та відображення відповідного інтерфейсу системи; відображення інформації, що постійно змінюється. [6, с.12]

Для веб-додатку на стороні сервера можуть застосовуватися різні мови програмування, найпопулярніші – PHP, Python, Java, C#. Для написання коду клієнтської частини використовуються HTML, CSS та JavaScript.

Взаємодія клієнтської та серверної частин зображена на Рисунку 1.1 – Схема взаємодії клієнтської та серверної частин веб-додатку. [8]

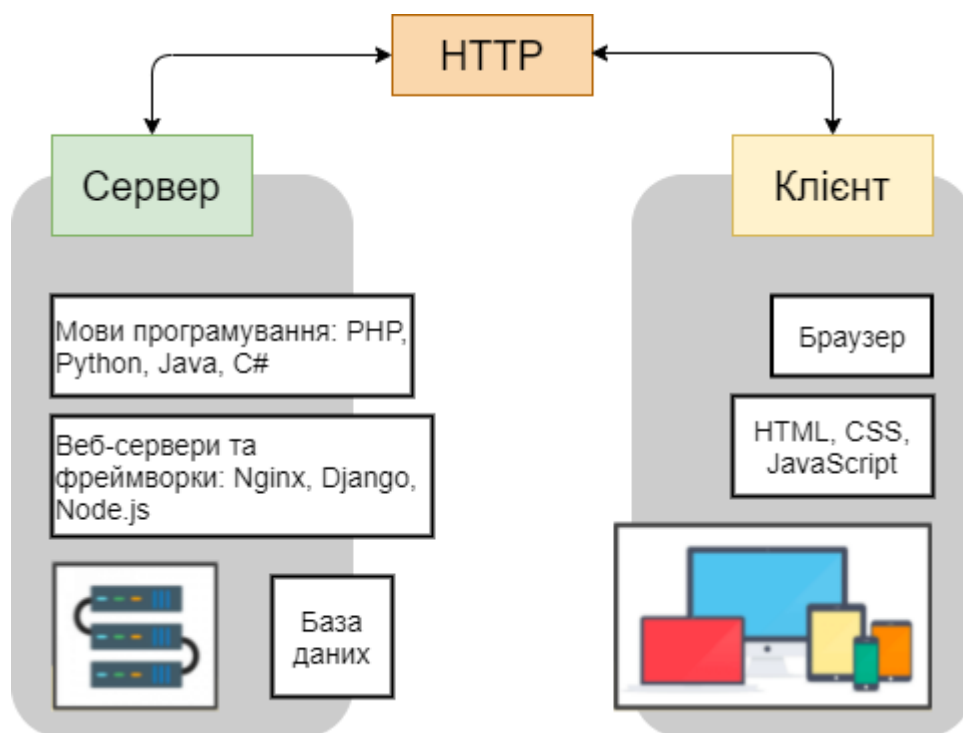


Рисунок 1.1 – Схема взаємодії клієнтської та серверної частин веб-додатку

Веб-додатки дуже зручні тим, що не потребують встановлення на комп'ютер – достатньо наявності браузера та доступу до мережі Інтернет. Крім цього, вони зазвичай є кросплатформними – практично не мають вимог до апаратного забезпечення та ресурсів. Оскільки веб-додатки знаходяться на сервері, то в користувачів не виникає потреби їх періодично оновлювати. Це важливо для шкіл, що не мають адміністраторів, які б займалися підтримкою.

Веб-додатки інтуїтивно більш зрозумілі для школярів та вчителів, адже кожен знає, як користуватися браузером. Натомість для ознайомлення з функціоналом десктопних додатків учні можуть потребувати допомоги вчителя, що витрачатиме час уроку. З боку вчителів також будуть витрати часу для отримання навичок роботи з додатком.

Для спрощення та прискорення проектування структури веб-додатку використовуються архітектурні шаблони (патерни), які описують перелік програмних компонентів та правила взаємодії між ними. Існує багато таких патернів, кожен з яких має своє призначення та переваги в певному контексті: Model-View-Controller (MVC), багаторівнева архітектура, канали та фільтри, клієнт-сервер, архітектура на основі мікросервісів та інші. [10] Найпопулярнішим із них є MVC, він лежить в основі багатьох бекенд-фреймворків.

Згідно з концепцією MVC, виділяються три фундаментальні логічні компоненти, кожен з яких виконує свою роль та відповідає за певні функції, а також безперервно взаємодіє з двома іншими. [12] Основна ціль цього шаблону – чітко відокремити реалізацію логіки додатку (функції, класи, скрипти) та його зовнішній вигляд (розмітку сторінки). [14] І хоча реалізація цієї концепції може відрізнятись в залежності від потреб конкретного додатку, основні принципи розподілу логіки між компонентами залишаються сталими.

Компонент Model відповідає за функціональну бізнес-логіку додатку та не має візуального інтерфейсу, адже безпосередньо не взаємодіє з користувачем. [16] Він містить опис даних, отриманих з бази, виконує їх кінцеву перевірку та обробку, а також взаємодіє з базою даних – через Model відбувається додавання, видалення та редагування записів у таблицях. [12] Отримані за запитом користувача дані надаються компоненту Controller. Model є незалежним від View та Controller, які, навпаки, повністю залежать

від нього. Тобто Model не повинен мати ніякої інформації про те, як будуть відображатися дані, їх модель незмінна, змінюється лише представлення. [18]

Компонент View відповідає за візуалізацію, реалізує спосіб представлення даних, а отже, в більшості випадків їх декілька. Необхідний для конкретної ситуації View обирає Controller. Дані View отримує від Model, адже не має безпосереднього доступу до бази даних та не може впливати на неї. У випадку динамічних сторінок View зазвичай містить не лише статичну розмітку, але й деякий код, в який закладена бізнес-логіка, адже HTML не має засобів для роботи з даними. View також взаємодіє з користувачем – збирає введену ним інформацію та передає на обробку. [16,18,20]

Компонент Controller відповідає за обробку змін інформації. Реагуючи на запити користувача при натисканні на елементи інтерфейсу, він виконує операції над Model та викликає відповідний View. Таким чином, Controller є сполучною ланкою між компонентами Model та View, при цьому, містить виключно організаційну логіку. Компонентів Controller може бути декілька, кожен із яких взаємодіє лише з одним View та Model. Серед декількох Controller'ів є один головний, який викликає інші в залежності від потреби. [10,16,20]

Схема взаємодії компонентів архітектурного шаблону MVC наведена на Рисунку 1.2 - Схема взаємодії компонентів патерна MVC. [85]

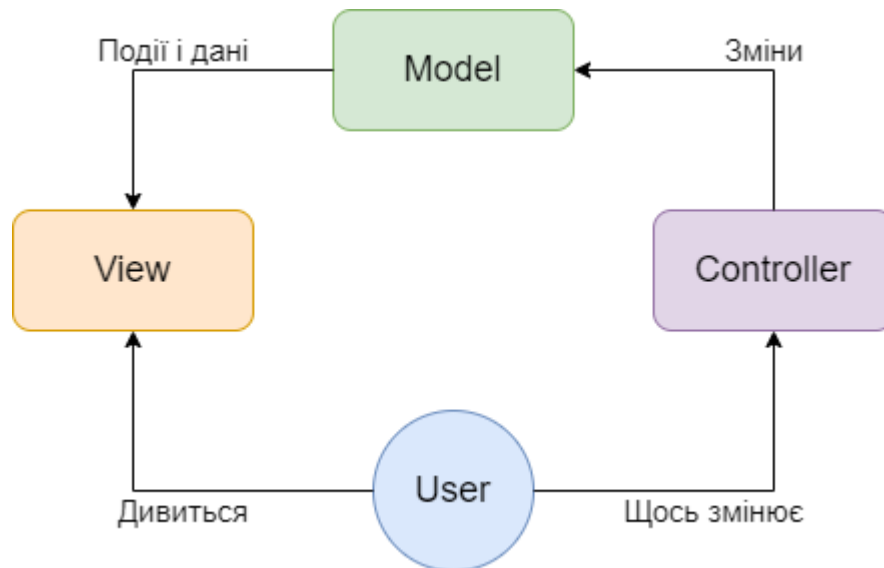


Рисунок 1.2 - Схема взаємодії компонентів патерна MVC

Існує декілька модифікацій шаблону MVC: HMVC (Hierarchical MVC), MVVM (Model-View-ViewModel), MVP (Model-View-Presenter).

Однією з основних переваг використання патерна MVC є структуризація коду. Завдяки чіткому розподілу бізнес-логіки та логіки візуалізації, код стає більш зрозумілим для сприйняття, а значить, полегшується його підтримка, тестування, а також локалізація проблем. Спрощується внесення змін, наприклад: модифікація коду, що відповідає за візуалізацію, не обов'язково тягне за собою необхідність змінювати бізнес-логіку і навпаки. Крім цього, з'являється можливість повторного використання коду. [10,12]

У деяких випадках використання концепції MVC може ускладнити архітектуру додатку, наприклад, коли він складається з великої кількості функціональних модулів. Тоді кожний такий модуль повинен мати три компоненти, адже їх наявність чітко визначена. Аналогічна проблема може виникнути при намаганні розширити функціонал додатку. Трапляються ситуації, коли є декілька Model, над кожною з яких необхідно виконувати дії. Це може призвести до небажаних залежностей. Проте, такі ситуації зазвичай не виникають при проектуванні додатків, для яких використання даного

шаблону повністю виправдане, адже MVC, як і будь-який інший шаблон, не є єдиним, а лише одним із можливих рішень. [12]

### **1.3 Огляд існуючих веб-додатків для інтерактивного навчання**

У результаті аналізу наявних у загальному доступі веб-додатків, було виявлено, що майже всі вони не мають ніяких елементів інтерактивності. Більшість побудовані на питаннях з однією або декількома відповідями. Деякі з таких питань містять зображення та схеми, що безперечно робить процес тестування більш цікавим, проте, взаємодіяти з ними користувач не може. Додатки, які призначені не тільки для тестування, а й для навчання, найчастіше містять викладення матеріалу у вигляді онлайн-підручника або презентацій, що дає можливість лише пасивно сприймати інформацію – слухати розповідь учителя або опрацьовувати тему самостійно.

[naurok.com.ua](http://naurok.com.ua) – безкоштовний веб-додаток з досить широким функціоналом. Містить тестові запитання з однією або декількома відповідями, а також завдання на відповідність. Дає вчителям змогу самостійно створювати тести з різних предметів для всіх класів за темами та додавати в них зображення. Крім цього, містить велику кількість допоміжного матеріалу для підвищення кваліфікації вчителів – дає можливість відвідувати та переглядати вебінари, брати участь в онлайн конференціях, ділитися своїми та ознайомлюватися з розробками колег. Не менш важливою є наявність навчальних відео з роботи з функціоналом додатку для вчителів. Веб-додаток є перспективним та постійно вдосконалюється, але на даний момент він має більшу цінність для вчителів, ніж для учнів. Приклад інтерфейсу цього веб-додатку наведений на Рисунку 1.3 – Інтерфейс веб-додатку [naurok.com.ua](http://naurok.com.ua).

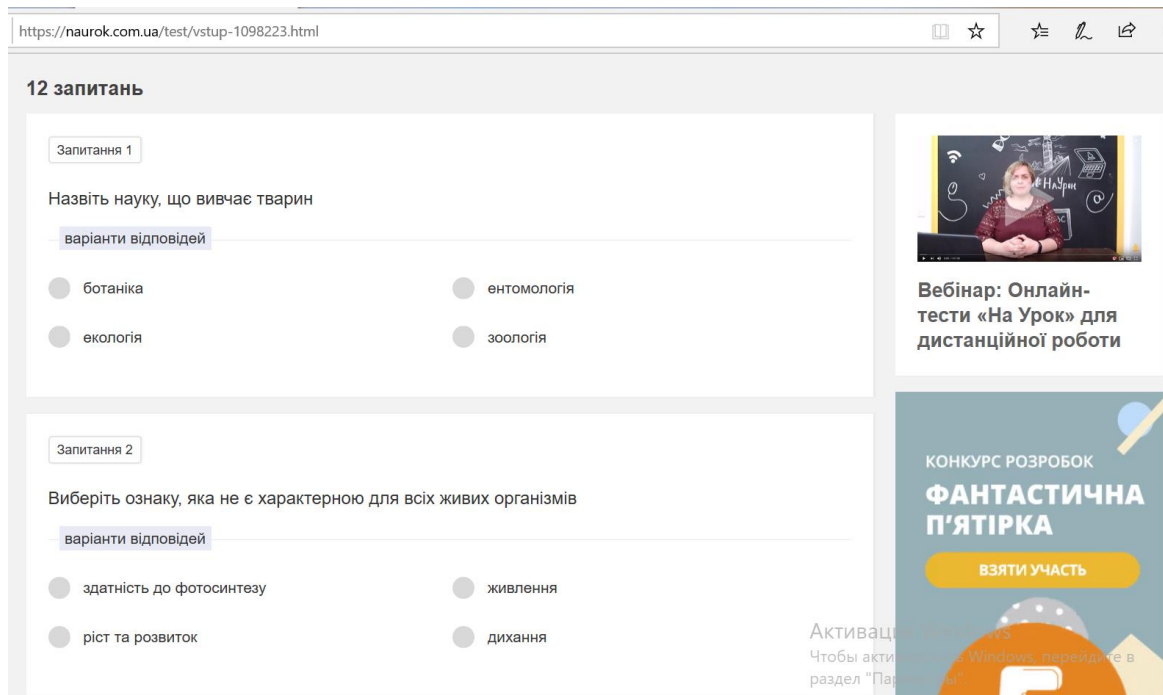


Рисунок 1.3 – Інтерфейс веб-додатку naurok.com.ua

ukrerudyt.com – безкоштовний веб-додаток з тестовими питаннями з різних предметів для різних класів, а також для підготовки до ЗНО. Функціонал додатку обмежується тестами та вікторинами, а також посиланнями на сайт, де можна придбати книги для підготовки до ЗНО. Інтерфейс веб-додатку спрощений, кількість тестів невелика, можливості додавати нові немає. Після проходження тесту відображається сторінка для перегляду правильних та неправильних відповідей з виділенням зеленим та червоним кольором відповідно. Кольори підібрані невдало – вони є занадто яскравими та викликають дискомфорт в очах. Приклад інтерфейсу цього веб-додатку наведений на Рисунку 1.4 – Інтерфейс веб-додатку ukrerudyt.com.

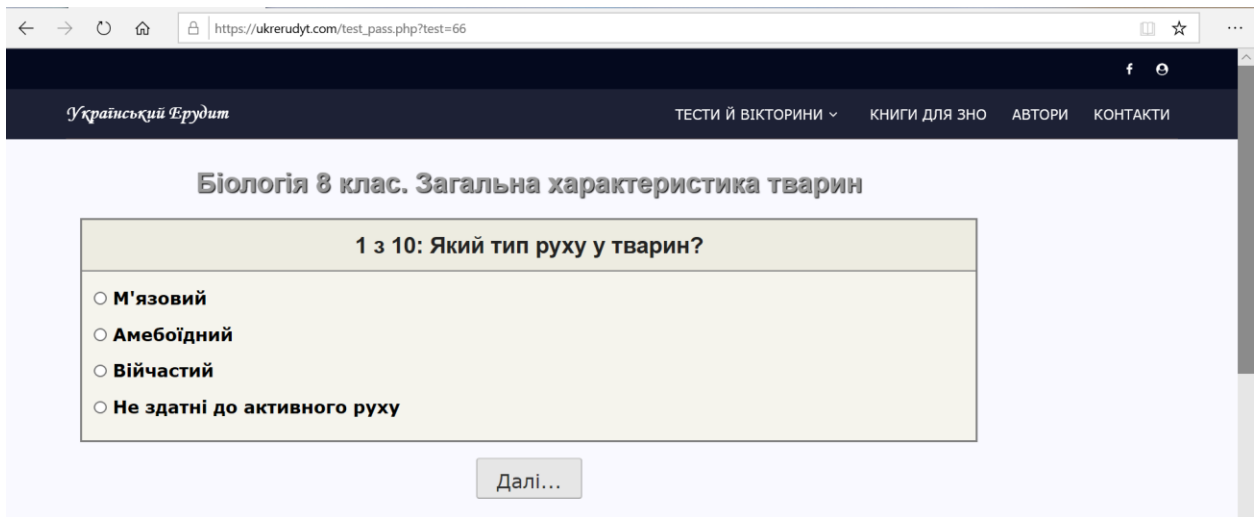


Рисунок 1.4 – Інтерфейс веб-додатку ukrerudyt.com

[miyklas.com.ua](http://miyklas.com.ua) – веб-додаток зі зручним інтуїтивно зрозумілим інтерфейсом та елементами інтерактивності. Має тести майже для всіх предметів за всіма темами, що входять до шкільної програми, з 1 по 11 клас. У випадках, коли це доречно, тести містять зображення, але можливості взаємодіяти з ними веб-додаток не надає. Відмінною рисою веб-додатку є те, що кожна тема містить не лише питання, а ще й теоретичні відомості, з якими пропонується ознайомитися перед початком проходження тестування. Матеріал оформлений дуже продумано – важливі для запам'ятовування визначення виділені окремим блоком, додані ілюстрації. Для вчителів є розділ з методичними рекомендаціями. Приклад інтерфейсу цього веб-додатку наведений на Рисунку 1.5 – Інтерфейс веб-додатку miyklas.com.ua.





Рисунок 1.5 – Інтерфейс веб-додатку miyklas.com.ua

[testy-bio.pp.ua](http://testy-bio.pp.ua) – веб-додаток з простим дизайном, який на деяких сторінках є не дуже зручним. Пропонує тести лише з біології та за обмеженою кількістю тем без можливості додавання власних. Відсутня авторизація, тому результати тестування не зберігаються. Серед переваг можна виділити те, що питання є різноплановими, деякі з них можна назвати інтерактивними. Наявні питання з однією та декількома правильними відповідями, питання типу «доповнити речення», «встановити відповідність», «показати на рисунку», «так/ні». При належному доопрацюванні та розширенні функціоналу веб-додаток може стати корисним та цікавим для учнів доповненням на уроках. Приклад інтерфейсу цього веб-додатку наведений на Рисунку 1.6 – Інтерфейс веб-додатку testy-bio.pp.ua.

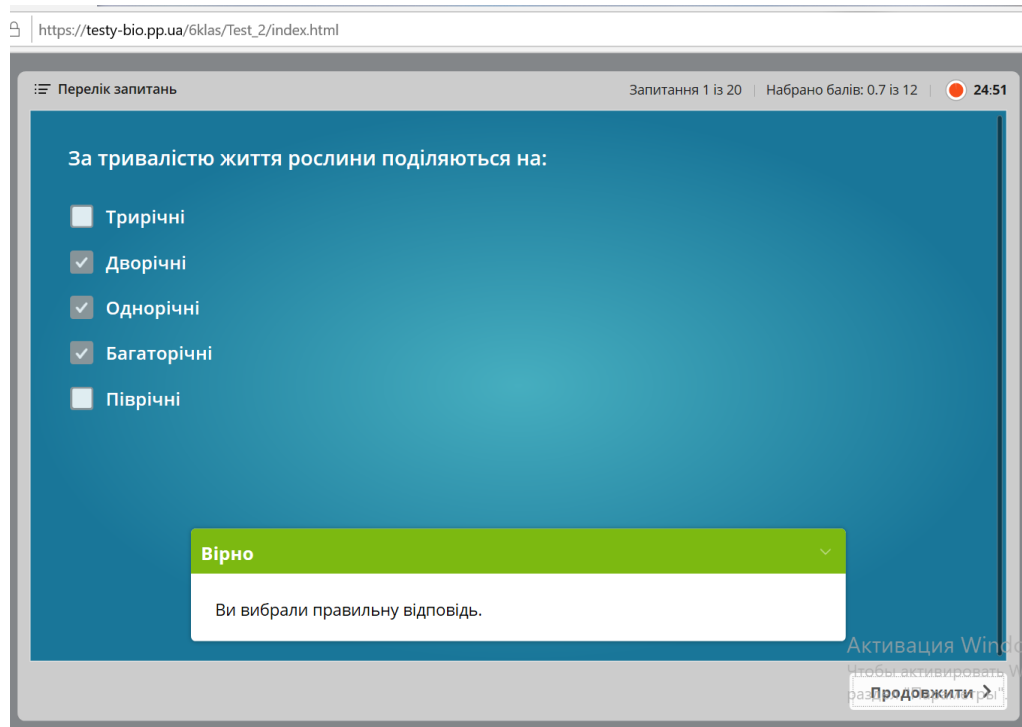


Рисунок 1.6 – Інтерфейс веб-додатку testy-bio.pp.ua.

[onlinetestpad.com](https://onlinetestpad.com) – веб-додаток з тестами з різних предметів для класів з 5 по 11. Недоліком є те, що сервіс не є виключно освітнім, наявні також розважальні тести, тому контент потрібно обов'язково фільтрувати. Є можливість переглянути тести з певного предмету для певного класу, але відсутній розподіл за темами. Через нагромадження тестів та реклами веб-додаток не є привабливим у використанні. Більшість питань стандартні, але ілюстровані, є також питання на встановлення відповідності. Приклад інтерфейсу цього веб-додатку наведений на Рисунку 1.7 – Інтерфейс веб-додатку [onlinetestpad.com](https://onlinetestpad.com).

https://onlinetestpad.com/ua/test/250811-uzagalnennya-procesi-zhittediyalnosti-tvarin

Укажіть інші ознаки тварини, яка має зображену будову покриву тіла

Кровоносна система	
Тип нервової системи	1 Земноводні
Кровоносна система	2 двокамерне серце, два кола кровообігу
Клас	3 запліднення зовнішнє, розвиток прямий
Розмноження і розвиток	4 трикамерне серце, два кола кровообігу
	5 двокамерне серце, одне коло кровообігу
	6 вузлова
	7 Плазуни
	8 Кісткові риби
	9 дифузна
	10 трубчаста
	11 запліднення внутрішнє, розвиток прямий
	12 запліднення внутрішнє, розвиток непрямої

Далі

Активация Windows  
Чтобы завершить настройку, перейдите в раздел "Параметры".

Рисунок 1.7 – Інтерфейс веб-додатку onlinetestpad.com

[testorium.net](http://testorium.net) – веб-додаток з оригінальним дизайном та великим вибором тестів з різних предметів, розподілених за темами та обмежених за часом виконання. Є можливість реєстрації та входу до особистого кабінету, де можна переглянути результати тестування. Деякі питання ілюстровані, крім звичайних тестів, є завдання на встановлення відповідності. Наявні також тести для підготовки до ЗНО. Недоліком можна назвати те, що всі тести відображаються списком, а через велику їх кількість список є дуже довгим. Переглянути список тестів по окремому предмету можна лише ввівши в поле для фільтрації його назву, що є не дуже зручним. Приклад інтерфейсу цього веб-додатку наведений на Рисунку 1.8 – Інтерфейс веб-додатку testorium.net.

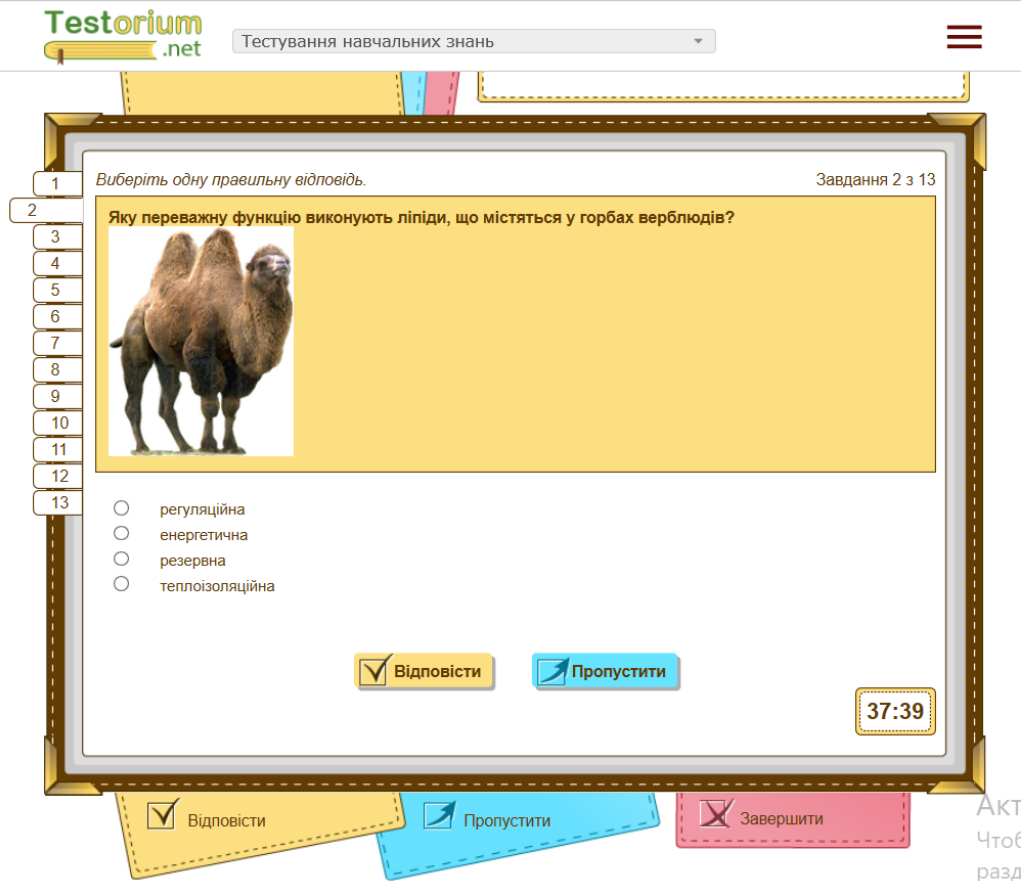


Рисунок 1.8 – Інтерфейс веб-додатку testorium.net

Отже, наразі є необхідність створення веб-додатку для вчителів та учнів, який би мав елементи інтерактивності та міг використовуватися на уроках для закріплення та перевірки знань. Для кращого засвоєння учнями нового матеріалу завдання повинні бути ілюстровані, містити схеми, рисунки, таблиці, які б давали можливість взаємодіяти з ними. Веб-додаток має бути цікавим та пізнавальним для школярів, а вчителям допомагати будувати урок на постійній взаємодії між учасниками, забезпечуючи більшу залученість та вмотивованість учнів. При цьому має бути дотриманим правило, що програма не є заміною вчителя, а лише його помічником. Тобто вона не повинна містити підручник чи аудіозапис з викладенням теми. Матеріал має бути структурований та візуалізований, що підвищить рівень засвоєння. Завдання ознайомлення залишається на стороні вчителя.

## 1.4 Постановка задачі

Спроекувати та створити базу даних, використовуючи СУБД MySQL.

Спроекувати та розробити інформаційну веб-систему на основі фреймворку Yii для забезпечення інтерактивності навчального процесу в загальноосвітніх школах, яка виконує функції:

- авторизації учнів та вчителів за попередньо створеними адміністратором акаунтами;
- розподілу практичних робіт за класами з обмеженням доступу до деяких завдань для учнів;
- відображення правильних та неправильних відповідей на кожне питання;
- розрахунку успішності проходження тесту у відсотках;
- збереження оцінок учня в його особистому кабінеті;
- збереження оцінок учнів у журналі для вчителя;
- адміністрування користувачів для користувача з роллю «адміністратор»;
- адміністрування практичних робіт для користувачів з роллю «вчитель».

Результатом роботи має бути протестована інформаційна веб-система з декількома варіантами завдань для практичних робіт, реалізованих засобами мови JavaScript, з можливістю перевірки правильності їх виконання, яка має зручний інтерфейс і добре продуманий дизайн.

## 2 ВИБІР МЕТОДІВ ВИРІШЕННЯ ЗАДАЧІ

### 2.1 Вибір методів розробки

Перед розробником, який збирається створювати програму, постає питання вибору стратегії: писати код повністю самостійно чи використовувати фреймворки. Кожен з підходів має свої переваги та недоліки, тому цей вибір іноді буває досить складним.

Написання вихідного коду з нуля дозволяє розробникові реалізовувати будь-який функціонал майже без обмежень, отримуючи при цьому гарантовано унікальний продукт, який повністю налаштований та оптимізований під потреби конкретного проекту. Проте розробка, тестування та підтримка такої програми може бути дуже трудомісткою. [22,24]

Останнім часом багато веб-розробників віддають перевагу використанню фреймворків – бібліотек, що значно спрощують розробку веб-додатків. Ці бібліотеки являють собою набори файлів, що містять об'єкти, функції, підпрограми, які можуть бути використані в процесі розробки. [26] Фреймворки можна класифікувати так: фреймворки додатків, фреймворки програмних модулів та фреймворки концептуальних моделей. [22]

Використання фреймворку значно прискорює розробку, оптимізуючи витрати часу й ресурсів, та підвищує якість кінцевого продукту, адже готові рішення реалізовувалися великою кількістю розробників і є стабільними та добре протестованими на різних платформах. Крім цього, з розробників знімається відповідальність за виправлення певної частини помилок, тому що вони вирішуються з регулярними оновленнями версії фреймворку. Масовість використання фреймворків дає розробникові можливість розраховувати на документацію, яка зазвичай є досить детальною, а також на допомогу більш досвідчених колег. При реалізації типових задач програмування та проектування застосовуються кращі практики, тому необхідність проведення рефакторингу скасовується – завдяки фреймворкам можна відразу отримати

максимально чистий код. [24,26]

За специфікою рішення задач можна виділити такі категорії фреймворків: `backend`, `frontend`, `full stack`. `Backend`-фреймворки працюють на стороні сервера та орієнтовані на такі функції, як проектування баз даних, зберігання та захист даних, забезпечення логіки роботи додатку. Натомість `frontend`-фреймворки ніяк не пов'язані з логікою, вони призначені для створення інтерфейсу користувача, тобто містять готові стилі й розмітку, які дозволяють швидко розробити дизайн сайту та зробити його адаптивним. `Full stack`-фреймворки є універсальними інструментами – вони ефективно працюють на сторонах `backend` та `frontend`, використовуючи при цьому єдину мову програмування для серверної та клієнтської частин. [28]

Оскільки фреймворки покликані забезпечувати готові рішення для різних потреб різних користувачів, то з'являється проблема великої кількості непотрібного коду. Також орієнтованість на масове використання означає, що потрібно буде витратити час на певні налаштування під потреби конкретного проекту. Додаткових часових витрат вимагатиме й освоєння принципів роботи фреймворку, адже по суті це результат роботи розробника, і майже всі фреймворки відрізняються один від одного. Вихідний код фреймворків зазвичай є відкритим, що створює проблеми із захищеністю додатків від зовнішніх загроз. [24,26,30]

З огляду на описані вище переваги й недоліки, можна зробити висновок, що рішення про використання фреймворку потрібно приймати для кожного проекту індивідуально. Для невеликих проектів відмова від них може стати виправданою, адже вбереже від надмірного ускладнення. Так само для додатків зі складною логікою і великою кількістю модулів фреймворк може створити проблеми збільшення об'єму коду та додаткової інтеграції сторонніх компонентів. Натомість для додатків з логікою середньої складності фреймворки можуть стати вигідним рішенням, яке зекономить час та зусилля.

## 2.2 Вибір засобів розробки

Frontend – це публічна частина веб-додатків, з якою користувач взаємодіє безпосередньо, відповідно в рамках frontend-розробки виконується веб-верстка та забезпечується виконання кожним візуальним елементом своїх функцій. [32] Для створення публічної частини веб-додатків розробники використовують HTML, CSS та JavaScript. HTML – мова гіпертекстової розмітки, за допомогою якої створюється структура та виконується форматування контенту. CSS – мова, що відповідає за зовнішній вигляд сторінки, дозволяє змінювати кольори, шрифти, розташування блоків та ін. JavaScript – мова, що допомагає реалізувати виконання різних дій на сторінці, таких, як анімація та відповідь на запити користувача. [34] Багато задач frontend-розробки вирішуються за допомогою frontend-фреймворків, які значно прискорюють та спрощують роботу.

Bootstrap – один із найпопулярніших HTML, CSS і JavaScript фреймворків з відкритим кодом, що використовується для верстки адаптивних веб-додатків. [36] Складається з великої кількості готових компонентів: веб-форми, кнопки, сітка, блоки навігації, таблиці, пагінація, модальні вікна, каруселі та ін. Ці елементи гармонійно поєднуються між собою, що дозволяє створювати сторінки в єдиному стилі. [38] Всі компоненти добре продумані та протестовані на різних платформах і пристроях. Bootstrap надає можливість гнучкого налаштування цих компонентів, що дозволяє підключати лише ті, які будуть необхідні для даного проекту. Простота освоєння Bootstrap прискорює верстку якісних сайтів навіть для розробників з невеликим досвідом, адже фреймворк має детальну документацію з безліччю прикладів. Крім цього, в Інтернеті можна знайти велику кількість уроків та інструкцій або звернутися за допомогою до досвідчених колег зі спільноти. Bootstrap продовжує активно розвиватися у відповідності до сучасних тенденцій веб-розробки, виходять нові версії з новим функціоналом та виправленими помилками. [40] Звідси випливає



недолік – для того, щоб веб-додатки на Bootstrap мали коректне відображення, потрібно мати нову версію браузера. Цей фреймворк не підходить у випадках, коли необхідно розробити веб-додаток з унікальним дизайном, адже він буде схожим на багато інших. [38] Спроба використання Bootstrap у таких випадках може призвести до значного переписування коду. Css та js-файли фреймворку мають досить великий розмір, що є наслідком універсальності Bootstrap. [36] Відокремити блоки, що не використовуються, розробникам-початківцям буде дуже складно. [42] Отже, Bootstrap – інструмент для створення якісного сайту зі стандартних блоків, який значно прискорює розробку та може бути успішно застосований на проектах, де унікальність не відіграє вирішальну роль.

Foundation – багатофункціональний адаптивний frontend-фреймворк з відкритим кодом, що підтримує широкий ряд платформ та пристроїв. Дозволяє реалізувати детально продумане мобільне представлення сайту завдяки великій кількості класів видимості, що надають більше гнучкості та управління. Foundation містить розширену колекцію готових компонентів користувацького інтерфейсу, додаткові функції, що надаються JavaScript, а також шаблони та фрагменти коду, серед яких система адаптивних зображень, цінова таблиця, перевірка форми, адаптивні вбудовування та ін. Фреймворк має інструменти для рішення майже всіх інтерфейсних задач (для сайтів та електронної пошти) та поставляється з інтерфейсом командного рядка. Через занадто велику кількість можливостей налаштувань робота з Foundation може бути досить складною, а різноманітність варіантів реалізації ускладнює вибір, затримуючи роботу над проектом. Фреймворк не є таким популярним, як, наприклад, Bootstrap, відповідно, має невелику спільноту, що також ускладнює його освоєння для розробників-початківців. [42,44]

Materialize – сучасна адаптивна CSS-платформа з відкритим кодом від корпорації Google, яка містить багато готових інтерактивних компонентів та вдосконалені функції JavaScript, наприклад, вбудовані анімації та переходи

для покращення взаємодії з користувачем. Також має готові теми та шаблони, такі, як портфоліо, таблиця цін, відгуки, розділ FAQ, форми зворотного зв'язку. Фреймворк пропонує розширені налаштування кольорів, різноманітні колірні схеми, шрифти від Google та інтеграцію з картами Google. Materialize є досить простим для освоєння, адже він орієнтований на досвід користувачів і використовує принципи матеріального дизайну, має детальну документацію, а також легко розширюється. Фреймворк має чітко визначену мову дизайну, що не підходить для нестандартних інтерфейсів. Іншим недоліком є те, що Materialize є невеликим проектом та практично не має корпоративної підтримки. [84,46]

Semantic UI – гнучкий адаптивний фреймворк, який є дуже зручним для мобільних пристроїв. Заснований на принципах звичайної англійської мови, тому дуже легкий для сприйняття, має детальну, зручно організовану по всім компонентам документацію. Надає безліч простих й елегантних дизайнів, тематичних ефектів, пропонує різноманіття варіантів користувацького інтерфейсу, які за необхідності можна легко кастомізувати. Semantic UI може бути легко інтегрований зі сторонніми бібліотеками й деякими популярними фреймворками. Іншими безперечними перевагами фреймворку є можливість завантаження лише необхідних для даного проекту компонентів, що значно зменшує розмір файлу, а також спрощене налаштування виведення інформації по налагодженню в консоль. Для роботи із Semantic UI необхідні базові знання JavaScript, тому він підходить не для всіх розробників. Має більш суворі правила кодування в порівнянні з іншими фреймворками. [48-52]

Backend – це програмно-апаратна частина веб-додатку, що відповідає за його логіку, працездатність і правильне функціонування, знаходиться на сервері та є невидимою для користувача. [53,54] Вона включає такі функції, як доступ до даних, управління базами даних, аутентифікація. [55] В рамках backend-розробки продумується архітектура веб-додатку і зв'язок між його

компонентами, виконуються налаштування бази даних, розробляється програма, яка безпосередньо буде формувати сторінки у відповідь на запити користувачів, підтримується безпека. Backend-розробники зазвичай застосовують мови програмування PHP, Python, Ruby, Java. [56] Для скорочення часу на реалізацію типових задач та отримання в результаті максимально надійного, безпечного і якісного веб-додатку використовують backend-фреймворки.

Yii – популярний об'єктно-орієнтований компонентний full stack php-фреймворк, що реалізує парадигму MVC. Є універсальним – підходить майже для будь-якого типу веб-додатків від звичайного блогу або сайту-візитівки до великого порталу, але використовується зазвичай для створення більш масштабних проектів, таких, як форуми; інтернет-магазини з великою кількістю товарів, розділів та категорій; веб-сервіси по наданню послуг; CMS-системи управління контентом; eCommerce додатки. Yii є дуже гнучким та має високу продуктивність, він надає широкі можливості для реалізації складного функціоналу. Крім готових реалізацій таких базових функцій, як аутентифікація й авторизація, побудова форм та їх валідація, фреймворк пропонує велику кількість розширень на Github, які можна встановити через Composer, та надає можливість інтеграції з іншими сторонніми системами та бібліотеками, що значно розширює функціонал. Yii забезпечує автоматичне тестування – перехоплює помилки у випадку їх наявності, обробляє та пропонує рішення. Фреймворк підтримує різні бази даних (SQLite, MySQL, SQL Server, Oracle та ін.) і має інтерфейси для роботи з ними. Надійна система безпеки попереджає атаки через cookie, міжсайтові запити, SQL-ін'єкції при реєстрації та авторизації користувачів. Yii надає візуальний інтерфейс Gii для автоматичної генерації класів Model, Controller та View, які за необхідності можуть бути змінені відповідно до потреб проекту. Фреймворк є досить простим в освоєнні та має дуже детальну документацію, перекладену різними мовами, а його велика популярність дає можливість

розраховувати на велику кількість допоміжних матеріалів в Інтернеті та поради від спільноти. Її є добре протестованим та активно підтримується. Окремою перевагою є підтримка локалізації. Її використовує вбудований Bootstrap, від якого буде дуже складно відмовитися в разі необхідності, тому іноді це є досить серйозним недоліком. Крім цього, фреймворк має не дуже гнучку систему маршрутизації. Корисні оновлення Її виходять досить рідко, тому в певній мірі він є застарілим. Фреймворк може бути складним для освоєння у випадку переходу з іншого, адже тоді, скоріше за все, доведеться повністю змінювати підхід до розробки. [57-60]

Laravel – php-фреймворк з відкритим кодом, що створений на основі компонентів іншого фреймворку – Symfony. Laravel реалізує архітектурний шаблон MVC, що забезпечує мінімальний зв'язок між компонентами системи, а значить, більшу стабільність роботи веб-додатку. Фреймворк має простий, зрозумілий, оптимізований та ретельно протестований код, який також є добре документованим, що спрощує його підтримку та забезпечує високу продуктивність веб-додатків. Laravel використовує об'єктно-орієнтований підхід до розробки, що запобігає дублюванню коду. Вбудовані бібліотеки та модулі допомагають зекономити час на реалізації типових задач, наприклад, фреймворк пропонує готову надійну систему реєстрації та аутентифікації користувачів. При цьому, Laravel залишає для розробника можливість за потреби змінювати компоненти, орієнтуючись на потреби проекту. Серед інших переваг фреймворку – гнучка система маршрутизації, підтримка локалізації, зручний механізм обробки помилок, кешування. Основна частина документації Laravel написана англійською мовою, тому необхідне її знання на достатньому рівні. Оновлення фреймворку іноді створює додаткові проблеми, на вирішення яких доводиться витратити певні ресурси. Крім цього, розташування деяких каталогів та файлів є нелогічним, що створює надлишкові непорозуміння. [61-63]

Symfony – php-фреймворк з відкритим кодом для швидкої розробки веб-додатків, що складається з набору непов'язаних компонентів, які можуть бути зібрані за допомогою Composer. Відсутність залежностей між компонентами спрощує внесення змін та запобігає виникненню додаткових проблем в інших компонентах при зміні одного. Symfony також надає можливість інтеграції зі сторонніми додатками та інструментами для frontend-розробки. Фреймворк заснований на паттерні MVC та має добре продуману чітку архітектуру, що полегшує підтримку веб-додатку та розуміння його коду. Symfony пропонує готові рішення для роботи з веб-формами, базами даних, email поштою і т.д. В основу цих рішень покладені кращі практики веб-розробки, що забезпечує високу функціональність та надійність. Підтримує різні бази даних, наприклад, MySQL, MariaDB, PostgreSQL, SQLite та ін. Для підвищення продуктивності процесу розробки Symfony пропонує панель інструментів налагодження та панель профілювальника. Фреймворк активно розвивається та підтримується, є стабільним та добре протестованим. Symfony має розгорнуту документацію, проте, є досить складним в освоєнні. Крім цього, продуктивність фреймворку іноді є досить низькою, що може бути помітним на прикладі веб-додатків із високим навантаженням у реальному часі. [64-67]

Zend – об'єктно-орієнтований php-фреймворк з відкритим вихідним кодом, який добре підходить для розробки комерційних веб-додатків та надає готові рішення для різноманітних задач. Базується на модульності та слабкій зв'язаності компонентів, використовує Composer у якості менеджера залежності пакетів. Такий підхід дозволяє повторно використовувати код, робить його чистим та зрозумілим для усіх членів команди розробників. Zend також надає можливість простої інтеграції з різними структурами й компонентами, а також видалення непотрібних. Фреймворк є стабільним і надійним – забезпечує стандартне шифрування та валідацію форм. Підтримує бази даних: Oracle Database, MariaDB, Microsoft SQL Server, MySQL,

PostgreSQL, SQLite та ін. Zend використовує архітектурний шаблон MVC, що дозволяє розділити бізнес-логіку та представлення, роблячи код більш структурованим. В порівнянні з іншими фреймворками, швидкість роботи Zend нижча. Фреймворк має велику спільноту розробників та розгорнуту документацію, проте, більшість матеріалів англійською мовою. Zend досить складний для освоєння, для його використання бажано мати досвід роботи з іншими аналогічними фреймворками. [68,69]

CodeIgniter – невеликий php-фреймворк з потужним набором функціональних можливостей для більш швидкої розробки веб-сайтів та веб-додатків. Заснований на архітектурному шаблоні MVC та має слабо пов'язані між собою компоненти, які не залежать один від одного. CodeIgniter має високу продуктивність завдяки вбудованому компоненту, що дозволяє кешувати сторінки. Безпека фреймворку забезпечується шифруванням даних, перевіркою форм і даних, XSS-фільтрацією та ін. CodeIgniter надає величезну кількість бібліотек з допоміжними функціями й класів, що містять прості рішення майже будь-яких задач розробки. Фреймворк дуже легко налаштовується й має зрозумілий синтаксис, його структура не вимагає суворих правил кодування. Документація є розширеною та зрозумілою, містить приклади коду. Недоліками CodeIgniter є те, що він рідко оновлюється та привчає до неправильного стилю програмування, наприклад, дозволяє повністю відмовитися від Model та місцями використовує процедурний код. Фреймворк є вдалим вибором для невеликих проектів, але не підходить для масштабних. [70-72]

Phalcon – ефективний та лаконічний php-фреймворк з відкритим вихідним кодом, написаний мовою C. Має дуже високу продуктивність та використовує мало ресурсів, підходить для середніх за масштабом проектів. Заснований на архітектурному шаблоні MVC і пропонує набір не пов'язаних між собою компонентів, які можуть бути використані окремо. Один із таких компонентів, наприклад, надає функції безпеки, такі, як хешування паролів,

захист від підробки міжсайтових запитів. Phalcon підтримує реляційні та нереляційні бази даних, має PhalconQL – діалект SQL, що стандартизує запити SQL для СУБД, які підтримує Phalcon. Фреймворк швидко розвивається та має розгорнуту документацію. Phalcon не є складним в освоєнні, проте потребує знання мови С, інакше можуть виникнути труднощі як із пошуком та виправленням дефектів, так і з розширеннями, необхідними для потреб конкретного проекту. [73-75]

SakePHP – php-фреймворк, заснований на концепції MVC, що має відкритий код та пропонує великий вибір бібліотек з готовими рішеннями найбільш розповсюджених задач. Надає компоненти безпеки, авторизації, генерації та заповнення форм, управління сесіями, обмеження доступу, електронної пошти, представлення деревовидної інформації, файлів cookie, обробки запитів, пагінації та ін. SakePHP має функцію генерації всього коду за схемою бази даних та CRUD-сторінок для сутностей, але це можливо лише при дотриманні певних стандартів іменування стовпців. Фреймворк також надає вбудовану верифікацію даних, кешування, скаффолдинг для автоматичного створення сторінок, підтримує локалізацію. SakePHP має низький поріг входження, активну спільноту розробників, є досить простим для освоєння. Зрозумілість та елегантність коду фреймворку забезпечуються завдяки вимогам дотримання угод при кодуванні, що для декого з користувачів може бути недоліком. [76-78]

У процесі аналізу frontend- та backend-фреймворків були виділені характеристики, які є бажаними для засобу програмування, що буде використовуватися при розробці веб-додатку:

- наявність розгорнутої документації з прикладами коду, інших навчальних матеріалів (відеоуроки, статті);
- можливість автоматичної генерації коду та CRUD-сторінок;
- підтримка локалізації;
- наявність вбудованого frontend-фреймворку;

- наявність компоненту для управління доступом;
- надійність, продуктивність;
- чітко визначена та продумана архітектура.

З огляду на перелічені вище якості та наявність попереднього досвіду розробки, було прийняте рішення використовувати full-stack фреймворк Yii.

Кожний веб-додаток використовує одну чи декілька баз даних для зберігання інформації. Вони надають інструменти для організації, додавання, пошуку, оновлення, видалення й виконання обчислень над даними. Створення та управління базами даних здійснюється через системи управління базами даних (СУБД). Для інформаційної веб-системи, розроблюваної в рамках даної роботи, було вирішено використовувати СУБД MySQL. Це одна з найпопулярніших систем управління реляційними базами даних, що має відкритий код, є надійною, продуктивною, гнучкою, функціональною та має високий рівень безпеки. Для взаємодії з MySQL-сервером існують утиліти, деякі з яких навіть мають графічний інтерфейс. Більшість розробників віддають перевагу phpMyAdmin, яка працює в браузері. Вона має зручний та інтуїтивно зрозумілий інтерфейс й інтегрована у веб-сервер XAMPP. Серед основних функцій phpMyAdmin:

- перегляд і видалення баз даних, таблиць, представлень, полів, індексів;
- створення, копіювання, видалення, перейменування, оновлення баз даних, таблиць, полів, індексів;
- виконання, редагування, збереження SQL-запитів (при цьому дуже зручно користуватися підказками);
- управління сервером, базами даних, таблицями, ключами, а також користувачами та їх правами;
- робота з процедурами, тригерами;
- імпорт даних з CSV, SQL та експорт даних у різних форматах.

Крім цього, phpMyAdmin надає підказки, що значно прискорює та спрощує роботу з базою даних. [79,80]



## 3 ПРОЕКТУВАННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ

### 3.1 Проектування структури сайту

Веб-додатки, реалізовані за допомогою фреймворку Yii, організовані відповідно до архітектурного шаблону MVC. Model являє собою дані, бізнес логіку, бізнес правила. View відповідає за відображення інформації, в тому числі на основі даних, отриманих з Model. Controller приймає вхідні дані від користувача та перетворює їх у зрозумілий для Model формат і команди, відповідає за відображення необхідного View.

Крім компонентів MVC, Yii також має наступні сутності:

- вхідні скрипти – це php-скрипти, які доступні безпосередньо кінцевому користувачу веб-додатка;
- додатки – це глобально доступні об'єкти, які здійснюють коректне функціонування компонентів додатку, їх координацію для обробки запиту;
- компоненти додатку – це об'єкти, зареєстровані в додатку, що надають різноманітні можливості для обробки поточного запиту;
- модулі – це самостійні пакети, що включають повністю всі засоби для MVC. Веб-додаток може бути організований за допомогою декількох модулів;
- фільтри – це код, що повинен бути виконаний до й після обробки запиту Controller'ом;
- віджети – це об'єкти, що можуть включати View. Вони можуть містити логіку й бути використані в різноманітних View.

Структурна схема Yii веб-додатку наведена на Рисунку 3.1 – Структурна схема Yii веб-додатку. [82,83]

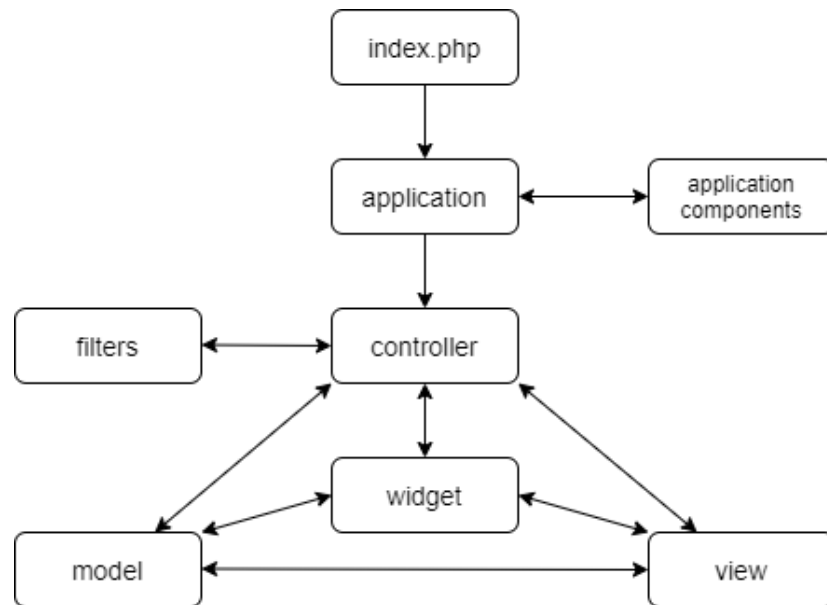


Рисунок 3.1 – Структурна схема Yii веб-додатку

Існує дві варіації фреймворку Yii – advanced та basic. В рамках даної роботи був використаний шаблон basic, структура папок якого наведена на Рисунок 3.2 – Структура папок шаблону basic.

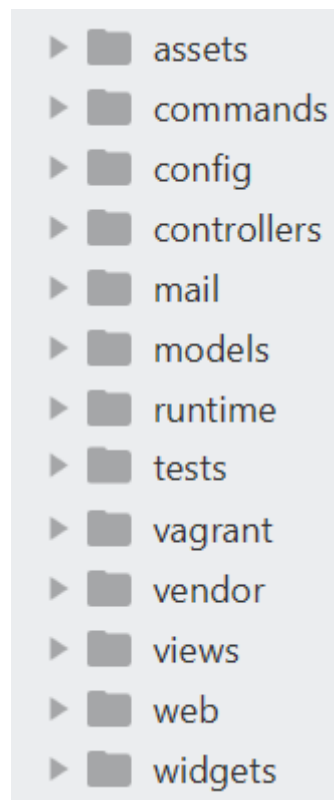


Рисунок 3.2 — Структура папок шаблону basic

Папка assets містить усі файли .css та .js, на які є посилання на веб-сторінці.

Папка commands містить консольні команди.

Папка config містить конфігураційні файли: console.php – конфігурація консольного додатку; db.php – конфігурація з'єднання з базою даних; params.php – додаткові параметри; web.php – конфігурація веб-додатку.

Папка controllers містить класи Controller. SiteController.php є головним Controller'ом. В рамках даної роботи були також створені TestsController.php, UsersController.php та QuestionsController.php.

Папка mail містить файли представлень електронної пошти.

Папка models містить класи Model. В рамках даної роботи до існуючих були додані: Gradebook.php, Results.php, Tests.php, TestsSearch.php, Users.php, UsersSearch.php та Questions.php.

Папка runtime містить файли, створені під час роботи сайту (логи, кеш).

Папка tests містить тести для основних додатків (приймальні, юніт, функціональні).

Папка vagrant містить файли-налаштування, необхідні для роботи з віртуальною машиною.

Папка vendor містить фреймворк Yii та сторонні бібліотеки, якими керує Composer.

Папка views містить файли View. Стандартно має підпапки layouts (містить шаблони) та site (містить окремі файли сторінок). В рамках даної роботи були додані підпапки tests, users, questions, що містять відповідні сторінки з адміністрування питань, користувачів і безпосередньо питання.

Папка web є основною папкою сайту, містить файли стилів, скрипти, зображення. Саме в папці web знаходиться файл index.php, що є точкою

входу. В рамках даної роботи до папки були додані зображення, що використовуються в питаннях.

Папка widgets містить віджети – компоненти, які призначені для реалізації елементів інтерфейсу й можуть бути неодноразово використані. [81-83]

### 3.2 Проектування інформаційної системи

Проектована інформаційна система призначена для використання в середніх навчальних закладах учителями й учнями. За допомогою даної системи вчителі можуть надавати доступ до виконання практичних робіт учням, а учні – виконувати їх і переглядати свої результати, які також доступні й для вчителів. Користувачами інформаційної системи є також адміністратори, які уповноважені виконувати реєстрацію вчителів і учнів. Процес, що представляє роботу інформаційної системи, зовнішні сутності та потоки даних між ними відображає DF-діаграма, наведена на Рисунку 3.3 – DF-діаграма нульового рівня деталізації.



Рисунок 3.3 – DF-діаграма нульового рівня деталізації

Процеси, сховища та потоки даних всередині системи відображає DF-діаграма, наведена на Рисунку 3.4 – DF-діаграма першого рівня деталізації.

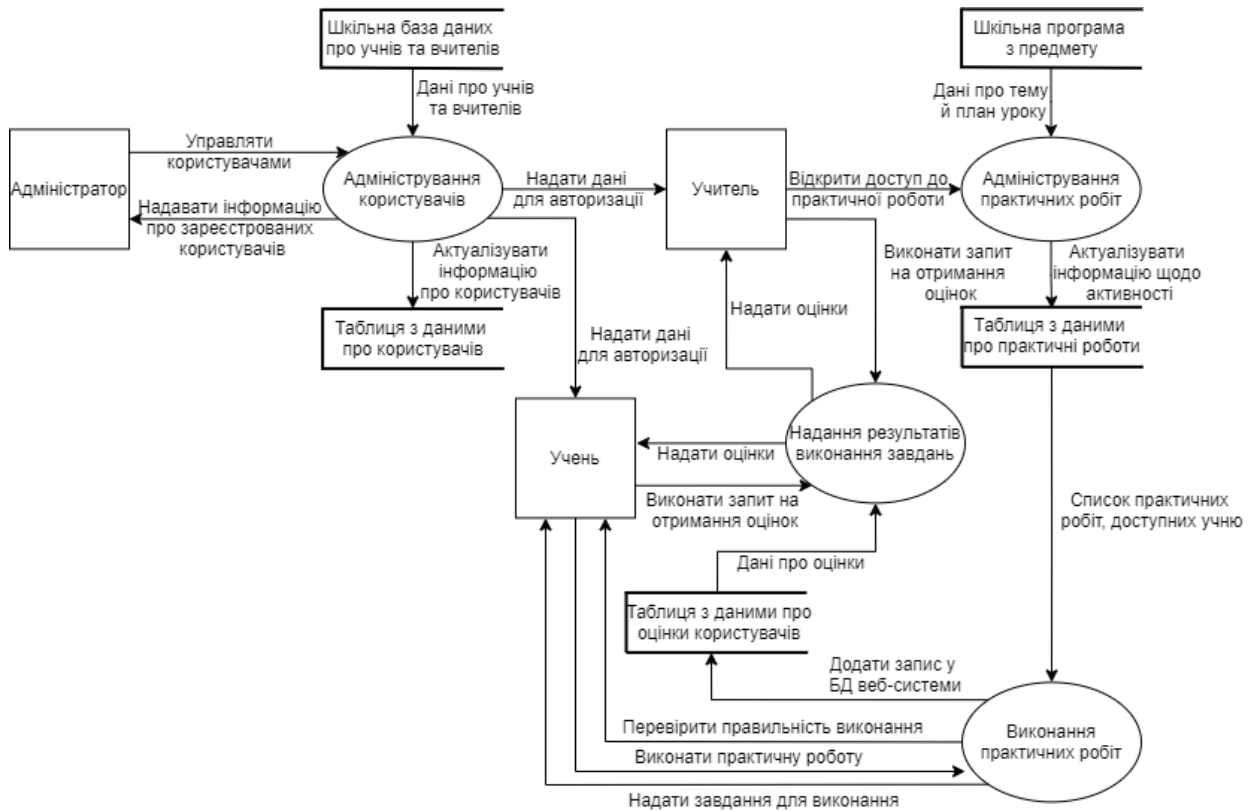


Рисунок 3.4 – DF-діаграма першого рівня деталізації

Інформаційна веб-система повинна забезпечувати контроль доступу користувачів до виконання операцій на основі їх ролей. Передбачено такі ролі: гість (користувач, який ще не пройшов авторизацію), адміністратор, учитель, учень. Функції, доступні кожній з ролей користувачів, ілюструє діаграма варіантів використання, наведена на Рисунку 3.5 – Діаграма варіантів використання.



Рисунок 3.5 – Діаграма варіантів використання

На етапі проектування бази даних для інформаційної веб-системи були виділені сутності: Користувач, Тест, Результат. Дані про них зберігаються в таблицях «users», «tests» і «results» відповідно.

Таблиця «users» зберігає дані про користувачів інформаційної веб-системи. Користувач має унікальний ідентифікатор і характеристики: ім'я користувача в системі, пароль, ім'я, прізвище, клас (для учнів), ідентифікатор вчителя (чи є користувач учителем), ідентифікатор адміністратора (чи є користувач адміністратором).

Таблиця «tests» зберігає дані про завдання, що наявні в інформаційній веб-системі. Тест має унікальний ідентифікатор і характеристики: назва, зображення (використовується на сторінці зі списком тестів), клас (для якого призначений тест), ідентифікатор активності (учні мають доступ тільки до активних тестів, учителі та адміністратори – до усіх).

Таблиця «results» зберігає дані про результати виконання завдань учнем. Результат визначається комбінацією ідентифікатора учня й тесту (така комбінація не обов'язково є унікальною) та характеризується оцінкою в 12-бальній системі й датою та часом завершення проходження тесту.

Вигляд таблиць «users», «tests» і «results» бази даних «biology», а також відношення між ними наведені на Рисунку 3.6 – ER-діаграма бази даних «biology».

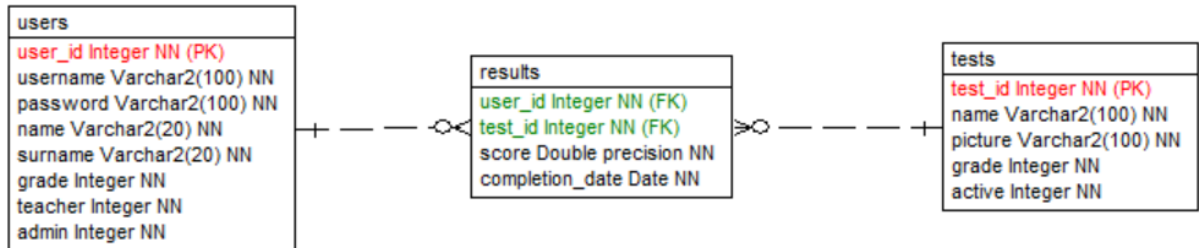


Рисунок 3.6 – ER-діаграма бази даних «biology»

### 3.3 Реалізації основних компонентів

Набір вкладок, що відображаються для користувача, відрізняється в залежності від його ролі. Так, для користувача, що не пройшов авторизацію (гостя), доступні лише вкладки «Увійти» й «Головна». Головна сторінка інформаційної веб-системи (вкладка «Головна») зображена на Рисунку 3.7 – Головна сторінка.

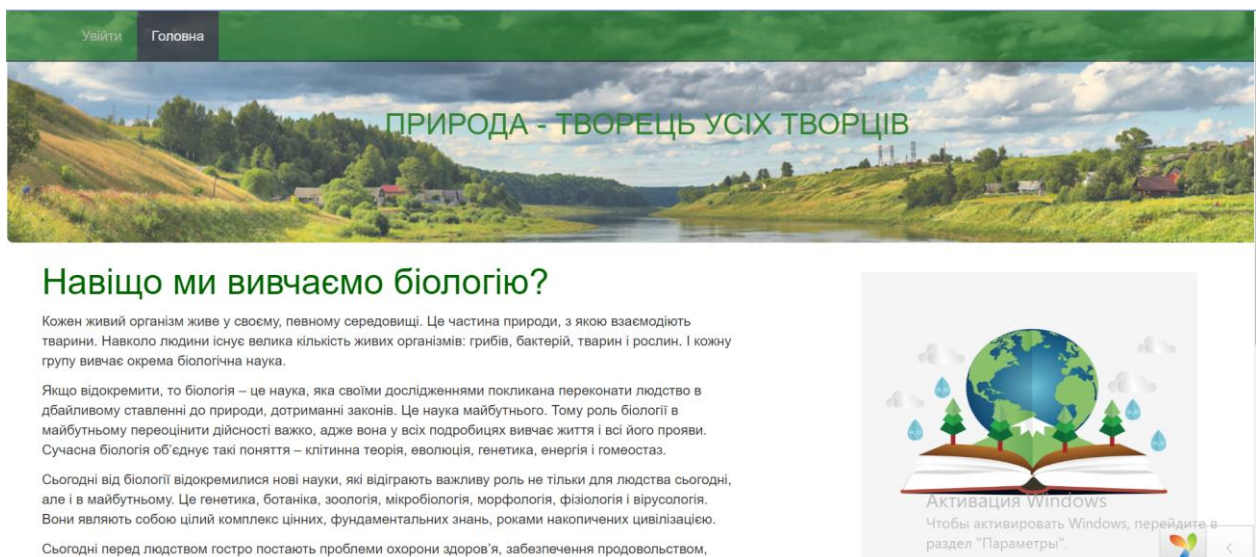


Рисунок 3.7 – Головна сторінка

Регулювання доступу до вкладок за ролями користувачів реалізоване за

допомогою класу `yii\web\User`, який дозволяє ідентифікувати роль аутентифікованого користувача й на основі цієї інформації надати йому доступ до певних вкладок. Відповідні налаштування містить файл `views/layouts/main.php`:

```
Yii::$app->user->isGuest ? (
['label' => 'Увійти', 'url' => ['/site/login']]
) : (
'<li>'
. Html::beginForm(['/site/logout'], 'post')
. Html::submitButton(
'Вийти (' . Yii::$app->user->identity->username . ')',
['class' => 'btn btn-link logout'])
. Html::endForm()
. '</li>'),
['label' => 'Головна', 'url' => ['/site/index']],
Yii::$app->user->isGuest ? (
['label' => 'Головна', 'url' => ['/site/index']]
) : (Yii::$app->user->identity->teacher == 1 ? (
['label' => 'Адміністрування', 'url' => ['/tests/index']]
) : (Yii::$app->user->identity->admin == 1 ? (
['label' => 'Адміністрування', 'url' => ['/users/index']]
) : ([ 'label' => 'Мої результати', 'url' =>
['/site/results']])),
['label' => 'Тести', 'url' => ['/site/tests']],
Yii::$app->user->isGuest ? (
['label' => ' ', 'url' => ['/users/index']]
) : (Yii::$app->user->identity->teacher == 1 ? (
['label' => 'Журнал', 'url' => ['/site/gradebook']]
) : ([ 'label' => ' ', 'url' => ['/tests/index']])),
```

Для роботи з інформаційною веб-системою користувачеві не потрібно реєструватися. Дані про учнів та вчителів заносить у базу даних адміністратор, використовуючи адмін-панель. Логін та пароль, необхідні для



авторизації, користувачі також отримують від нього. Увійти в систему можна перейшовши на вкладку «Увійти», вигляд якої наведений на Рисунку 3.8 – Форма авторизації користувача.

Рисунок 3.8 – Форма авторизації користувача

Дані, якими користувач заповнює поля форми, проходять валідацію на рівні моделі. Клас *LoginForm* містить дві функції: *rules()*, які підтримує валідацію на стороні клієнта й сервера, та *validatePassword()*, яка підтримує валідацію на стороні сервера. Функція *rules()* перевіряє, що обов'язкові поля заповнені без звернення до сервера, і у випадку наявності пустих, відображає повідомлення про помилку. Функція *validatePassword()* перевіряє, що в базі даних існує користувач із заданими логіном та паролем. Обидві функції містить файл *models/LoginForm.php*:

```
public function rules(){
    return [
        ['username', 'required', 'message' => 'Будь ласка, введіть логін'],
        ['password', 'required', 'message' => 'Будь ласка, введіть пароль'],
    ],
}
```

```

        ['password', 'validatePassword'],
    ];
}

public function validatePassword($attribute, $params){
    if (!$this->hasErrors()) {
        $user = $this->getUser();
        if (!$user || !$user->validatePassword($this->password)) {
            $this->addError($attribute, 'Неправильні вхідні дані.');
```

Власне форма генерується за допомогою віджета *ActiveForm*. Код створення форми містить файл *views/site/login.php*:

```

<?php $form = ActiveForm::begin([
    'id' => 'login-form',
    'layout' => 'horizontal',
    'fieldConfig' => [
        'template' => "{label}\n<div class=\"col-sm-8\">{input}</div>\n<div class=\"col-sm-8\">{error}</div>",
        'labelOptions' => ['class' => 'col-sm-4 control-label'],],],]); ?>

<div class="labels">
<?=$form->field($model, 'username')->textInput(['autofocus' => false]) ?>
<?=$form->field($model, 'password')->passwordInput() ?></div>
<div class="row">
<div class="col-lg-3" align="center">
<?=$form->submitButton('Увійти', ['class' => 'btn btn-success', 'name' => 'login-button']) ?></div>
<div class="col-lg-3" align="left"></div></div>
```

```
<?php ActiveForm::end(); ?>
```

Для авторизованого користувача відображається вкладка «Тести», вміст якої залежить від його ролі – вчитель та адміністратор можуть бачити всі практичні роботи, наявні в інформаційній веб-системі, учні – лише ті практичні роботи, що призначені для їх класу. Практична робота відображається у вигляді блоку, що містить її назву, тематичне зображення, попередній результат у відсотках та кнопку «Розпочати», яка є активною лише у випадку, якщо доступ до практичної роботи відкрито вчителем. Вигляд вкладки «Тести» для користувача-учня наведений на Рисунку 3.9 – Вкладка «Тести» для учня.

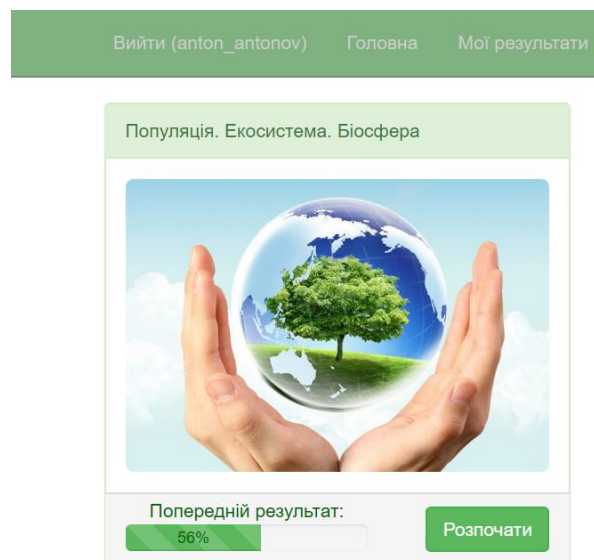


Рисунок 3.9 – Вкладка «Тести» для учня.

Для отримання з бази попереднього результату виконання практичної роботи використаний конструктор запитів *yii\db\Query*, який дозволяє створювати SQL-вирази в не залежному від СУБД вигляді, робить їх більш безпечними та простими для сприйняття. Після створення об'єкту *yii\db\Query* з'являється можливість виклику різних методів, кожний із яких повертає свій об'єкт та має назву, схожу на назву ключового слова SQL.

Об'єднання викликів цих методів утворює SQL-запит. Наприклад, в даному випадку були використані такі методи: *select()* (дозволяє вказати стовпці таблиці, які потрібно вибрати з бази даних), *from()* (дозволяє вказати назви таблиць, до яких необхідно зробити запит), *innerJoin()* (дозволяє вказати назви таблиць, які необхідно об'єднати, умову об'єднання), *where()* (дозволяє задавати умови вибору), *orderBy()* (дозволяє задавати сортування за стовпцями та його вид), *one()* (повертає перший рядок результатів запиту). [81] Запит на фільтрацію практичних робіт для певного учня в залежності від його класу (для вчителя та адміністратора такої фільтрації немає) та отримання попереднього результату виконання практичної роботи (якщо такий є, інакше – попередній результат дорівнює 0%):

```
<?php foreach ($tests as $test):
if (Html::encode("{ $test->grade}") == Yii::$app->user->identity->grade || Yii::$app->user->identity->admin == 1 || Yii::$app->user->identity->teacher == 1) {
$scores = (new \yii\db\Query())
    ->select ('score')
    ->from('results')
    ->innerJoin('users', 'results.pupil_id = users.id')
    ->innerJoin('tests', 'results.test_id = tests.id')
    ->where([
        'users.id' => Yii::$app->user->identity->id,
        'tests.id' => $test->id, ])
    ->orderBy('results.completion_date DESC')
    ->one(); } endforeach;?>
```

Для початку виконання практичної роботи необхідно натиснути кнопку «Розпочати», яка направляє користувача на сторінку з першим завданням, в рамках якого потрібно за допомогою миші перетягнути зображення у відповідні блоки. Вигляд завдання наведений на Рисунку 3.10 – Перше завдання практичної роботи.

Вийти (anton\_antonov) Головна Мої результати Тести

## Практична робота: Тема 1 Популяція. Екосистема. Біосфера

1. Залежно від природи та особливостей дії екологічні фактори ділять на абіотичні, біотичні та антропогенні. Розподіліть наведені екологічні фактори за групами, до яких вони належать.

The screenshot shows a web interface for a practical assignment. At the top, there is a navigation bar with links for 'Вийти (anton\_antonov)', 'Головна', 'Мої результати', and 'Тести'. The main heading is 'Практична робота: Тема 1 Популяція. Екосистема. Біосфера'. Below this is a task instruction: '1. Залежно від природи та особливостей дії екологічні фактори ділять на абіотичні, біотичні та антропогенні. Розподіліть наведені екологічні фактори за групами, до яких вони належать.' The interface contains three empty rectangular boxes with green headers: 'АБІОТИЧНІ', 'БІОТИЧНІ', and 'АНТРОПОГЕННІ'. Below these boxes is a row of icons representing different ecological factors: a globe, a tree, a water drop, a satellite, a rocket, a thermometer, a cat, and a tree in a pot. A 'Ставки' (Bets) button is located above the tree-in-pot icon. At the bottom right, there are two buttons: 'Дати відповідь' (Give answer) and 'Далі' (Next).

Рисунок 3.10 – Перше завдання практичної роботи

На сторінці із завданням є кнопки «Дати відповідь» і «Далі». Після натискання кнопки «Дати відповідь» виконується перевірка правильності виконання завдання, результат якої відразу візуалізується для користувача. Крім цього, стає активною кнопка переходу до наступного завдання – «Далі».

Обробка рухів мишею відбувається засобами мови JavaScript, а саме використовується інтерфейс *DataTransfer*. Об'єкт *DataTransfer* зберігає дані, що перетягуються під час виконання операцій *drag&drop*. Для елементів, які потрібно перетягувати (в даному випадку зображень), був встановлений атрибут *draggable="true"*, а також доданий обробник події *dragStart()* (початок виконання операції перетягування користувачем). Для визначення цільових елементів (в даному випадку блоків) були додані обробники подій *dragEnter()* (досягнення елементом, що перетягується, цільового елемента), *dragDrop()* (закінчення перетягування елемента), *dragOver()* (наведення курсору миші на елемент при перетягуванні).

Перевірка правильності розміщення зображень відбувається шляхом визначення їх положення на сторінці та подальшого порівняння з координатами цільових блоків. Для доступу до елементів використовується метод *document.querySelector()*, їх координати визначаються за допомогою методу *getBoundingClientRect()*. Після цього виконується перевірка, що зображення знаходиться в межах певного блоку. Відповідно до правильності

розміщення зображення змінюється його стиль – додається зелена або червона рамка. Повний код вищеописаної реалізації наведений у Додатку Б.

У другому завданні пропонується встановити відповідність. Користувач має поставити прапорець біля блоку з лівого стовпця й відповідного блоку з правого стовпця, після чого ці блоки поєднуються лінією. Встановлену відповідність можна скасувати натисканням кнопки «Скасувати», але лише до того, як була натиснута кнопка «Дати відповідь» – після її натискання кнопка «Скасувати» стає неактивною. В рамках перевірки правильності виконання завдання лінії залишаються зеленими або змінюють колір на червоний. Вигляд завдання наведений на Рисунку 3.11 – Друге завдання практичної роботи.

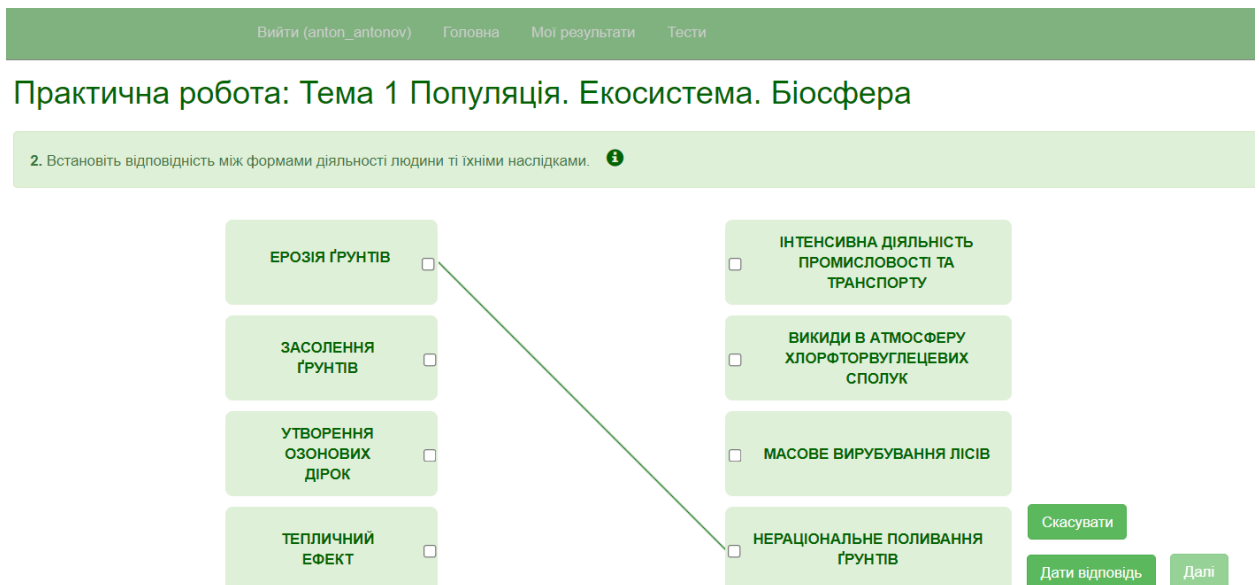


Рисунок 3.11 – Друге завдання практичної роботи

Для утворення ліній між стовпцями розташований HTML-елемент *canvas*, що використовується для малювання графіки засобами JavaScript. В даному випадку були використані методи: *beginPath()* (починає нову лінію), *moveTo()* (визначає початкову точку лінії), *lineTo()* (визначає кінцеву точку лінії), *stroke()* (відображає побудовану лінію). Початок і кінець лінії визначаються як початок і кінець елемента *canvas* (по осі *x*) та як середини відповідних блоків, координати яких отримуються за допомогою методу

`getBoundingClientRect()` (по осі *y*). Повний код вищеописаної реалізації наведений у Додатку В.





Третє завдання полягає у зафарбовуванні сірих зображень шляхом натискання кнопки відповідного кольору. Крім безпосередньо постановки завдання на сторінку, додана спливаюча підказка, які саме дії має виконати користувач. Вигляд завдання наведений на Рисунку 3.12 – Третє завдання практичної роботи.





Вийти (anton\_antonov) Головна Мої результати Тести





### Практична робота: Тема 1 Популяція. Екосистема. Біосфера

3. Встановіть, про який документ ідеться: Червона книга, Чорна книга або Зелена книга. ?

Зверніть увагу! Для того, щоб відповісти на питання, необхідно пофарбувати знак книги у відповідний колір.

- 




державний документ про сучасний стан видів тварин і рослин, які перебувають під загрозою зникнення, та про заходи щодо їхнього збереження та науково обґрунтованого відтворення.
- 




державний документ про рідкісні і типові для певної для певної місцевості рослинні угруповання, що потребують встановлення особливого режиму їх використання.
- 




державний документ про види, які зникли з лиця Землі починаючи з 1600 року. Підставою для введення виду до книги є відсутність достовірних його знахідок протягом останніх 50 років.

Дати відповідь Далі

Рисунок 3.12 – Третє завдання практичної роботи

Завдання реалізоване засобами мови JavaScript. При натисканні кнопки відповідного кольору за допомогою атрибута подій *onclick* викликається функція, яка знаходить потрібний елемент з використанням метода `document.getElementById()` та змінює його стиль (колір). Встановлені кольори зберігаються в змінних, а при натисканні кнопки «Дати відповідь» порівнюються з еталонними значеннями. Аналогічним способом напроти кожного зображення-книги для користувача відображаються прапорці зеленого або червоного кольору (спочатку вони є невидимими), які свідчать про правильну або неправильну відповідь. Повний код вищеописаної реалізації наведений у Додатку Г.

У четвертому питанні користувач має розташувати зображення у відповідні комірки для них шляхом перетягування. Пояснення значень зображень додане у вигляді спливаючої підказки, яка з'являється при наведенні мишею на зображення. Розміщення зображень можна повернути до початкового вигляду шляхом натискання кнопки «Скасувати», але лише у випадку, якщо не була натиснута кнопка «Дати відповідь», яка робить неактивною кнопку «Скасувати». Вигляд завдання наведений на Рисунку 3.13 – Четверте завдання практичної роботи.



Рисунок 3.13 – Четверте завдання практичної роботи

Аналогічно першому завданню обробка рухів мишею та перевірка правильності розміщення зображень виконуються засобами мови JavaScript з використанням інтерфейсу *DataTransfer* та обробників події *dragStart()*, *dragEnter()*, *dragDrop()*, *dragOver()*. Для доступу до елементів використовується метод *document.querySelector()*, для визначення їх координат – *getBoundingClientRect()*. Повний код вищеописаної реалізації наведений у Додатку Д.

У рамках п'ятого завдання користувач має розв'язати кросворд, завдання до нього подані у вигляді спливаючих панелей. Після натискання кнопки «Дати відповідь» виконується перевірка правильності відповідей –



букви залишаються зеленими або змінюють колір на червоний. Вигляд завдання наведений на Рисунку 3.14 – П'яте завдання практичної роботи.

Рисунок 3.14 – П'яте завдання практичної роботи

Сітка кросворда створена засобами HTML-розмітки. Для доступу до комірок з метою перевірки правильності заповнення та зміни кольору використовується JavaScript-метод `document.getElementById()`. Повний код вищеописаної реалізації наведений у Додатку Е.

Для користувача-учня передбачена спеціальна вкладка «Мої результати», яка містить всі результати виконання практичних робіт учнем у вигляді таблиці. Результати фільтруються за поточним класом учня, тому після його переходу до наступного класу минулорічні оцінки відображені не будуть. Таблиця надає такі дані: назва практичної роботи, результат у відсотках та 12-бальній системі, дата проходження. Передбачена можливість сортування кожного стовпця в порядку зростання й спадання, а також пошук результатів за назвою практичної роботи. Вкладка «Мої результати» наведена на Рисунку 3.15 – Оцінки учня на вкладці «Мої результати».

Назва практичної роботи	Результат, %	Результат, балів	Дата проходження
Популяція. Екосистема. Біосфера	67	8	0000-00-00 00:00:00
Популяція. Екосистема. Біосфера	56	7	2020-04-04 00:00:00
Популяція. Екосистема. Біосфера	36	4	2020-04-04 00:00:00
Популяція. Екосистема. Біосфера	93	11	2020-04-04 00:00:00

Рисунок 3.15 – Оцінки учня на вкладці «Мої результати»

Для отримання результатів учня з бази даних використаний метод `yii\db\ActiveRecord::findByPrimaryKey()`, який дозволяє створювати запити на «чистому» SQL, що є зручним у даному випадку:

```
$sql_name = 'SELECT name, score, completion_date from tests join
results on id = test_id where pupil_id = :id';
$res_name = Tests::findByPrimaryKey($sql_name, [':id' => Yii::$app-
>user->identity->id])->all();
$sql_score = 'SELECT name, score, completion_date from tests
join results on id = test_id where pupil_id = :id';
$res_score = Results::findByPrimaryKey($sql_score, [':id' => Yii::$app-
>user->identity->id])->all();
```

Для забезпечення сортування даних була створена функція `sortTable()` мовою JavaScript, яка викликається при натисканні на назву стовпця таблиці. Для забезпечення пошуку результатів за назвою практичної роботи створена функція `search()` мовою JavaScript. Код цих функцій наведений у Додатку А.

Для відображення рядків таблиці з результатами, а також визначення їх кольору в залежності від рівня оцінки використовується цикл `foreach` мовою PHP та відповідні HTML-теги:

```
<tbody> <?php $j = 0;
foreach ($res_name as $name):
    if (round($scores[$j] * 12 / 100) >= 10) { ?>
```

```

<tr class="success">
    <td><?php echo $name->name; ?></td>
    <td><?php echo $scores[$j]; ?></td>
    <td><?php echo round($scores[$j] * 12 / 100);
    ?></td>
    <td><?php echo $dates[$j]; ?></td> </tr> <?php }
elseif (round($scores[$j] * 12 / 100) < 10 &&
round($scores[$j] * 12 / 100) >= 6) { ?>
    <tr class="warning">
        <td><?php echo $name->name; ?></td>
        <td><?php echo $scores[$j]; ?></td>
        <td><?php echo round($scores[$j] * 12 / 100);
        ?></td>
        <td><?php echo $dates[$j]; ?></td></tr> <?php }
else{ ?>
    <tr class="danger">
        <td><?php echo $name->name; ?></td>
        <td><?php echo $scores[$j]; ?></td>
        <td><?php echo round($scores[$j] * 12 / 100);
        ?></td>
        <td><?php echo $dates[$j];?></td></tr><?php }
    $j = $j + 1;
endforeach; ?></tbody>

```

Для користувача-адміністратора доступна вкладка «Адміністрування» користувачів. Оскільки реєстрація в інформаційній веб-системі не передбачена, користувач-адміністратор є попередньо створеним. Саме він додає всіх інших користувачів (учителів та учнів), визначаючи їх роль. Такий підхід, по-перше, гарантує наявність у базі коректних даних про користувачів (наприклад, виключає можливість використання учнями вигаданих імен при реєстрації), по-друге, економить час уроку (не потрібно витратити час на реєстрацію, можна відразу перейти безпосередньо до використання інформаційної веб-системи). Адміністратор має можливість редагувати

користувачів (наприклад, коли учні переходять до іншого класу) та видаляти їх (наприклад, коли учні випускаються зі школи або вчителі звільняються).

Адміністративна сторінка створена з використанням генератора коду Gii. Був згенерований клас моделі *Users* (*models/Users.php*) на основі таблиці «users» за допомогою Model Generator. Налаштування, що були виконані для цього, наведені на Рисунку 3.16 – Налаштування Model Generator для генерації коду класу моделі Users.

The screenshot shows the Gii Model Generator configuration page. The sidebar on the left lists various generators, with 'Model Generator' selected. The main content area is titled 'Model Generator' and includes a description: 'This generator generates an ActiveRecord class for the specified database table.' The configuration fields are as follows:

- Table Name:** users
- Model Class Name:** Users
- Standardize Capitals:**
- Singularize:**
- Namespace:** app/models
- Base Class:** yii\db\ActiveRecord
- Database Connection ID:** db
- Use Table Prefix:**
- Generate Relations:** All relations
- Generate Relations from Current Schema:**
- Generate Labels from DB Comments:**
- Generate ActiveQuery:**

On the right side, there is a note: 'Активац Чтобы акти раздел "Па'.'

Рисунок 3.16 – Налаштування Model Generator для генерації коду класу моделі Users

Отриманий клас моделі *Users* був відредагований відповідно до потреб інформаційної веб-системи. Функція *rules()*, яка задає правила для перевірки даних, введених користувачем, набула вигляду:

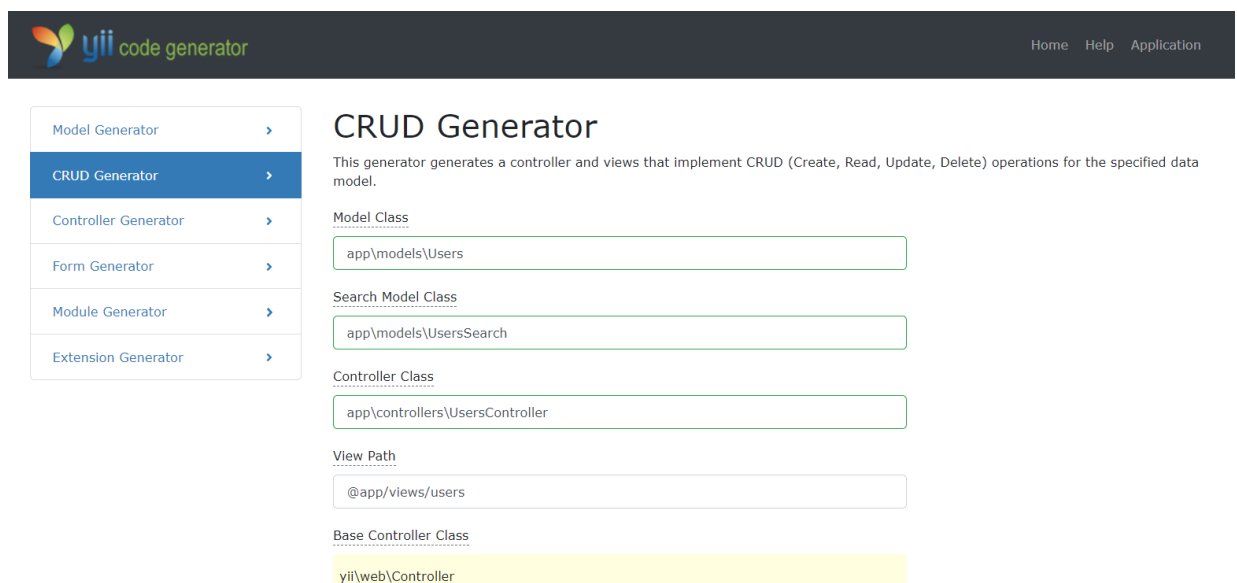
```
public function rules() {
    return [
        ['name', 'required', 'message' => 'Будь ласка, введіть ім\'я!'],
        ['surname', 'required', 'message' => 'Будь ласка, введіть прізвище!'],
    ];
}
```

```

['username', 'required', 'message' => 'Будь ласка, введіть логін!'],
['password', 'required', 'message' => 'Будь ласка, введіть пароль!'],
['teacher', 'boolean', 'message' => 'Будь ласка, введіть 0 або 1!'],
['teacher', 'required', 'message' => 'Будь ласка, визначте роль користувача!'],
['admin', 'boolean', 'message' => 'Будь ласка, введіть 0 або 1!'],
['admin', 'required', 'message' => 'Будь ласка, визначте роль користувача!'],
[['username', 'name', 'surname'], 'string', 'max' => 25],
['grade', 'in', 'range' => [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11],
'message' => 'Будь ласка, введіть число від 1 до 11!'],];}

```

Для створення функціоналу CRUD був використаний CRUD Generator. Налаштування, що були виконані для цього, наведені на Рисунку 3.17 – Налаштування CRUD Generator для створення функціоналу CRUD за класом моделі Users.



The screenshot shows the 'yii code generator' interface. On the left is a sidebar with navigation links: Model Generator, **CRUD Generator**, Controller Generator, Form Generator, Module Generator, and Extension Generator. The main area is titled 'CRUD Generator' and contains the following configuration fields:

- Model Class:**
- Search Model Class:**
- Controller Class:**
- View Path:**
- Base Controller Class:**

Рисунок 3.17 – Налаштування CRUD Generator для створення функціоналу CRUD за класом моделі Users

Отриманий код був відредагований відповідно до потреб інформаційної веб-системи. Функція *behaviors()* класу *UsersController* (*controllers/UsersController.php*), яка задає правила доступу ролей користувачів до певних дій, набула вигляду:

```
public function behaviors(){
return [
'access' => [
'class' => AccessControl::className(),
'only' => ['index', 'view', 'create', 'update', 'delete'],
'rules' => [
['actions' => ['index', 'view', 'create', 'update', 'delete'],
'allow' => true,
'roles' => ['@'],
'matchCallback' => function($rule, $action){ return Yii::$app->user->identity->admin === 1; }],],],
'verbs' => [
'class' => VerbFilter::className(),
'actions' => [
'delete' => ['POST'],],],];
```

У результаті була отримана адміністративна сторінка, яка дозволяє виконувати операції створення, перегляду, редагування й видалення користувачів. Вкладка «Адміністрування» містить таблицю «Користувачі», яка відповідає таблиці «users» в базі даних. Кожен стовпець таблиці має поле для введення значення, за яким буде виконаний пошук, а також може бути відсортований у порядку зростання й спадання. Вигляд сторінки наведений на Рисунку 3.18 – Адміністративна сторінка для адміністратора.









#	ID	Логін	Пароль	Ім'я	Прізвище	Клас	Учитель	Адміністратор	
1	1	anton_antonov	12	Антон	Антонов	11	0	0	 
2	2	artem_bolgov	12	Артем	Болгов	11	0	0	 
3	3	denys_vasyliiev	12	Денис	Васильєв	11	0	0	 
4	4	natalia_vasyliieva	12	Наталія	Васильєва	11	0	0	 

Рисунок 3.18 – Адміністративна сторінка для адміністратора

Сторінка створення користувача відкривається при натисканні кнопки «Створити користувача». Її вигляд наведений на Рисунку 3.19 – Сторінка «Створити користувача».

Вийти (admin)	Головна	Адміністрування	Тести
<h2>Створити користувача</h2>			
<b>Логін</b>			
<input type="text" value="Введіть логін користувача"/>			
<b>Пароль</b>			
<input type="text" value="Введіть пароль користувача"/>			
<b>Ім'я</b>			
<input type="text" value="Введіть ім'я користувача"/>			
<b>Прізвище</b>			
<input type="text" value="Введіть прізвище користувача"/>			
<b>Клас</b>			
<input type="text" value="Введіть клас, у якому навчається користувач (для учнів)"/>			
<b>Учитель</b>			
<input type="text" value="Введіть 1, якщо користувач є вчителем"/>			
<b>Адміністратор</b>			
<input type="text" value="Введіть 1, якщо користувач є адміністратором"/>			
<input type="button" value="Зберегти"/>			

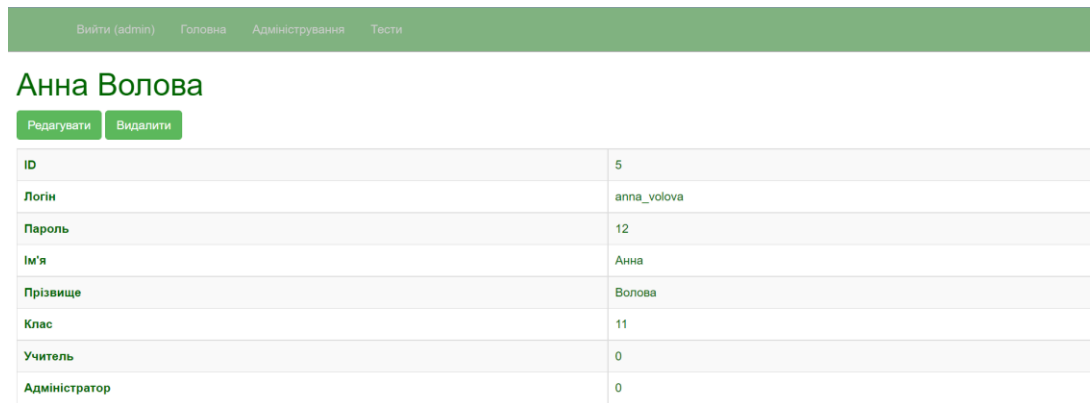
Рисунок 3.19 – Сторінка «Створити користувача»

Форма створення користувача генерується за допомогою класу `yii\widgets\ActiveForm`. Відповідний код міститься у файлі `views/users/_form.php`:

```
<?php $form = ActiveForm::begin(); ?>
<div class="col-xs-5">
<?= $form->field($model, 'username')->textInput(['maxlength' =>
true, 'placeholder' => 'Введіть логін користувача']) ?>
<?= $form->field($model, 'password')->textInput(['maxlength' =>
true, 'placeholder' => 'Введіть пароль користувача']) ?>
<?= $form->field($model, 'name')->textInput(['maxlength' =>
true, 'placeholder' => 'Введіть ім'я користувача']) ?>
<?= $form->field($model, 'surname')->textInput(['maxlength' =>
true, 'placeholder' => 'Введіть прізвище користувача']) ?>
<?= $form->field($model, 'grade')->textInput(['maxlength' =>
true, 'placeholder' => 'Введіть клас, у якому навчається
користувач (для учнів)']) ?>
<?= $form->field($model, 'teacher')->textInput(['maxlength' =>
true, 'placeholder' => 'Введіть 1, якщо користувач є
вчителем']) ?>
<?= $form->field($model, 'admin')->textInput(['maxlength' =>
true, 'placeholder' => 'Введіть 1, якщо користувач є
адміністратором']) ?>
<div class="form-group">
<?= Html::submitButton('Зберегти', ['class' => 'btn btn-
success']) ?></div></div>
<?php ActiveForm::end(); ?>
```

Для перегляду даних про користувача необхідно натиснути кнопку «Переглянути» (зображення у вигляді ока) на початковій сторінці адміністрування. Сторінка перегляду користувача наведена на Рисунку 3.20 – Сторінка перегляду даних про користувача.





Анна Волова	
Редагувати	Видалити
ІД	5
Логін	anna_volova
Пароль	12
Ім'я	Анна
Прізвище	Волова
Клас	11
Учитель	0
Адміністратор	0

Рисунок 3.20 – Сторінка перегляду даних про користувача

Сторінка сформована за допомогою віджета *DetailView*, який дає можливість відобразити дані про атрибути об'єкту у вигляді рядків таблиці. Налаштування списку атрибутів для відображення знаходиться у файлі *views/users/view.php*:

```
<?= DetailView::widget([
    'model' => $model,
    'attributes' => ['id', 'username', 'password', 'name',
        'surname', 'grade', 'teacher', 'admin', ], ])?>
```

Для редагування даних про користувача необхідно натиснути кнопку «Оновити» (зображення у вигляді олівця) на початковій сторінці адміністрування. Для створення сторінки редагування використовується форма *views/users/\_form.php*, тому її вигляд аналогічний сторінці створення користувача.

Для видалення запису про користувача з бази даних необхідно натиснути кнопку «Видалити» (зображення у вигляді кошика) на початковій сторінці адміністрування. Система вимагає підтвердження цієї операції – запит підтвердження наведений на Рисунку 3.21 – Запит підтвердження видалення користувача.

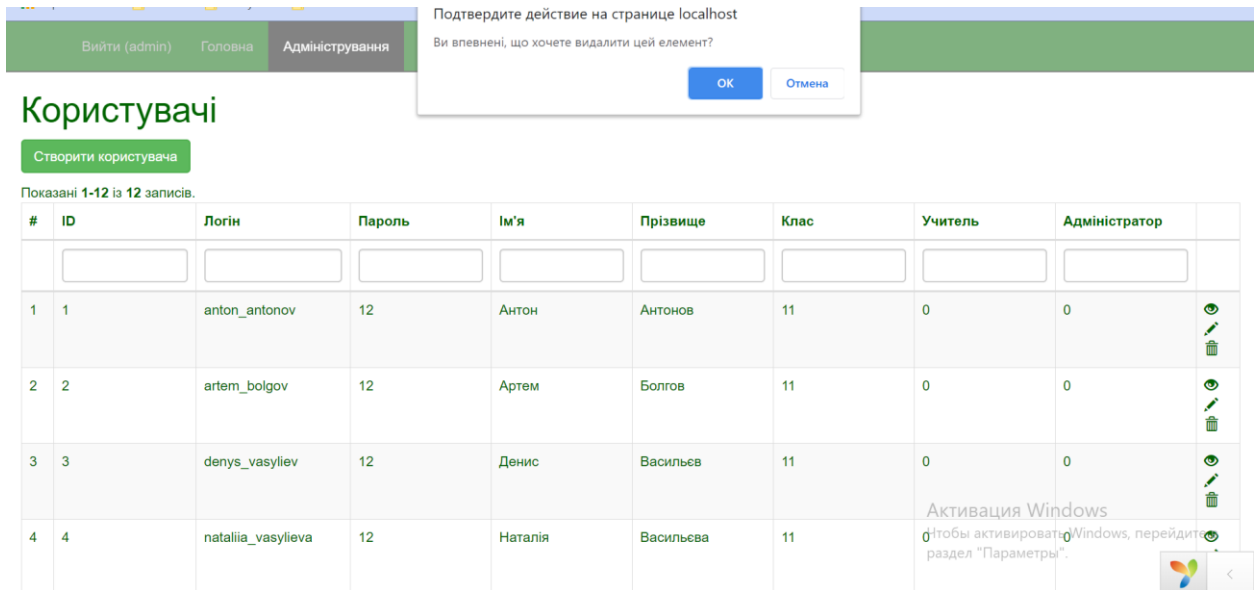


Рисунок 3.21 – Запит підтвердження видалення користувача

Для користувача-вчителя доступна вкладка «Адміністрування» практичних робіт. Вона створена аналогічно відповідній вкладці для адміністратора – клас моделі *models/Tests.php* та функціонал CRUD згенеровані за допомогою Gii, проте, надана можливість лише редагувати записи про практичні роботи – відкривати доступ до завдань для учнів. Вигляд сторінки наведений на Рисунку 3.22 – Адміністративна сторінка для вчителя.

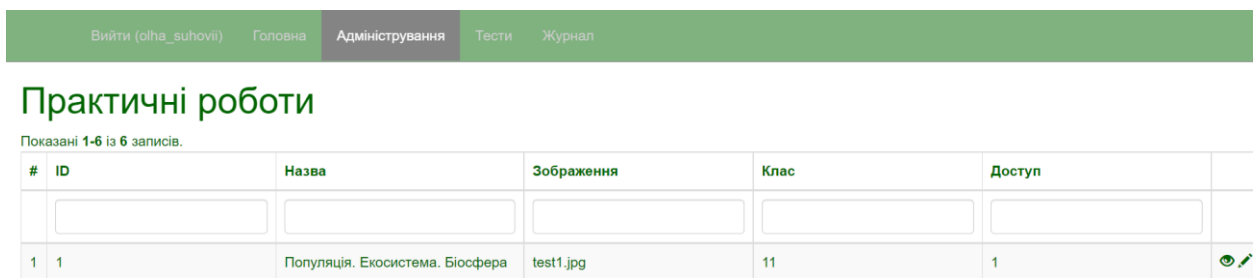
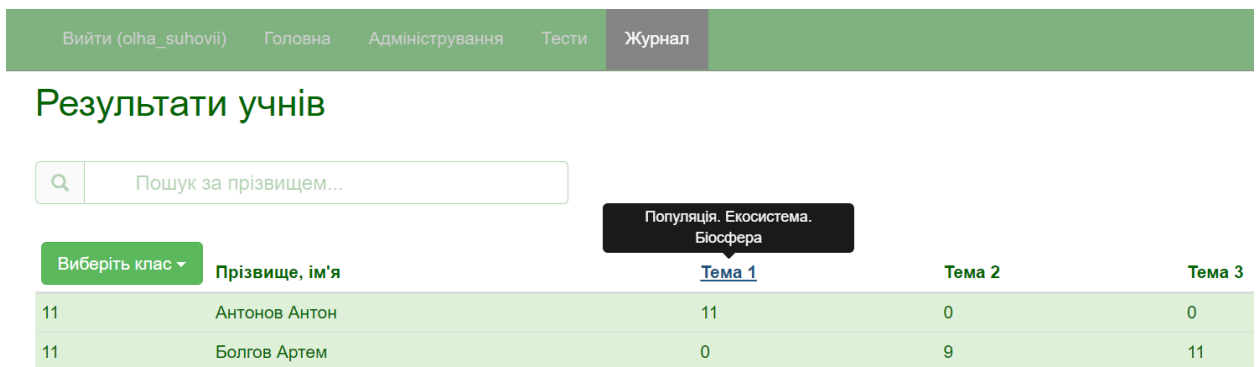


Рисунок 3.22 – Адміністративна сторінка для вчителя

Для користувача-вчителя також доступна спеціальна вкладка «Журнал», яка дає можливість переглянути результати виконання практичних робіт учнями певного класу, причому відображені їх найуспішніші спроби. При переході на цю вкладку необхідно обрати клас, після чого з'явиться таблиця

зі стовпцями «Прізвище, ім'я» та «Тема <номер>» (кількість таких стовпців відповідає кількості практичних робіт, що існують в системі для заданого класу). Записи таблиці можна сортувати та фільтрувати за стовпцем «Прізвище, ім'я». Дату виконання практичної можна побачити навівши мишею на відповідний результат. Аналогічним способом можна побачити й назву практичної роботи. Вигляд вкладки «Журнал» наведений на Рисунку 3.23 – Оцінки учнів на вкладці «Журнал».



Виберіть клас ▾	Прізвище, ім'я	Тема 1	Тема 2	Тема 3
11	Антонов Антон	11	0	0
11	Болгов Артем	0	9	11

Рисунок 3. 23 – Оцінки учнів на вкладці «Журнал»

Для отримання результатів учня з бази даних використаний метод `yii\db\ActiveRecord::findBySql()`:

```
$sql_name = 'SELECT distinct u.id, u.name, u.surname from users
u join results r on r.pupil_id = u.id where u.grade = :grade';
$res_name = Users::findBySQL($sql_name, ['grade' =>
$_GET['grade']])->all();
$sql_section = 'SELECT distinct t.section, t.id, t.name from
tests t join results r on r.test_id = t.id where t.grade =
:grade';
$res_section = Tests::findBySQL($sql_section, ['grade' =>
$_GET['grade']])->all(); ?>
```

Сортування та фільтрація даних реалізовані аналогічно відповідним функціям на вкладці «Мої результати» для учнів. Для відображення рядків таблиці з результатами використовується цикл `foreach` мовою PHP та відповідні HTML-теги:

```

foreach ($res_name as $name): ?>
    <tr class="success">
    <td><?php echo $_GET['grade']; ?></td>
    <td><?php echo $name->surname . ' ' . $name->name; ?></td>
    <?php foreach ($res_section as $test) {
        $sql_score = 'SELECT r.score, r.completion_date from
        tests t join (select test_id, max(score) score,
        completion_date from results where pupil_id = :pupil_id
        and test_id = :test_id group by pupil_id) r on r.test_id
        = t.id where t.section = :section';
        $res_score = Results::findBySQL($sql_score, ['pupil_id'
        => $name->id, 'test_id' => $test->id, 'section' => $test-
        >section])->all();
        $i = 0; $mark = 0; $date = '';
        foreach ($res_score as $score) {
            $mark = $score->score;
            $date = $score->completion_date;} ?>
    <td><a href="#" data-toggle="tooltip" title="<?php echo
    $date; ?>"><?php echo round($mark / 100 * 12);
    ?></a></td> <?php }
    <td><?php echo ' '; ?></td></tr>
    <?php $j = $j + 1;
    endforeach; ?>

```

### 3.4 Тестування інформаційної веб-системи

Розроблена інформаційна веб-система була протестована у наступних ситуаціях:

1. Залишити порожнім поле при заповненні форми авторизації.

Інформаційна веб-система реагує відображенням повідомлення про необхідність заповнення поля – Рисунок 3.24 – Повідомлення про необхідність заповнення поля.

**Авторизація**

Будь ласка, заповніть указані поля для входу в систему:

Логін

Пароль

Будь ласка, введіть пароль

Рисунок 3.24 – Повідомлення про необхідність заповнення поля

2. Ввести неправильний пароль при заповненні форми авторизації.

Інформаційна веб-система реагує відображенням повідомлення про неправильні вхідні дані – Рисунок 3.25 – Повідомлення про неправильні вхідні дані.

**Авторизація**

Будь ласка, заповніть указані поля для входу в систему:

Логін

Пароль

Неправильні вхідні дані.

Рисунок 3.25 – Повідомлення про неправильні вхідні дані

3. Авторизуватися з правильними вхідними даними.

Інформаційна веб-система реагує направленням користувача на головну сторінку, відображенням вкладки «Вийти (<логін користувача>)» й інших вкладок у відповідності до його ролі – Рисунок 3.26 – Успішна авторизація користувача.

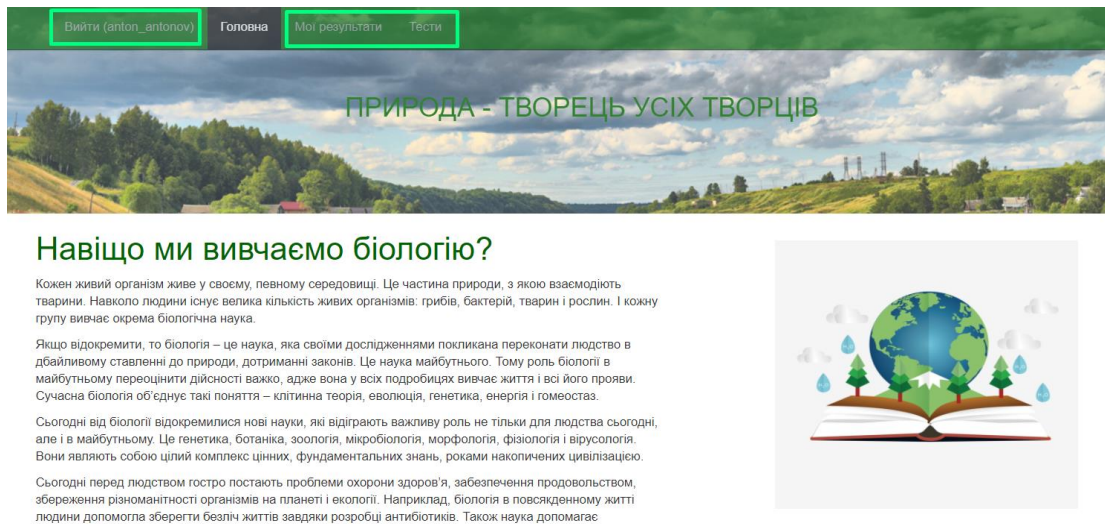


Рисунок 3.26 – Успішна авторизація користувача

4. Перейти на вкладку, доступ до якої заборонений, за посиланням.

При намаганні перейти на вкладку адміністрування користувачем-учнем інформаційна веб-система відображає повідомлення про заборону доступу до даної сторінки – Рисунок 3.27 – Повідомлення про заборону доступу до сторінки.

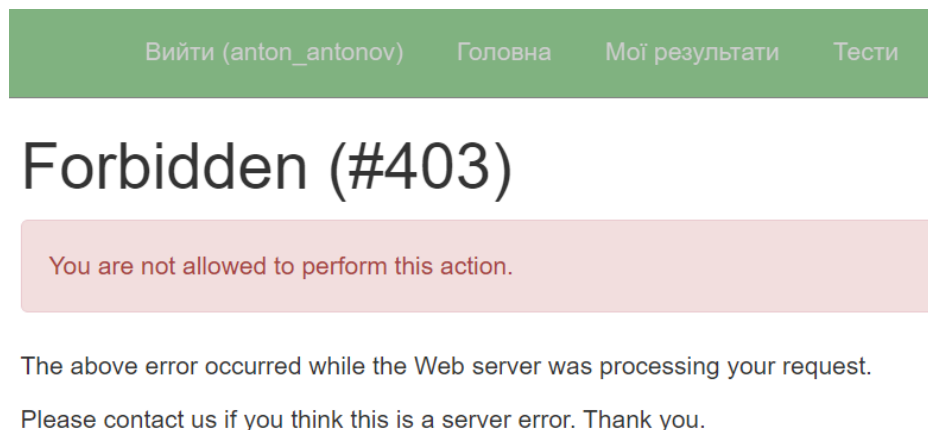


Рисунок 3.27 – Повідомлення про заборону доступу до сторінки

5. Правильно виконати завдання практичної роботи.

Інформаційна веб-система реагує виділенням елементів зеленою рамкою –




Рисунок 3.28 – Правильно виконане завдання практичної роботи.

Вийти (anton\_antonov) Головна Мої результати Тести




### Практична робота: Тема 1 Популяція. Екосистема. Біосфера

1. Залежно від природи та особливостей дії екологічні фактори ділять на абіотичні, біотичні та антропогенні. Розподіліть наведені екологічні фактори за групами, до яких вони належать.

АБІОТИЧНІ

БІОТИЧНІ

АНТРОПОГЕННІ








Рисунок 3.28 – Правильно виконане завдання практичної роботи

6. Неправильно виконати завдання практичної роботи.

Інформаційна веб-система реагує виділенням елементів червоним кольором –

Рисунок 3.29 – Неправильно виконане завдання практичної роботи.

Вийти (anton\_antonov) Головна Мої результати Тести

### Практична робота: Тема 1 Популяція. Екосистема. Біосфера

2. Встановіть відповідність між формами діяльності людини та їхніми наслідками. !

ЕРОЗІЯ ҐРУНТІВ <input type="checkbox"/>	ІНТЕНСИВНА ДІЯЛЬНІСТЬ ПРОМИСЛОВОСТІ ТА ТРАНСПОРТУ <input type="checkbox"/>
ЗАСОЛЕННЯ ҐРУНТІВ <input type="checkbox"/>	ВИКИДИ В АТМОСФЕРУ ХЛОРОФТОРВУГЛЕЦЕВИХ СПОЛУК <input type="checkbox"/>
УТВОРЕННЯ ОЗОНОВИХ ДІРОК <input type="checkbox"/>	МАСОВЕ ВИРУБУВАННЯ ЛІСІВ <input type="checkbox"/>
ТЕПЛИЧНИЙ ЕФЕКТ <input type="checkbox"/>	НЕРАЦІОНАЛЬНЕ ПОЛИВАННЯ ҐРУНТІВ <input type="checkbox"/>

Рисунок 3.29 – Неправильно виконане завдання практичної роботи

7. Створити нового користувача.

Заповнена форма створення нового користувача наведена на Рисунку 3.30 –

Заповнена форма створення нового користувача.

Вийти (admin) Головна Адміністрування Тести

## Створити користувача

Логін  
test\_user

Пароль  
123

Ім'я  
Test

Прізвище  
User

Клас  
7

Учитель  
0

Адміністратор  
0

Зберегти

Рисунок 3.30 – Заповнена форма створення нового користувача

Сторінка перегляду нового користувача, що відкривається після натискання кнопки «Зберегти», наведена на Рисунку 3.31 – Сторінка перегляду користувача.

Вийти (admin) Головна Адміністрування Тести

## Test User

Редагувати Видалити

ІД	13
Логін	test_user
Пароль	123
Ім'я	Test
Прізвище	User
Клас	7
Учитель	0
Адміністратор	0

Рисунку 3.31 – Сторінка перегляду користувача

Пошук нового користувача в таблиці зі списком користувачів наведений на Рисунку 3.32 – Пошук користувача в списку.



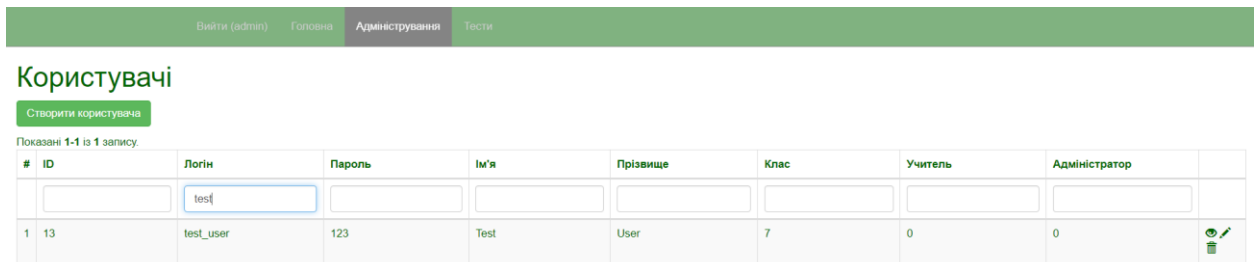


Рисунок 3.32 – Пошук користувача в списку

8. Відредагувати дані про користувача.

Заповнена форма редагування існуючого користувача наведена на Рисунку 3.33 – Заповнена форма редагування користувача.

Рисунок 3.33 – Заповнена форма редагування користувача

Сторінка перегляду відредагованого користувача, що відкривається після натискання кнопки «Зберегти», наведена на Рисунку 3.34 – Сторінка перегляду користувача.

Test User	
ID	13
Логін	test_user
Пароль	123
Ім'я	Test
Прізвище	User
Клас	(не задано)
Учитель	1
Адміністратор	0

Рисунок 3.34 – Сторінка перегляду користувача

## 9. Видалити користувача.

При натисканні кнопки «Видалити» інформаційна веб-система запитує підтвердження виконання цієї операції. Повідомлення, що з'являється при цьому, наведене на Рисунку 3.35 – Запит підтвердження операції видалення користувача.

Test User	
ID	13
Логін	test_user
Пароль	123
Ім'я	Test
Прізвище	User
Клас	(не задано)
Учитель	1
Адміністратор	0

Рисунок 3.35 – Запит підтвердження операції видалення користувача

Після видалення система направляє користувача на сторінку зі списком користувачів. Пошук видаленого користувача результатів не дає – Рисунок 3.36 – Спроба пошуку видаленого користувача.

Користувачі								
#	ID	Логін	Пароль	Ім'я	Прізвище	Клас	Учитель	Адміністратор
		test						
Нічого не знайдено.								

Рисунок 3.36 – Спроба пошуку видаленого користувача

10. Переглянути в журналі результати учнів 11 класу, відсортовані за прізвищем у порядку спадання.

При виборі 11 класу інформаційна веб-система відображає результати учнів лише 11 класу. При натисканні на назву стовпця «Прізвище, ім'я» інформаційна веб-система сортує таблицю за цим стовпцем у порядку спадання. Результат виконання цих дій наведений на Рисунку 3.37 – Таблиця с результатами учнів 11 класу, відсортована за стовпцем «Прізвище, ім'я» в порядку спадання.

Вийти (olha\_suhovii) Головна Адміністрування Тести Журнал

### Результати учнів

Пошук за прізвищем...

Виберіть клас ▾

Клас	Прізвище, ім'я	Тема 1	Тема 2	Тема 3
11	Болгов Артем	0	9	11
11	Антонов Антон	11	0	0

Рисунок 3.37 – Таблиця с результатами учнів 11 класу, відсортована за стовпцем «Прізвище, ім'я» в порядку спадання

## ВИСНОВОК

На початковому етапі виконання магістерської роботи було проведено дослідження рівня інформатизації в загальноосвітніх школах. Був виконаний пошук та аналіз веб-додатків для тестування й інтерактивного навчання, призначених для вчителів і учнів. Дослідження показало, що більшість веб-систем є примітивними й не відповідають рівню розвитку інформаційних технологій в Україні. Значна частина веб-додатків заснована на питаннях з однією або декількома відповідями, які не викликають в учнів інтерес до навчання. Елементи інтерактивності наявні в невеликій кількості додатків та є однотипними. З урахуванням того, що не всі школи мають сучасне апаратне та програмне забезпечення, було зроблено висновок, що використання веб-технологій є найкращим рішенням для українських шкіл. Поставлена задача розробки інформаційної веб-системи для забезпечення інтерактивного навчання.

Відповідно до поставленої задачі, спроектована й створена база даних із використанням СУБД MySQL. Були виконані проектування, розробка на основі фреймворку Yii та тестування інформаційної веб-системи для інтерактивного навчання. Отримане рішення призначене для використання учнями та вчителями на уроках під час практичних робіт з біології. Інформаційна веб-система виконує функцію авторизації попередньо зареєстрованих адміністратором користувачів, забезпечує контроль доступу на основі їх ролей, надає можливості адміністрування вчителям. Крім цього, для учнів є функція перегляду власних оцінок, для вчителів – журналу класу. Засобами мови JavaScript розроблені концептуальні зразки інтерактивних завдань та алгоритм перевірки правильності їх виконання.

Тестування показало, що інформаційна веб-система успішно виконує свої функції та при належному доопрацюванні, а саме наповненості практичними роботами, вона може бути запропонована до інтеграції в навчальний процес.

## СПИСОК ЛИТЕРАТУРИ

1. Кириченко А.В, Хрусталёв А.А HTML5+CSS3. Основы современного Web-дизайна – СПб.: «Наука и техника», 2018 г. – 352 с., ил.
2. Основные различия между веб-сайтом и веб-приложением [Электронный ресурс] URL: <http://artismedia.by/blog/osnovnye-razlichiya-mezhdu-veb-sajtom-i-veb-prilozheniem/>
3. МакГрат, Майк. PHP7 для начинающих с пошаговыми инструкциями / Майк МакГрат. – Москва : Издательство «Э», 2018. 256 с. – (Программирование – это просто).
4. Общие сведения о веб-приложениях [Электронный ресурс] URL: <https://helpx.adobe.com/ru/dreamweaver/using/web-applications.html>
5. Кузнецов М.В, Симдянов И.В PHP 7 – «БХВ-Петербург», 2020 г. – 448 с.
6. Принципы работы и структура Web-приложений на основе ASP.NET [Электронный ресурс] URL: <https://www.sibsau.ru/sveden/edufiles/145507/>
7. Дронов В.А JavaScript. 20 уроков для начинающих – «БХВ-Петербург», 2021 г. – 352 с.
8. Урок 1. Web-приложение: понятие, компоненты и принципы работы [Электронный ресурс] URL: <https://smartiqa.ru/courses/web/lesson-1>
9. Флёнов М.Е PHP. Объекты, шаблоны и методики программирования – Москва: «Диалектика», 2020 г. – 736 с.
10. Самые важные архитектурные шаблоны, которые нужно знать [Электронный ресурс] URL: <https://habr.com/ru/company/alconost/blog/522662/>
11. Котеров Д.В, Симдянов И.В PHP 7 – «БХВ-Петербург», 2020 г. – 1088 с.
12. Kyrylo Kovalenko. Model View Controller (MVC) опыт использования, выводы [Электронный ресурс] URL: <https://habr.com/ru/post/249263/>

13. Скляр Д. Изучаем PHP 7. Руководство по созданию интерактивных веб-сайтов – Москва: «Диалектика», 2020 г. – 464 с.
14. Дмитрий Ченгаев. Шаблон проектирования (программирования) MVC. [Электронный ресурс] URL: <https://webkyrs.info/page/shablon-proektirovaniia-programmirovaniia-mvc>
15. Дронов В.А PHP и MySQL. 25 уроков для начинающих – «БХВ-Петербург», 2021 г. – 432 с.
16. Паттерны для новичков: MVC vs MVP vs MVVM [Электронный ресурс] URL: <https://habr.com/ru/post/215605/>
17. Фленаган Д. JavaScript. Карманный справочник – Москва: «Диалектика», 2020 г. – 320 с.
18. Охота на мифический MVC. Обзор, возвращение к первоисточникам и про то, как анализировать и выводить шаблоны самому [Электронный ресурс] URL: <https://habr.com/ru/post/321050/>
19. Фленаган Д. JavaScript. Полное руководство. 7-е издание – Москва: «Диалектика», 2021 г. – 722 с.
20. MVC — модель-представление-контроллер [Электронный ресурс] URL: <https://web-creator.ru/articles/mvc>
21. Минник К., Холланд Е. JavaScript для чайников – Москва: «Диалектика», 2019 г. – 320 с.
22. Фреймворк - важный инструмент программиста. Обзор HTML/CSS, PHP, JavaScript и Python-фреймворков и библиотек [Электронный ресурс] URL: <https://fructcode.com/ru/blog/features-of-popular-frameworks-html-css-php-and-python-frameworks/>
23. Атенцио Л. Функциональное программирование на JavaScript. Как улучшить код JavaScript-программ – Москва: «Диалектика», 2019 г. – 304 с.
24. Фронтенд фреймворки: собственные решения против готовых к использованию [Электронный ресурс] URL: <https://canonium.com/articles/html-front-end-frameworks-custom-vs-ready-use-solutions/>

25. Браун Э. Изучаем JavaScript. Руководство по созданию современных веб-сайтов – Москва: «Диалектика», 2020 г. – 368 с.
26. Framework [Электронный ресурс] URL: <https://drozd.red/glossary/framework/>
27. Резиг Дж., Бибо Б., Марас И Секреты JavaScript ниндзя – Москва: «Диалектика», 2019 г. – 544 с.
28. Фреймворки для веб-разработки [Электронный ресурс] URL: <https://voll.com.ua/blog/frejmvorki-dlya-veb-razrabotki>
29. Роббинс Дж. Н. Веб-дизайн для начинающих. HTML, CSS, JavaScript и веб-графика – «БХВ-Петербург», 2021 г. – 956 с.
30. Что такое фреймворки и для чего они нужны? [Электронный ресурс] URL: <https://saikt-online.ru/chto-takoe-frejmvorki-i-dlya-chego-oni-nuzhny/>
31. Веллинг Л., Томсон Л. Разработка веб-приложений с помощью PHP и MySQL – СПб: ООО «ДиаСофтЮП», 2017 г. – 768 с.
32. Разработка со стороны Front end [Электронный ресурс] URL: что это такое и чем отличается от Back end? – <https://dan-it.com.ua/blog/razrabotka-so-storony-front-end-chto-jeto-takoe-i-chem-otlichaetsja-ot-back-end/>
33. Кириченко А., Никольский А., Дубовик Е. Web на практике. CSS, HTML, JavaScript, MySQL, PHP для fullstack-разработчиков – СПб.: «Наука и техника», 2021 г. – 432 с.
34. Frontend и backend: различия, особенности и требования к специалистам [Электронный ресурс] URL: <https://timeweb.com/ru/community/articles/frontend-i-backend-razlichiya-osobennosti-i-trebovaniya-k-specialistam>
35. Зандстра М. PHP 8: объекты, шаблоны и методики программирования, 6-е издание – Москва: «Диалектика», 2021 г. – 866 с.
36. Александр Мальцев. Что такое Bootstrap и зачем он нужен? [Электронный ресурс] URL: <https://itchief.ru/bootstrap/introduction>

37. Поляков Е.В. PHP на примерах – СПб.: «Наука и техника», 2017 г. – 256 с.
38. Плюсы и минусы Bootstrap [Электронный ресурс] URL: <https://timeweb.com/ru/community/articles/plyusy-i-minusy-bootstrap-1>
39. Никсон Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5 – СПб.: «Наука и техника», 2019 г. – 816 с.
40. BOOTSTRAP - фреймворк для адаптивной разработки [Электронный ресурс] URL: <https://redline.by/novosti/bootstrap.html>
41. Васильев А. Н. Программирование на PHP в примерах и задачах – Москва: «Эксмо», 2021 г. – 352 с.
42. 5 лучших фреймворков для верстки: как они устроены и есть ли у них минусы [Электронный ресурс] URL: <https://ru.hexlet.io/blog/posts/5-luchshih-freymvorkov-dlya-verstki>
43. Лукьянов М. Ю. PHP. Полное руководство и СПРАВОЧНИК функций – СПб.: «Наука и техника», 2020 г. – 432 с.
44. Фреймворк Foundation для создания адаптивных сайтов на WordPress [Электронный ресурс] URL: <https://oddstyle.ru/wordpress-2/stati-wordpress/frejmvork-foundation-dlya-sozdaniya-adaptivnyx-sajtov-na-wordpress.html>
45. Фримен Э., Робсон Э. Изучаем программирование на JavaScript – СПб.: «Наука и техника», 2018 г. – 640 с.
46. 9 лучших фреймворков CSS, актуальных в 2021 году [Электронный ресурс] URL: <https://techrocks.ru/2021/03/13/9-css-frameworks-for-2021/>
47. Хавербеке М. Выразительный JavaScript. Современное веб-программирование – СПб.: «Наука и техника», 2019 г. – 480 с.
48. Лучшие CSS-фреймворки для Front-End разработчиков в 2021 году [Электронный ресурс] URL: <https://senior.ua/articles/luchshie-cssfreymvorki-dlya-frontend-razrabotchikov-v-2021-godu>



49. Чиннатхамби К. JavaScript с нуля – СПб: ООО «Прогресс Книга», 2021 г. – 400 с.
50. Юрий Мусиенко. Самые популярные CSS фреймворки в 2020 для мобильных устройств [Электронный ресурс] URL: <https://merehead.com/ru/blog/trends-of-css-frameworks-2020/>
51. Симпсон К. (Вы пока еще не знаете JS) Познакомьтесь, JavaScript. 2-е изд. – СПб: ООО «Прогресс Книга», 2021 г. – 192 с.
52. Игорь Антонов. Обзор CSS-фреймворков [Электронный ресурс] URL: <http://iantonov.me/page/obzor-css-frejmworkov>
53. Андрей Павленко. Что такое backend и frontend? [Электронный ресурс] URL: <https://otus.ru/nest/post/943/>
54. Frontend- и backend-разработка на примере создания сайтов [Электронный ресурс] URL: [https://skillbox.ru/media/code/frontend\\_i\\_backend/](https://skillbox.ru/media/code/frontend_i_backend/)
55. Значение бэкенда и все, что вам нужно знать о нем [Электронный ресурс] URL: <https://blog.back4app.com/ru/значение-бэкенда-и-все-что-вам-нужно-зн/>
56. Чем занимаются бэкенд-разработчики [Электронный ресурс] URL: <https://thecode.media/backend/>
57. Фреймворк Yii [Электронный ресурс] URL: <https://unetway.com/blog/yii-framework-review>
58. Преимущества и возможности Yii framework для создания веб-сайтов [Электронный ресурс] URL: <https://wezom.com.ua/blog/yii-framework-dlya-sozdaniya-veb-sajtov>
59. Yii [Электронный ресурс] URL: фреймворк для веб-разработки на PHP – <https://web-creator.ru/articles/yii>
60. Преимущества и недостатки разработки на Yii [Электронный ресурс] URL: <https://www.castcom.ru/publications/web/preimuschestva-i-nedostatki-razrabotki-na-yii.html>

61. Артур Казимирский. Топ преимуществ Laravel в разработке коммерческих веб-приложений [Электронный ресурс] URL: <https://solardigital.com.ua/blog/razrabotka-na-laravel>
62. 17 Преимуществ Laravel [Электронный ресурс] URL: <https://blog.webformula.pro/article/17-preimushchestv-laravel/>
63. Фреймворк Laravel [Электронный ресурс] URL: <https://unetway.com/blog/laravel-framework-review>
64. Фреймворк Symfony [Электронный ресурс] URL: <https://unetway.com/blog/symfony-framework-review>
65. Дмитрий Ченгаев. Что такое Symfony? [Электронный ресурс] URL: <https://webkyrs.info/page/chtotakoe-symfony>
66. Главные причины, почему мы разрабатываем веб-приложения на Symfony [Электронный ресурс] URL: <https://habr.com/ru/company/otus/blog/564532/>
67. Плюсы и минусы веб-разработки на Symfony [Электронный ресурс] URL: <https://www.developers.dev/tech-talk/ru/technology/the-good-and-the-bad-of-symfony-web-development.html>
68. Что выбрать: Yii или Zend? [Электронный ресурс] URL: <https://www.castcom.ru/publications/web/chtovybrat-yii-ili-zend.html>
69. Фреймворк Zend [Электронный ресурс] URL: <https://unetway.com/blog/zend-framework-review>
70. CodeIgniter [Электронный ресурс] URL: Краткое руководство - <https://coderlessons.com/tutorials/web-razrabotka/vyuchit-codeigniter/codeigniter-kratkoe-rukovodstvo>
71. Обзор фреймворка CodeIgniter [Электронный ресурс] URL: <https://habr.com/ru/post/168015/>
72. Фреймворк CodeIgniter [Электронный ресурс] URL: <https://unetway.com/blog/codeigniter-framework-review>

73. Phalcon [Электронный ресурс] URL: Краткое руководство - <https://coderlessons.com/tutorials/veb-razrabotka/vyuchit-phalcon/phalcon-kratkoe-rukovodstvo>
74. Phalcon [Электронный ресурс] URL: скомпилированный PHP MVC Framework – <https://habr.com/ru/post/159217/>
75. Знакомство с Phalcon PHP [Электронный ресурс] URL: <https://webformyself.com/znakomstvo-s-phalcon-php/>
76. CakePHP [Электронный ресурс] URL: что это такое и с чем его едят. PHP фреймворк - <http://www.znannya.org/?view=cakephp>
77. CakePHP [Электронный ресурс] URL: <https://bizzapps.ru/p/cakephp/>
78. CakePHP [Электронный ресурс] URL: Краткое руководство - <https://coderlessons.com/tutorials/veb-razrabotka/uznaite-cakephp/cakephp-kratkoe-rukovodstvo>
79. Архитектура веба: основы для начинающих разработчиков [Электронный ресурс] URL: <https://medium.com/webbdev/webwebweb-c61f5fe2232a>
80. Аннотация к phpMyAdmin [Электронный ресурс] URL: <https://php-myadmin.ru/about/notice/>
81. Полное руководство по Yii 2.0 [Электронный ресурс] URL: <https://www.yiiframework.com/doc/guide/2.0/ru>
82. Yii [Электронный ресурс] URL: Структура приложения - <https://coderlessons.com/tutorials/veb-razrabotka/vyuchi-yii/yii-struktura-prilozheniia>
83. Назначение папок и файлов в Yii2 [Электронный ресурс] URL: <https://liblessons.ru/programming/framework-yii2/naznachenie-papok-faylov-yii2/>
84. Materialize [Электронный ресурс] URL: <https://bizzapps.ru/p/materialize/>
85. MVC [Электронный ресурс] URL: <https://javarush.ru/quests/lectures/questcollections.level06.lecture01>

## ДОДАТКИ

### Додаток А

У даному додатку наведений код функцій *search()* і *sortTable()*

```
function search() {
    var input, filter, table, tr, td, i;
    input = document.getElementById("input");
    filter = input.value.toUpperCase();
    table = document.getElementById("resultsTable");
    tr = table.getElementsByTagName("tr");
    for (i = 0; i < tr.length; i++) {
        td = tr[i].getElementsByTagName("td")[1];
        if (td) {
            if (td.innerHTML.toUpperCase().indexOf(filter) > -1) {
                tr[i].style.display = "";
            } else {
                tr[i].style.display = "none"; }}}
}

function sortTable(n, t) {
    var table, rows, switching, i, x, y, shouldSwitch, dir,
    switchcount = 0;
    table = document.getElementById("resultsTable");
    switching = true;
    dir = "asc";
    while (switching) {
        switching = false;
        rows = table.getElementsByTagName("TR");
        for (i = 1; i < (rows.length - 1); i++) {
            shouldSwitch = false;
            x = rows[i].getElementsByTagName("TD")[n];
            y = rows[i + 1].getElementsByTagName("TD")[n];
            if (dir == "asc") {
                if (t == 1) {
                    if (Number(x.innerHTML) > Number(y.innerHTML)) {
                        shouldSwitch = true;
                        break;}}
                if (t == 0) {
                    if (x.innerHTML.toLowerCase() >
                        y.innerHTML.toLowerCase()) {
                        shouldSwitch = true;
                        break;}}
            } else if (dir == "desc") {
                if (t == 1) {
                    if (Number(x.innerHTML) < Number(y.innerHTML)) {
                        shouldSwitch = true;
                        break;}}
                if (t == 0) {
```

```
        if (x.innerHTML.toLowerCase() <
            y.innerHTML.toLowerCase()) {
            shouldSwitch = true;
            break;}}}}
if (shouldSwitch) {
    rows[i].parentNode.insertBefore(rows[i + 1], rows[i]);
    switching = true;
    switchcount ++;
} else {
if (switchcount == 0 && dir == "asc") {
    dir = "desc";
    switching = true;}}}}
```

## Додаток Б

У даному додатку наведений код першого завдання

```
$(function() {
    $("#btn").click(function() {
        let mark = 0;
        let barometr = document.querySelector('#barometr');
        let bar_x = barometr.getBoundingClientRect().x;
        let bar_y = barometr.getBoundingClientRect().y;

        let water = document.querySelector('#water');
        let wat_x = water.getBoundingClientRect().x;
        let wat_y = water.getBoundingClientRect().y;

        let thermometer =
document.querySelector('#thermometer');
        let therm_x = thermometer.getBoundingClientRect().x;
        let therm_y = thermometer.getBoundingClientRect().y;

        let column1 = document.querySelector('#column1');
        let coll_x1 = column1.getBoundingClientRect().x;
        let width1 = column1.getBoundingClientRect().width;
        let coll_x2 = coll_x1 + width1;
        let height1 = column1.getBoundingClientRect().height;
        let coll_y1 = column1.getBoundingClientRect().y;
        let coll_y2 = coll_y1 + height1;
        if (bar_x >= coll_x1 && bar_x <= coll_x2 && bar_y >=
coll_y1 && bar_y <= coll_y2) {
            mark = mark + 1;
            document.querySelector('#barometr').style.border =
"2px solid darkgreen";
        }
        else {
            document.querySelector('#barometr').style.border =
"2px solid darkred";
        }

        if (wat_x >= coll_x1 && wat_x <= coll_x2 && wat_y >=
coll_y1 && wat_y <= coll_y2) {
            mark = mark + 1;
            document.querySelector('#water').style.border = "2px
solid darkgreen";
        }
        else {
            document.querySelector('#water').style.border = "2px
solid darkred";
        }

        if (therm_x >= coll_x1 && therm_x <= coll_x2 &&
therm_y >= coll_y1 && therm_y <= coll_y2) {
            mark = mark + 1;
        }
    }
});
```

```

        document.querySelector('#thermometer').style.border
= "2px solid darkgreen";
    }
    else {
        document.querySelector('#thermometer').style.border
= "2px solid darkred";
    }

    let reproduction =
document.querySelector('#reproduction');
    let reproduct_x =
reproduction.getBoundingClientRect().x;
    let reproduct_y =
reproduction.getBoundingClientRect().y;

    let symbiosis = document.querySelector('#symbiosis');
    let symb_x = symbiosis.getBoundingClientRect().x;
    let symb_y = symbiosis.getBoundingClientRect().y;

    let parasite = document.querySelector('#parasite');
    let par_x = parasite.getBoundingClientRect().x;
    let par_y = parasite.getBoundingClientRect().y;

    let column2 = document.querySelector('#column2');
    let col2_x1 = column2.getBoundingClientRect().x;
    let width2 = column2.getBoundingClientRect().width;
    let col2_x2 = col2_x1 + width2;
    let height2 = column2.getBoundingClientRect().height;
    let col2_y1 = column2.getBoundingClientRect().y;
    let col2_y2 = col2_y1 + height2;

    if (reproduct_x >= col2_x1 && reproduct_x <= col2_x2
&& reproduct_y >= col2_y1 && reproduct_y <= col2_y2) {
        mark = mark + 1;
        document.querySelector('#reproduction').style.border
= "2px solid darkgreen";
    }
    else {
        document.querySelector('#reproduction').style.border
= "2px solid darkred";
    }

    if (symb_x >= col2_x1 && symb_x <= col2_x2 && symb_y
>= col2_y1 && symb_y <= col2_y2) {
        mark = mark + 1;
        document.querySelector('#symbiosis').style.border =
"2px solid darkgreen";
    }
    else {
        document.querySelector('#symbiosis').style.border =
"2px solid darkred";
    }

```

```

        if (par_x >= col2_x1 && par_x <= col2_x2 && par_y >=
col2_y1 && par_y <= col2_y2) {
            mark = mark + 1;
            document.querySelector('#parasite').style.border =
"2px solid darkgreen";
        }
        else {
            document.querySelector('#parasite').style.border =
"2px solid darkred";
        }

        let pond = document.querySelector('#pond');
        let pond_x = pond.getBoundingClientRect().x;
        let pond_y = pond.getBoundingClientRect().y;

        let chemicals = document.querySelector('#chemicals');
        let chem_x = chemicals.getBoundingClientRect().x;
        let chem_y = chemicals.getBoundingClientRect().y;

        let radiowaves =
document.querySelector('#radiowaves');
        let radio_x = radiowaves.getBoundingClientRect().x;
        let radio_y = radiowaves.getBoundingClientRect().y;

        let column3 = document.querySelector('#column3');
        let col3_x1 = column3.getBoundingClientRect().x;
        let width3 = column3.getBoundingClientRect().width;
        let col3_x2 = col3_x1 + width3;
        let height3 = column3.getBoundingClientRect().height;
        let col3_y1 = column3.getBoundingClientRect().y;
        let col3_y2 = col3_y1 + height3;

        if (pond_x >= col3_x1 && pond_x <= col3_x2 && pond_y
>= col3_y1 && pond_y <= col3_y2) {
            mark = mark + 1;
            document.querySelector('#pond').style.border = "2px
solid darkgreen";
        }
        else {
            document.querySelector('#pond').style.border = "2px
solid darkred";
        }

        if (chem_x >= col3_x1 && chem_x <= col3_x2 && chem_y
>= col3_y1 && chem_y <= col3_y2) {
            mark = mark + 1;
            document.querySelector('#chemicals').style.border =
"2px solid darkgreen";
        }
        else {

```



```

        document.querySelector('#chemicals').style.border =
"2px solid darkred";
    }

    if (radio_x >= col3_x1 && radio_x <= col3_x2 &&
radio_y >= col3_y1 && radio_y <= col3_y2) {
        mark = mark + 1;
        document.querySelector('#radiowaves').style.border =
"2px solid darkgreen";
    }
    else {
        document.querySelector('#radiowaves').style.border =
"2px solid darkred";
    }
    console.log(mark);

document.querySelector('#btn').setAttribute("disabled", "true");
$('#btn2').removeAttr('disabled');
document.querySelector('#input').value = mark;

<script>
function handleDragOver(e) {
    if (e.preventDefault) {
        e.preventDefault();
    }
    e.dataTransfer.dropEffect = 'move';
    return false;
}

function handleDragEnter(e) {
    this.classList.add('over');
}

function handleDragLeave(e) {
    this.classList.remove('over');
}

function handleDrop(e) {
    if (e.stopPropagation) {
        e.stopPropagation();
    }
    return false;
}

function handleDragEnd(e) {

    [].forEach.call(cols, function (col) {
        col.classList.remove('over');
    });
}

var cols = document.querySelectorAll('#columns .column');
```

```

    []).forEach.call(cols, function(col) {
        col.addEventListener('dragstart', handleDragStart, false);
        col.addEventListener('dragenter', handleDragEnter, false);
        col.addEventListener('dragover', handleDragOver, false);
        col.addEventListener('dragleave', handleDragLeave, false);
        col.addEventListener('drop', handleDrop, false);
        col.addEventListener('dragend', handleDragEnd, false);
    });
</script>
<div id="columns">
    <div class="column" id="column1" draggable="false"
ondragenter="dragEnter(event)" ondrop="dragDrop(event)"
ondragover="dragOver(event)"><header><strong>АБІОТИЧНИ</strong></
header></div>
    <div class="column" id="column2" draggable="false"
ondragenter="dragEnter(event)" ondrop="dragDrop(event)"
ondragover="dragOver(event)"><header><strong>БІОТИЧНИ</strong></
header></div>
    <div class="column" id="column3" draggable="false"
ondragenter="dragEnter(event)" ondrop="dragDrop(event)"
ondragover="dragOver(event)"><header><strong>АНТРОПОГЕННИ</stron
g></header></div>
</div>
<tr>
    <td></td>
    <td></td>
    <td></td>
    <td></td>
    <td></td>
    <td></td>
    <td></td>
    <td></td>
    <td></td>
    <td></td></tr>

```

## Додаток В

У даному додатку наведений код другого завдання

```
function drawLine(n, m, color){
    var canvas = document.getElementById("myCanvas"),
        context = canvas.getContext("2d");
    if (color == 0) {
        context.strokeStyle = "darkgreen";
    }
    else if (color == 1){
        context.strokeStyle = "darkred";
    }
    context.strokeStyle = color;
    var column = document.querySelector('#column1');
    var col_height = column.getBoundingClientRect().height;
    context.beginPath();
    context.moveTo(0, col_height / 2 + col_height * n + 10 *
n);
    context.lineTo(270, col_height / 2 + col_height * m + 10
* m);
    context.stroke();
    context.closePath();
};
```

. . .

```
<div id="columns">
```

```
    <div id="firstColumn">
```

```
        <div class="column" id="column1" draggable="false"
ondragenter="dragEnter(event)" ondrop="dragDrop(event)"
ondragover="dragOver(event)"><p><b>ЕРОЗІЯ ГРУНТІВ</b></p><input
type="checkbox" id="radio1" class="checking" value="1"></div>
```

```
    <script>
```

```
        var checkbox_1 = document.getElementById("radio1");
        checkbox_1.onclick = function() {
            document.getElementById("radio2").checked = false;
            document.getElementById("radio3").checked = false;
            document.getElementById("radio4").checked = false;
        };
    </script>
```

```
        <div class="column" id="column2" draggable="false"
ondragenter="dragEnter(event)" ondrop="dragDrop(event)"
ondragover="dragOver(event)"><b>ЗАСОЛЕННЯ ГРУНТІВ</b><input
type="checkbox" id="radio2" class="checking" value="2"></div>
```

```

<script>
    var checkbox_2 = document.getElementById("radio2");
    checkbox_2.onclick = function() {
        document.getElementById("radio1").checked = false;
        document.getElementById("radio3").checked = false;
        document.getElementById("radio4").checked = false;
    };
</script>

<div class="column" id="column3" draggable="false"
ondragenter="dragEnter(event)" ondrop="dragDrop(event)"
ondragover="dragOver(event)"><b>УТВОРЕННЯ ОЗОНОВИХ
ДІПОК</b><input type="checkbox" id="radio3" class="checking"
value="3"></div>

<script>
    var checkbox_3 = document.getElementById("radio3");
    checkbox_3.onclick = function() {
        document.getElementById("radio1").checked = false;
        document.getElementById("radio2").checked = false;
        document.getElementById("radio4").checked = false;};
</script>

<div class="column" id="column4" draggable="false"
ondragenter="dragEnter(event)" ondrop="dragDrop(event)"
ondragover="dragOver(event)"><b>ТЕПЛИЧНИЙ ЕФЕКТ</b><input
type="checkbox" id="radio4" class="checking" value="4"></div>
</div>

<script>
    var checkbox_4 = document.getElementById("radio4");
    checkbox_4.onclick = function() {
        document.getElementById("radio1").checked = false;
        document.getElementById("radio2").checked = false;
        document.getElementById("radio3").checked = false;};
</script>

<div id="canvasColumn">
    <canvas id="myCanvas" width="270" height="350"></canvas>
</div>

```

```

<div id="secondColumn">
    <div class="column" id="column5" draggable="false"
ondragenter="dragEnter(event)" ondrop="dragDrop(event)"
ondragover="dragOver(event)"><p><b>ІНТЕНСИВНА
ДІЯЛЬНІСТЬ<br>ПРОМИСЛОВОСТІ ТА<br>ТРАНСПОРТУ</b></p><input
type="checkbox" id="radio5"></div>

<script>
    document.getElementById("radio5").onclick = function() {
        if (document.getElementById("radio1").checked) {
            drawLine(0, 0, 0);
            document.getElementById("radio1").checked = false;
            document.getElementById("radio5").checked = false;
            marks.mark1 = 0;
            marks.line1 = 15;}
        else if (document.getElementById("radio2").checked) {
            drawLine(1, 0, 0);
            document.getElementById("radio2").checked = false;
            document.getElementById("radio5").checked = false;
            marks.mark1 = 0;
            marks.line1 = 25;
        }
        else if (document.getElementById("radio3").checked){
            drawLine(2, 0, 0);
            document.getElementById("radio3").checked = false;
            document.getElementById("radio5").checked = false;
            marks.mark1 = 0;
            marks.line1 = 35;
        }
        else if (document.getElementById("radio4").checked){
            drawLine(3, 0, 0);
            document.getElementById("radio4").checked = false;
            document.getElementById("radio5").checked = false;
            marks.mark1 = 1;
            marks.line1 = 45;
        }
    }
}

```

```

    }
};
</script>

<div class="column" id="column6" draggable="false"
ondragenter="dragEnter(event)" ondrop="dragDrop(event)"
ondragover="dragOver(event)"><p><b>ВИКИДИ В
АТМОСФЕРУ<br>ХЛОПФТОРВУГЛЕЦЕВИХ<br>СПОЛУК</b></p><input
type="checkbox" id="radio6"></div>

<script>
    document.getElementById("radio6").onclick = function() {
        if (document.getElementById("radio1").checked) {
            drawLine(0, 1, 0);
            document.getElementById("radio1").checked = false;
            document.getElementById("radio6").checked = false;
            marks.mark2 = 0;
            marks.line2 = 16;
        }
        else if (document.getElementById("radio2").checked) {
            drawLine(1, 1, 0);
            document.getElementById("radio2").checked = false;
            document.getElementById("radio6").checked = false;
            marks.mark2 = 0;
            marks.line2 = 26;
        }
        else if (document.getElementById("radio3").checked) {
            drawLine(2, 1, 0);
            document.getElementById("radio3").checked = false;
            document.getElementById("radio6").checked = false;
            marks.mark2 = 1;
            marks.line2 = 36;
        }
        else if (document.getElementById("radio4").checked) {
            drawLine(3, 1, 0);
            document.getElementById("radio4").checked = false;

```

```

        document.getElementById("radio6").checked = false;
        marks.mark2 = 0;
        marks.line2 = 46;
    }
};
</script>
<div class="column" id="column7" draggable="false"
ondragenter="dragEnter(event)" ondrop="dragDrop(event)"
ondragover="dragOver(event)"><b>MACOBE BИPYБYBAHHЯ
ЛICIB</b><input type="checkbox" id="radio7"></div>
<script>
    document.getElementById("radio7").onclick = function() {
        if (document.getElementById("radio1").checked) {
            drawLine(0, 2, 0);
            document.getElementById("radio1").checked = false;
            document.getElementById("radio7").checked = false;
            marks.mark3 = 1;
            marks.line3 = 17;
        }
        else if (document.getElementById("radio2").checked) {
            drawLine(1, 2, 0);
            document.getElementById("radio2").checked = false;
            document.getElementById("radio7").checked = false;
            marks.mark3 = 0;
            marks.line3 = 27;
        }
        else if (document.getElementById("radio3").checked) {
            drawLine(2, 2, 0);
            document.getElementById("radio3").checked = false;
            document.getElementById("radio7").checked = false;
            marks.mark3 = 0;
            marks.line3 = 37;
        }
        else if (document.getElementById("radio4").checked) {

```

```

        drawLine(3, 2, 0);
        document.getElementById("radio4").checked = false;
        document.getElementById("radio7").checked = false;
        marks.mark3 = 0;
        marks.line3 = 47;
    }
};
</script>

```

```

<div class="column" id="column8" draggable="false"
ondragenter="dragEnter(event)" ondrop="dragDrop(event)"
ondragover="dragOver(event)"><p><b>НЕРАЦІОНАЛЬНЕ
ПОЛИВАННЯ<br>ГРПУНТИВ</b></p><input type="checkbox"
id="radio8"></div>

```

```

<script>
    document.getElementById("radio8").onclick = function() {
        if (document.getElementById("radio1").checked) {
            drawLine(0, 3, 0);
            document.getElementById("radio1").checked = false;
            document.getElementById("radio8").checked = false;
            marks.mark4 = 0;
            marks.line4 = 18;
        }
        else if (document.getElementById("radio2").checked) {
            drawLine(1, 3, 0);
            document.getElementById("radio2").checked = false;
            document.getElementById("radio8").checked = false;
            marks.mark4 = 1;
            marks.line4 = 28;
        }
        else if (document.getElementById("radio3").checked) {
            drawLine(2, 3, 0);
            document.getElementById("radio3").checked = false;
            document.getElementById("radio8").checked = false;
        }
    }

```



```

        marks.mark4 = 0;
        marks.line4 = 38;
    }
    else if (document.getElementById("radio4").checked) {
        drawLine(3, 3, 0);
        document.getElementById("radio4").checked = false;
        document.getElementById("radio8").checked = false;
        marks.mark4 = 0;
        marks.line4 = 48;
    }
};
</script>
</div>

</div>

<button id="btn3" type="button" class="btn btn-
success">Скасувати</button>

<script>
    var btn = document.querySelector('#btn3');
    document.querySelector('#btn3').onclick = function() {
        var canvas = document.getElementById("myCanvas"),
            context = canvas.getContext("2d");
        context.clearRect(0, 0, 270, 350);
    };
</script>

<button id="btn" type="button" class="btn btn-success">Дати
відповідь</button>

<script>
    var btn = document.querySelector('#btn');
    document.querySelector('#btn').onclick = function() {
        document.getElementById("btn").disabled = true;
        document.getElementById("btn3").disabled = true;
    };
</script>

```

```
document.getElementById("btn2").disabled = false;
console.log(marks.mark1 + marks.mark2 + marks.mark3 +
marks.mark4);
var canvas = document.getElementById("myCanvas"),
context = canvas.getContext("2d");
context.clearRect(0, 0, 270, 350);
if (marks.line1 == 45) {
    drawLine(3, 0, 0);
}
else if (marks.line1 == 15) {
    drawLine(0, 0, 1);
}
else if (marks.line1 == 25) {
    drawLine(1, 0, 1);
}
else if (marks.line1 == 35) {
    drawLine(2, 0, 1);
}
else {

}
if (marks.line2 == 36) {
    drawLine(2, 1, 0);
}
else if (marks.line2 == 16) {
    drawLine(0, 1, 1);
}
else if (marks.line2 == 26) {
    drawLine(1, 1, 1);
}
else if (marks.line2 == 46) {
    drawLine(3, 1, 1);
}
```

```
else {

}

if (marks.line3 == 17) {
    drawLine(0, 2, 0);
}
else if (marks.line3 == 27) {
    drawLine(1, 2, 1);
}
else if (marks.line3 == 37) {
    drawLine(2, 2, 1);
}
else if (marks.line3 == 47) {
    drawLine(3, 2, 1);
}
else {

}

if (marks.line4 == 28) {
    drawLine(1, 3, 0);
}
else if (marks.line4 == 18) {
    drawLine(0, 3, 1);
}
else if (marks.line4 == 38) {
    drawLine(2, 3, 1);
}
else if (marks.line4 == 48) {
    drawLine(3, 3, 1);
}
}
</script>
```

## Додаток Г

У даному додатку наведений код третього завдання

```

<div id="columns">
  <div id="firstColumn">
    <div class="column" id="column1">
      <span class="glyphicon glyphicon-ok" id="sign1"></span>
      <span class="glyphicon glyphicon-remove"
id="sign5"></span>
      <div class="column" id="buttons1">

        <button id="red1" type="button" class="btn btn-
danger">50</button><br>

        <script>
          document.querySelector('#red1').onclick = function()
{
          document.getElementById("book1").style.color =
'#d9534f';
          colors.color1 = 1;
          };
        </script>

        <button id="green1" type="button" class="btn btn-
success">50</button><br>

        <script>
          document.querySelector('#green1').onclick =
function() {
          document.getElementById("book1").style.color =
'#5cb85c';
          colors.color1 = 2;
          };
        </script>
        <button id="black1" type="button" class="btn btn-
success">50</button>

        <script>
          document.querySelector('#black1').onclick =
function() {
          document.getElementById("book1").style.color =
'black';
          colors.color1 = 3;
          };
        </script>
      </div>
      <span class="glyphicon glyphicon-book"
id="book1"></span>
    </div>
    <div class="column" id="column2">
      <span class="glyphicon glyphicon-ok" id="sign2"></span>

```

```

        <span class="glyphicon glyphicon-remove"
id="sign6"></span>
        <div class="column" id="buttons2">
            <button id="red2" type="button" class="btn btn-
danger">50</button><br>
            <script>
                document.querySelector('#red2').onclick = function()
{
                    document.getElementById("book2").style.color =
'#d9534f';
                    colors.color2 = 1;
                };
            </script>
            <button id="green2" type="button" class="btn btn-
success">50</button><br>

            <script>
                document.querySelector('#green2').onclick =
function() {
                    document.getElementById("book2").style.color =
'#5cb85c';
                    colors.color2 = 2;
                };
            </script>
            <button id="black2" type="button" class="btn btn-
success">50</button>

            <script>
                document.querySelector('#black2').onclick =
function() {
                    document.getElementById("book2").style.color =
'black';
                    colors.color2 = 3;
                };
            </script>
        </div>
        <span class="glyphicon glyphicon-book"
id="book2"></span>
        </div>
        <div class="column" id="column3">
            <span class="glyphicon glyphicon-ok" id="sign3"></span>
            <span class="glyphicon glyphicon-remove"
id="sign7"></span>
            <div class="column" id="buttons3">
                <button id="red3" type="button" class="btn btn-
danger">50</button><br>
                <script>
                    document.querySelector('#red3').onclick = function()
{
                        document.getElementById("book3").style.color =
'#d9534f';
                        colors.color3 = 1;
                    }
                </script>
            </div>
        </div>
    </div>
</div>

```

```

    };
</script>
<button id="green3" type="button" class="btn btn-
success">50</button><br>
<script>
    document.querySelector('#green3').onclick =
function() {
    document.getElementById("book3").style.color =
'#5cb85c';
    colors.color3 = 2;
    };
</script>
<button id="black3" type="button" class="btn btn-
success">50</button>
<script>
    document.querySelector('#black3').onclick =
function() {
    document.getElementById("book3").style.color =
'black';
    colors.color3 = 3;
    };
</script>
</div>
<span class="glyphicon glyphicon-book"
id="book3"></span></div>
</div>
<div id="secondColumn">
    <div class="column" id="column5">державний документ про
сучасний стан видів тварин і рослин, які перебувають під
загрозою зникнення, та про заходи щодо їхнього збереження та
науково обґрунтованого відтворення.</div>
    <div class="column" id="column6">державний документ про
рідкісні і типові для певної для певної місцевості рослинні
угруповання, що потребують встановлення особливого режиму їх
використання.</div>
    <div class="column" id="column7">державний документ про
види, які зникли з лиця Землі починаючи з 1600 року. Підставою
для введення виду до книги є відсутність достовірних його
знахідок принаймні протягом останніх 50 років.</div>
</div>
</div>
<button id="btn" type="button" class="btn btn-success">Дати
відповідь</button>
<script>
    var btn = document.querySelector('#btn');
    document.querySelector('#btn').onclick = function() {
    document.getElementById("btn").disabled = true;
    document.getElementById("btn2").disabled = false;
    if (colors.color1 == 1) {
        document.getElementById("sign1").style.color =
'#5cb85c';
    }

```

```
        else{
            document.getElementById("sign5").style.color =
'#d9534f';
        }
        if (colors.color2 == 2) {
            document.getElementById("sign2").style.color =
'#5cb85c';
        }
        else{
            document.getElementById("sign6").style.color =
'#d9534f';
        }
        if (colors.color3 == 3) {
            document.getElementById("sign3").style.color =
'#5cb85c';
        }
        else{
            document.getElementById("sign7").style.color =
'#d9534f';
        }
    };

</script>
```

## Додаток Д

У даному додатку наведений код четвертого завдання

```

<div class="images" align="center" id="images1">
  <table border="0" align="center">
    <tr>
      <td id="td1"></td>
    </tr>
    <tr>
      <td id="td2"><br></td>
    </tr>
    <tr>
      <td id="td3"></td>
    </tr>
  </table>
</div>
<p class="map"></p>
<div class="columns">
  <p id="sumy" class="sumy" ondragenter="dragEnter(event) "
ondrop="dragDrop(event) " ondragover="dragOver(event) "></p>
  <p id="kharkiv" class="kharkiv"
ondragenter="dragEnter(event) " ondrop="dragDrop(event) "
ondragover="dragOver(event) "></p>
  <p id="kherson" class="kherson"
ondragenter="dragEnter(event) " ondrop="dragDrop(event) "
ondragover="dragOver(event) "></p>
  <p id="ternopil" class="ternopil"
ondragenter="dragEnter(event) " ondrop="dragDrop(event) "
ondragover="dragOver(event) "></p>
  <p id="zhytomyr" class="zhytomyr"
ondragenter="dragEnter(event) " ondrop="dragDrop(event) "
ondragover="dragOver(event) "></p>
</div>
<div class="images" align="center" id="images2">
  <table border="0" align="center">
    <tr>
      <td id="td4"></td>
    </tr>
    <tr>
      <td id="td5"><br></td>
    </tr>
  </table>

```



```

    </table>
</div>
<script>
    function handleDragOver(e) {
        if (e.preventDefault) {
            e.preventDefault();
        }
        e.dataTransfer.dropEffect = 'move';
        return false;
    }
    function handleDragEnter(e) {
        this.classList.add('over');
    }
    function handleDragLeave(e) {
        this.classList.remove('over'); // this / e.target is
previous target element.
    }
    function handleDrop(e) {
        if (e.stopPropagation) {
            e.stopPropagation(); // stops the browser from
redirecting.
        }
        return false;
    }
    function handleDragEnd(e) {
        [].forEach.call(cols, function (col) {
            col.classList.remove('over');
        });
    }
    var cols = document.querySelectorAll('#columns .column');
    [].forEach.call(cols, function(col) {
        col.addEventListener('dragstart', handleDragStart, false);
        col.addEventListener('dragenter', handleDragEnter, false);
        col.addEventListener('dragover', handleDragOver, false);
        col.addEventListener('dragleave', handleDragLeave, false);
        col.addEventListener('drop', handleDrop, false);
        col.addEventListener('dragend', handleDragEnd, false);
    });
</script>
</div>
<button id="btn" type="button" class="btn btn-success">Дати
відповідь</button>
<script>
    var btn = document.querySelector('#btn');
    document.querySelector('#btn').onclick = function() {
        document.getElementById("btn").disabled = true;
        document.getElementById("btn3").disabled = true;
        document.getElementById("btn2").disabled = false;
        let mark = 0;
        let an = document.querySelector('#a_n');
        let an_x = an.getBoundingClientRect().x;
        let kherson = document.querySelector('#kherson');

```

```

        let kherson_x1 = kherson.getBoundingClientRect().x;
        let kherson_width =
kherson.getBoundingClientRect().width;
        let kherson_x2 = kherson_x1 + kherson_width;
        if (an_x >= kherson_x1 && an_x <= kherson_x2) {
            mark = mark + 1;
        }
        else {
            document.querySelector('#kherson').style.border =
"2px solid darkred";
        }
        let mc = document.querySelector('#m_c');
        let mc_x = mc.getBoundingClientRect().x;
        let sumy = document.querySelector('#sumy');
        let sumy_x1 = sumy.getBoundingClientRect().x;
        let sumy_width = sumy.getBoundingClientRect().width;
        let sumy_x2 = sumy_x1 + sumy_width;
        if (mc_x >= sumy_x1 && mc_x <= sumy_x2) {
            mark = mark + 1;
        }
        else {
            document.querySelector('#sumy').style.border = "2px
solid darkred";
        }
        let medobory = document.querySelector('#medobory');
        let medobory_x = medobory.getBoundingClientRect().x;
        let ternopil = document.querySelector('#ternopil');
        let ternopil_x1 = ternopil.getBoundingClientRect().x;
        let ternopil_width =
ternopil.getBoundingClientRect().width;
        let ternopil_x2 = ternopil_x1 + ternopil_width;
        if (medobory_x >= ternopil_x1 && medobory_x <=
ternopil_x2) {
            mark = mark + 1;
        }
        else {
            document.querySelector('#ternopil').style.border =
"2px solid darkred";
        }
        let gl = document.querySelector('#g_1');
        let gl_x = gl.getBoundingClientRect().x;
        let kharkiv = document.querySelector('#kharkiv');
        //верхний левый угол первой колонки
        let kharkiv_x1 = kharkiv.getBoundingClientRect().x;
        //ширина первой колонки
        let kharkiv_width =
kharkiv.getBoundingClientRect().width;
        //верхний правый угол первой колонки
        let kharkiv_x2 = kharkiv_x1 + kharkiv_width;
        if (gl_x >= kharkiv_x1 && gl_x <= kharkiv_x2) {
            mark = mark + 1;
        }
    }

```

```
        else {
            document.querySelector('#kharkiv').style.border =
"2px solid darkred";
        }
        let drev = document.querySelector('#drev');
        let drev_x = drev.getBoundingClientRect().x;
        let zhytomyr = document.querySelector('#zhytomyr');
        let zhytomyr_x1 = zhytomyr.getBoundingClientRect().x;
        let zhytomyr_width =
zhytomyr.getBoundingClientRect().width;
        let zhytomyr_x2 = zhytomyr_x1 + zhytomyr_width;
        if (drev_x >= zhytomyr_x1 && drev_x <= zhytomyr_x2) {
            mark = mark + 1;
        }
        else {
            document.querySelector('#zhytomyr').style.border =
"2px solid darkred";
        }
    };

</script>
```

## Додаток Е

У даному додатку наведений код п'ятого завдання

```
<div class="cross">
  <table border="1" align="center" class="cross">
    <tr>
      <td id="td1"></td>
      <td id="td2"></td>
      <td id="td3"></td>
      <td id="td0"><b>1 </b></td>
      <td><b><input id="input1" type="text" size="1"
maxlength="1" tabindex="1"></b></td>
      <td><b><input id="input2" type="text" size="1"
maxlength="1" tabindex="2"></b></td>
      <td><b><input id="input3" type="text" size="1"
maxlength="1" tabindex="3"></b></td>
      <td id="td4"><b><input id="input4" type="text" size="1"
maxlength="1" tabindex="4"></b></td>
      <td><b><input id="input5" type="text" size="1"
maxlength="1" tabindex="5"></b></td>
      <td><b><input id="input6" type="text" size="1"
maxlength="1" tabindex="6"></b></td>
      <td><b><input id="input7" type="text" size="1"
maxlength="1" tabindex="7"></b></td>
    </tr>
    <tr>
      <td id="td5"></td>
      <td id="td6"></td>
      <td id="td0"><b>2 </b></td>
      <td><b><input id="input8" type="text" size="1"
maxlength="1" tabindex="8"></b></td>
      <td><b><input id="input9" type="text" size="1"
maxlength="1" tabindex="9"></b></td>
      <td><b><input id="input10" type="text" size="1"
maxlength="1" tabindex="10"></b></td>
      <td><b><input id="input11" type="text" size="1"
maxlength="1" tabindex="11"></b></td>
      <td id="td7"><b><input id="input12" type="text" size="1"
maxlength="1" tabindex="12"></b></td>
      <td><b><input id="input13" type="text" size="1"
maxlength="1" tabindex="13"></b></td>
      <td><b><input id="input14" type="text" size="1"
maxlength="1" tabindex="14"></b></td>
      <td><b><input id="input15" type="text" size="1"
maxlength="1" tabindex="15"></b></td>
    </tr>
    <tr>
      <td id="td01"><b>3 </b></td>
      <td><b><input id="input16" type="text" size="1"
maxlength="1" tabindex="16"></b></td>
```

```

        <td><b><input id="input17" type="text" size="1"
maxlength="1" tabindex="17"></b></td>
        <td><b><input id="input18" type="text" size="1"
maxlength="1" tabindex="18"></b></td>
        <td><b><input id="input19" type="text" size="1"
maxlength="1" tabindex="19"></b></td>
        <td><b><input id="input20" type="text" size="1"
maxlength="1" tabindex="20"></b></td>
        <td><b><input id="input21" type="text" size="1"
maxlength="1" tabindex="21"></b></td>
        <td id="td8"><b><input id="input22" type="text" size="1"
maxlength="1" tabindex="22"></b></td>
        <td><b><input id="input23" type="text" size="1"
maxlength="1" tabindex="23"></b></td>
        <td><b><input id="input24" type="text" size="1"
maxlength="1" tabindex="24"></b></td>
    </tr>
    <tr>
        <td id="td9"></td>
        <td id="td10"></td>
        <td id="td11"></td>
        <td id="td12"></td>
        <td id="td03"><b>4 </b></td>
        <td><b><input id="input25" type="text" size="1"
maxlength="1" tabindex="25"></b></td>
        <td><b><input id="input26" type="text" size="1"
maxlength="1" tabindex="26"></b></td>
        <td id="td13"><b><input id="input27" type="text"
size="1" maxlength="1" tabindex="27"></b></td>
        <td><b><input id="input28" type="text" size="1"
maxlength="1" tabindex="28"></b></td>
        <td><b><input id="input29" type="text" size="1"
maxlength="1" tabindex="29"></b></td>
        <td><b><input id="input30" type="text" size="1"
maxlength="1" tabindex="30"></b></td>
        <td><b><input id="input31" type="text" size="1"
maxlength="1" tabindex="31"></b></td>
        <td><b><input id="input32" type="text" size="1"
maxlength="1" tabindex="32"></b></td>
        <td><b><input id="input33" type="text" size="1"
maxlength="1" tabindex="33"></b></td>
    </tr>
    <tr>
        <td id="td14"></td>
        <td id="td15"></td>
        <td id="td16"></td>
        <td id="td17"></td>
        <td id="td18"></td>
        <td id="td0"><b>5 </b></td>
        <td><b><input id="input34" type="text" size="1"
maxlength="1" tabindex="34"></b></td>

```

```

        <td id="td19"><b><input id="input35" type="text"
size="1" maxlength="1" tabindex="35"></b></td>
        <td><b><input id="input36" type="text" size="1"
maxlength="1" tabindex="36"></b></td>
        <td><b><input id="input37" type="text" size="1"
maxlength="1" tabindex="37"></b></td>
        <td><b><input id="input38" type="text" size="1"
maxlength="1" tabindex="38"></b></td>
        <td><b><input id="input39" type="text" size="1"
maxlength="1" tabindex="39"></b></td>
        <td><b><input id="input40" type="text" size="1"
maxlength="1" tabindex="40"></b></td>
        <td><b><input id="input41" type="text" size="1"
maxlength="1" tabindex="41"></b></td>
        <td><b><input id="input42" type="text" size="1"
maxlength="1" tabindex="42"></b></td>
    </tr>
    <tr>
        <td id="td20"></td>
        <td id="td21"></td>
        <td id="td22"></td>
        <td id="td02"><b>6 </b></td>
        <td><b><input id="input43" type="text" size="1"
maxlength="1" tabindex="43"></b></td>
        <td><b><input id="input44" type="text" size="1"
maxlength="1" tabindex="44"></b></td>
        <td><b><input id="input45" type="text" size="1"
maxlength="1" tabindex="45"></b></td>
        <td id="td23"><b><input id="input46" type="text"
size="1" maxlength="1" tabindex="46"></b></td>
        <td><b><input id="input47" type="text" size="1"
maxlength="1" tabindex="47"></b></td>
        <td><b><input id="input48" type="text" size="1"
maxlength="1" tabindex="48"></b></td>
        <td><b><input id="input49" type="text" size="1"
maxlength="1" tabindex="49"></b></td>
        <td><b><input id="input50" type="text" size="1"
maxlength="1" tabindex="50"></b></td>
    </tr>
</table>
</div>
<div class="panel panel-success" id="questions">
    <div class="panel-heading">ПИТАННЯ</div>
    <div class="panel-body">
        <a href="#" data-toggle="popover" data-content="Усі форми
співіснування різних видів.">Питання 1</a><br>
        <a href="#" data-toggle="popover" data-
content="Угруповання популяцій організмів різних видів, які
населяють ділянку біосфери з однорідними умовами існування і
пов'язані між собою різноманітними взаємозв'язками.">Питання
2</a><br>

```

[Питання 3](#)<br>

[Питання 4](#)<br>

[Питання 5](#)<br>

[Питання 6](#)</div>

</div>

</div>

<button id="btn" type="button" class="btn btn-success">Дати відповідь</button>

<script>

```
var btn = document.querySelector('#btn');
```

```
document.querySelector('#btn').onclick = function() {
```

```
  document.getElementById("btn").disabled = true;
```

```
  document.getElementById("btn2").disabled = false;
```

```
  let mark = 0;
```

```
  if (document.getElementById("input1").value == 'с' &&
```

```
    document.getElementById("input2").value == 'и' &&
```

```
    document.getElementById("input3").value == 'м' &&
```

```
    document.getElementById("input4").value == 'б' &&
```

```
    document.getElementById("input5").value == 'і' &&
```

```
    document.getElementById("input6").value == 'о' &&
```

```
    document.getElementById("input7").value == 'з') {
```

```
      mark = mark + 1;
```

```
  }
```

```
  else{
```

```
    document.getElementById("input1").style.color = 'darkred';
```

```
    document.getElementById("input2").style.color = 'darkred';
```

```
    document.getElementById("input3").style.color = 'darkred';
```

```
    document.getElementById("input4").style.color = 'darkred';
```

```
    document.getElementById("input5").style.color = 'darkred';
```

```
    document.getElementById("input6").style.color = 'darkred';
```

```
    document.getElementById("input7").style.color = 'darkred';
```

```
  }
```

```

        if (document.getElementById("input8").value == 'б' &&
            document.getElementById("input9").value == 'и' &&
            document.getElementById("input10").value == 'о' &&
            document.getElementById("input11").value == 'ц' &&
            document.getElementById("input12").value == 'е' &&
            document.getElementById("input13").value == 'н' &&
            document.getElementById("input14").value == 'о' &&
            document.getElementById("input15").value == 'з') {
            mark = mark + 1;
        }
        else{
            document.getElementById("input8").style.color =
'darkred';
            document.getElementById("input9").style.color =
'darkred';
            document.getElementById("input10").style.color =
'darkred';
            document.getElementById("input11").style.color =
'darkred';
            document.getElementById("input12").style.color =
'darkred';
            document.getElementById("input13").style.color =
'darkred';
            document.getElementById("input14").style.color =
'darkred';
            document.getElementById("input15").style.color =
'darkred';
        }
        if (document.getElementById("input16").value == 'р' &&
            document.getElementById("input17").value == 'е' &&
            document.getElementById("input18").value == 'д' &&
            document.getElementById("input19").value == 'у' &&
            document.getElementById("input20").value == 'ц' &&
            document.getElementById("input21").value == 'е' &&
            document.getElementById("input22").value == 'н' &&
            document.getElementById("input23").value == 'т' &&
            document.getElementById("input24").value == 'и') {
            mark = mark + 1;
        }
        else{
            document.getElementById("input16").style.color =
'darkred';
            document.getElementById("input17").style.color =
'darkred';
            document.getElementById("input18").style.color =
'darkred';
            document.getElementById("input19").style.color =
'darkred';
            document.getElementById("input20").style.color =
'darkred';
            document.getElementById("input21").style.color =
'darkred';
        }
    
```



```

        document.getElementById("input22").style.color =
'darkred';
        document.getElementById("input23").style.color =
'darkred';
        document.getElementById("input24").style.color =
'darkred';
    }
    if (document.getElementById("input25").value == 'Ф' &&
        document.getElementById("input26").value == 'И' &&
        document.getElementById("input27").value == 'Т' &&
        document.getElementById("input28").value == 'О' &&
        document.getElementById("input29").value == 'Ф' &&
        document.getElementById("input30").value == 'А' &&
        document.getElementById("input31").value == 'Г' &&
        document.getElementById("input32").value == 'И' &&
        document.getElementById("input33").value == 'Я') {
        mark = mark + 1;
        console.log(mark);
    }
    else{
'darkred';
        document.getElementById("input25").style.color =
'darkred';
        document.getElementById("input26").style.color =
'darkred';
        document.getElementById("input27").style.color =
'darkred';
        document.getElementById("input28").style.color =
'darkred';
        document.getElementById("input29").style.color =
'darkred';
        document.getElementById("input30").style.color =
'darkred';
        document.getElementById("input31").style.color =
'darkred';
        document.getElementById("input32").style.color =
'darkred';
        document.getElementById("input33").style.color =
'darkred';
    }
    if (document.getElementById("input34").value == 'П' &&
        document.getElementById("input35").value == 'О' &&
        document.getElementById("input36").value == 'П' &&
        document.getElementById("input37").value == 'У' &&
        document.getElementById("input38").value == 'Л' &&
        document.getElementById("input39").value == 'Я' &&
        document.getElementById("input40").value == 'Ц' &&
        document.getElementById("input41").value == 'И' &&
        document.getElementById("input42").value == 'Я') {
        mark = mark + 1;
    }
    else{

```

```

        document.getElementById("input34").style.color =
'darkred';
        document.getElementById("input35").style.color =
'darkred';
        document.getElementById("input36").style.color =
'darkred';
        document.getElementById("input37").style.color =
'darkred';
        document.getElementById("input38").style.color =
'darkred';
        document.getElementById("input39").style.color =
'darkred';
        document.getElementById("input40").style.color =
'darkred';
        document.getElementById("input41").style.color =
'darkred';
        document.getElementById("input42").style.color =
'darkred';
    }
    if (document.getElementById("input43").value == 'H' &&
        document.getElementById("input44").value == 'O' &&
        document.getElementById("input45").value == 'O' &&
        document.getElementById("input46").value == 'C' &&
        document.getElementById("input47").value == 'Φ' &&
        document.getElementById("input48").value == 'e' &&
        document.getElementById("input49").value == 'p' &&
        document.getElementById("input50").value == 'a') {
        mark = mark + 1;
    }
    else{
'darkred';
        document.getElementById("input43").style.color =
'darkred';
        document.getElementById("input44").style.color =
'darkred';
        document.getElementById("input45").style.color =
'darkred';
        document.getElementById("input46").style.color =
'darkred';
        document.getElementById("input47").style.color =
'darkred';
        document.getElementById("input48").style.color =
'darkred';
        document.getElementById("input49").style.color =
'darkred';
        document.getElementById("input50").style.color =
'darkred';
    } };
</script>

```