

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК

ВИПУСКНА РОБОТА

на тему:

**“Веб-додаток для автоматизації тестування в
середніх навчальних закладах”**

**Завідувач
випускаючої кафедри**

Довбиш А.С.

Керівник роботи

Проценко О.Б.

Студентка гр. ІН–61

Онічко В.А.

СУМИ 2020

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ

Кафедра комп'ютерних наук

Затверджую _____

Зав. кафедрою Довбиш А.С.

“ _____ ” _____ 2020 р.

Завдання

до випускної роботи

Студентки четвертого курсу, групи ІН-61 спеціальності “Інформатика”
денної форми навчання Онічко Вікторії Анатоліївни.

**Тема: “Веб-додаток для автоматизації тестування в середніх навчальних
закладах”**

Затверджена наказом по СумДУ

№ _____ від _____ 2020 р.

Зміст пояснювальної записки: 1) аналітичний огляд актуальності розробки додатку, огляд існуючих веб-додатків; 2) формування завдань для дослідження та постановка задачі; 3) огляд і вибір методів та засобів розробки; 4) розробка веб-додатку для автоматизації тестування; 5) аналіз отриманих результатів.

Дата видачі завдання “ _____ ” _____ 2020 г.

Керівник випускної роботи _____ Проценко О.Б.

Завдання прийняла до виконання _____ Онічко В.А.

РЕФЕРАТ

Записка: 83 стор., 30 рис., 5 додатків, 16 джерел.

Об'єкт дослідження — Веб-додаток для автоматизації тестування в середніх навчальних закладах

Мета роботи — розробити веб-додаток для автоматизації тестування з біології в середніх навчальних закладах. Додаток повинен бути зручним, інтуїтивно зрозумілим, дружнім до учня, а також мати елементи інтерактивності.

Результати — виконаний аналіз актуальності розробки додатку, вибрані засоби вирішення задачі. Додаток розроблений за допомогою фреймворку Yii з вбудованим фронтенд-фреймворком Bootstrap.

ВЕБ-ДОДАТОК, ТЕСТУВАННЯ УЧНІВ, ФРЕЙМВОРК, HTML, CSS,
JAVASCRIPT, PHP, YII, BOOTSTRAP

ЗМІСТ

ВСТУП.....	6
1 ІНФОРМАЦІЙНИЙ ОГЛЯД	7
1.1 ІТ технології в освітній галузі України	7
1.2 Використання концепції MVC у веб-додатках.....	8
1.3 Огляд веб-додатків для автоматизованого тестування	11
1.4 Постановка задачі	17
2 ВИБІР МЕТОДІВ ВИРІШЕННЯ ЗАДАЧІ	18
2.1 Вибір методів розробки	18
2.2 Вибір засобів програмування	19
3 ПРОГРАМНА РЕАЛІЗАЦІЯ	29
3.1 Проектування структури сайту	29
3.2 Реалізації основних компонентів	31
ВИСНОВКИ	51
СПИСОК ЛІТЕРАТУРИ.....	52
ДОДАТКИ	54

ВСТУП

З кожним днем залишається все менше галузей, які б не користувалися перевагами комп'ютерних технологій. Зокрема, вони є одним з основних пріоритетів у розвитку сучасної освітньої галузі. Засоби навчання, засновані на комп'ютерних технологіях, забезпечують вищий рівень інформованості людини, а значить, її освіченість. Крім цього, вони можуть значно спростити процес викладання, адміністрування, управління. Прикладом впровадження комп'ютерних технологій в навчальні процеси може слугувати використання презентацій, які вже давно набули масового поширення в навчальних закладах різних рівнів. Вони дозволяють поєднувати в собі різні елементи: анімації, відеофрагменти, аудіозаписи, тестування, що робить викладання матеріалу більш цікавим та різнобічним, а також підвищує рівень його засвоєння. Популярністю також користуються додатки з розкладом предметів для учнів, студентів та викладачів. Зараз чи не кожен має смартфон, який завжди під рукою, тому такі додатки чудово заміняють паперові щоденники, які в багатьох ситуаціях є зовсім незручними.

Набагато менш популярними та використовуваними в нашій країні є додатки, які дозволяють проводити тестування учнів середніх навчальних закладів. Наявні програми є застарілими, а більшість вчителів проводять тестування в письмовому вигляді, що не завжди доречно. При вивченні таких предметів природничого циклу, як біологія, географія, хімія часто застосовуються схеми, рисунки, карти, таблиці. Іноді вони бувають досить громіздкими, що ускладнює їх друкування для проведення тестування. В таких ситуаціях дуже зручно було б користуватися електронними засобами, які б дозволяли проводити тестування не лише у вигляді питань з варіантами відповідей, а й шляхом заповнення таблиць, доповнення схем та рисунків, нанесення об'єктів на карти. Це може значно полегшити оцінювання знань учителями, а для учнів зробити процес тестування цікавим та пізнавальним.

1 ІНФОРМАЦІЙНИЙ ОГЛЯД

1.1 ІТ технології в освітній галузі України

Інформаційні технології з моменту їх появи активно використовувалися у сфері освіти в багатьох розвинених країнах. В Україні лише останнім часом досить швидкими темпами відбувається оснащення шкіл комп'ютерною технікою. Залишається все менше навчальних закладів, у яких було б повністю відсутнє мультимедійне обладнання, комп'ютерні класи, планшети. На фоні різкого зростання забезпеченості комп'ютерною технікою учителів та учнів більш чітко вимальовується проблема ефективності використання наявного обладнання.

Традиційно виділяються такі напрямки використання ІТ в освітній галузі:

- вивчення власне комп'ютерів;
- навчання та розвиток учнів;
- використання в управлінні освітніми закладами;
- підвищення ефективності науково-педагогічних досліджень завдяки комп'ютерам;
- спілкування в мережі Інтернет;
- підвищення кваліфікації вчителів.

На сьогоднішній день майже немає програмного забезпечення, яке б активно використовувалося на уроках для навчання і перевірки знань учнів та було при цьому затверджено на державному рівні. Наявні програми є досить примітивними та вузько спеціалізованими, але навіть вони застосовуються дуже рідко. Найбільш розповсюдженим способом використання комп'ютерного та мультимедійного обладнання є демонстрація презентацій або відеофрагментів на задану тему. Хоча комп'ютерна техніка у школах дуже часто є застарілою, та навіть її можливості використовуються далеко не в повному обсязі.

Є багато методів використання інформаційних технологій в освітній галузі, та, враховуючи реалії нашої держави, найбільш простим для впровадження є використання веб-технологій. Основною перевагою додатків, заснованих на веб-сайтах, є можливість використання на будь-яких моделях комп'ютерів, адже такий додаток залежить тільки від наявності веб-браузера. Веб-додатки при цьому можуть бути як вузько спеціалізованими (забезпечувати одну чи декілька функцій, наприклад, тестування з певного предмету), так і більш широко (наприклад, зберігати оцінки учнів, частково або повністю замінюючи паперові шкільні журнали). В майбутньому функціонал таких веб-додатків може бути розширений до використання не тільки вчителями та учнями, а ще й батьками, та брати участь у більшій кількості етапів навчального процесу.

1.2 Використання концепції MVC у веб-додатках

Веб-додаток – прикладна програма з архітектурою «клієнт-сервер». Веб-додаток працює на стороні веб-сервера, а в якості клієнта використовує веб-браузер. Веб-браузери взаємодіють з веб-серверами за допомогою протоколу передачі гіпертексту (HTTP).

Код клієнтської частини запускається у веб-браузері. В першу чергу він відповідає за зовнішній вигляд та поведінку відображуваної веб-сторінки. Зазвичай такий код написаний за допомогою HTML, CSS та JavaScript.

Код на стороні сервера в основному виконує такі задачі, як перевірка даних і запитів, які були відправлені клієнтом, використання баз даних, відправка необхідної інформації клієнту. Такий код може бути написаний великою кількістю мов програмування, найпопулярнішими серед яких є PHP, Java, JavaScript, C# тощо.

Для прискорення та спрощення розробки як клієнтської, так і серверної частин дуже часто використовують веб-фреймворки. Оскільки ці частини є

дуже різними, то і фреймворки для них теж різні. Для клієнтської частини існують фронтенд-фреймворки, які допомагають виключно при верстці. Вони не пов'язані із логікою додатку та працюють у браузері. Водночас існують бекенд-фреймворки, які можуть стати у нагоді при роботі із серверною частиною. Вони зазвичай містять уже готову реалізацію стандартних функцій, таких як аутентифікація, організація доступу до бази даних тощо. Крім цього, фреймворки також визначають спосіб побудови структури додатку, взаємодію частин його структури.

Однією з найбільш популярних та широко використовуваних концепцій є MVC – Model-View-Controller. Вона підтримується багатьма веб-фреймворками. Прикладами можуть слугувати Angular, Yii, Zend, Django, Laravel тощо. В основі цього конструкційного шаблону лежить чіткий розподіл відповідальності за окремі функції між трьома компонентами, які постійно взаємодіють та є неподільними. Реалізації даного шаблону можуть дещо відрізнитися в залежності від потреб, але основні принципи є незмінними.

Компонент Controller відповідає за організаційну логіку додатку та не містить SQL-запитів або HTML-розмітку. Він виконує обробку запитів користувача та виклик відповідних ресурсів. При натисканні користувачем на елементи інтерфейсу задаються певні дії. Саме Controller координує необхідні ресурси та об'єкти для виконання цих дій. Він викликає відповідні Model та View. Веб-додаток не завжди містить тільки один Controller. Їх може бути декілька, але обов'язково є один головний, який викликає всі інші в залежності від ситуації.

Компонент Model відповідає за бізнес-логіку, її правила та функції. В ньому міститься опис даних, відбувається їх обробка, виконуються дії із базою даних (запис, видалення, перегляд). Сам Model не містить дані, а отримує їх із бази, після чого надає компоненту Controller необхідні дані, на які надіслав запит користувач, але сам при цьому безпосередньо з

користувачем не взаємодіє. Модель даних завжди однакова, відрізняється лише її представлення. Компонент Model не залежить від View та Controller, але View та Controller залежать від Model.

Компонент View відповідає за презентаційну логіку, описує зовнішній вигляд додатку. Він не взаємодіє із базою даних безпосередньо, а отримує дані від Model та забезпечує їх представлення одним із способів, може бути шаблоном, який заповнюється даними. В більшості випадків View декілька. Необхідний View обирає Controller в залежності від поточної ситуації. Хоча View відповідає за представлення даних, він містить не лише HTML-код. Зазвичай додається простий PHP-код, адже HTML є лише статичною розміткою документа й не має засобів для роботи з даними, отриманими з Model.

Головною перевагою використання концепції є чіткий розподіл логіки візуалізації та логіки власне додатку. Код додатків стає більш структурованим, що полегшує сприйняття, тестування та підтримку. Оскільки кожний компонент виконує свою фіксовану роль, у випадку виникнення проблеми локалізація відбувається швидше. Крім цього, з'являється можливість модифікації одного з компонентів без істотного впливу на інші. Додатки, розроблені з використанням даної концепції, стають більш стійкими до злому.

Недоліком концепції можна назвати складність освоєння. На етапі вивчення доводиться докладати певні зусилля, щоб розібратися, за що відповідає кожний з компонентів. Крім цього, трапляються випадки, коли одночасно потрібно виконати дії над декількома Model. У таких ситуаціях з'являється Controller, який містить операції не лише над Model, до якого він належить, але й над іншими Model. Це створює небажані залежності – між Model та декількома View і Controller, між View та Controller і однією Model.

Концепція MVC – це фундаментальний шаблон, найпоширенішими модифікаціями якого є MVVM (Model-View-ViewModel), MVP (Model-View-

Presenter) та HMVC (Hierarchical MVC).

У концепції MVP замість компонента Controller присутній Presenter. Головними відмінностями MVP від MVC є те, що зв'язок між Model та View більш слабкий, за прив'язку Model до View відповідає Presenter. В MVC Controller визначає, який View викликається у відповідь на певну дію користувача. При цьому в MVP дії направляються через View в Presenter. Це означає, що в MVP, на відміну від MVC, View містить певну логіку.

В концепції MVVM немає класичного Controller, але використовується додаткова модель ViewModel, яка слугує сполучною ланкою між Model та View і не містить ніякої бізнес-логіки. Оскільки цей компонент не взаємодіє із View безпосередньо, він представляє властивості і методи, до яких може прив'язуватися View з метою отримання інформації із ViewModel і виклику із нього методів. Сам ViewModel отримує дані із Model.

Характерною рисою використання HMVC є наявність MVC триад, кожна з яких функціонує незалежно від інших, але може виконати запит через Controller. Такі ієрархії вибудовуються окремо для Model, View і Controller та мають прямі посилання на свої дочірні компоненти.

1.3 Огляд веб-додатків для автоматизованого тестування

На сьогодні не існує подібних веб-додатків, які є в загальному користуванні. Всі додатки, які вдалося знайти та протестувати, дають можливість працювати лише з тестами у вигляді «питання – декілька варіантів відповідей». Вони містять схеми та рисунки, але статичні, тобто немає можливості самостійно складати схему чи доповнювати рисунок. Також є велика кількість веб-додатків для підготовки до ЗНО, що також передбачає роботу тільки з питаннями і відповідями до них.

naurok.com.ua – безкоштовний веб-додаток. Простий у використанні,

містить тести для всіх класів з 1 по 11 за темами. Дає можливість учителям самостійно створювати та додавати тести. Додаток має приємний дизайн. Після кожної відповіді учень має можливість побачити результат – з'являється приємна мінімалістична анімація. Після завершення тестування учневі надається результат за 12-бальною системою, а також у відсотках правильних відповідей. Основними недоліками є те, що naurok.com.ua підтримує роботу тільки з питаннями з одним або декількома варіантами відповідей. Є можливість додавання фотографій, але не передбачена робота зі схемами та питаннями відкритого характеру (відповіддю є одне або декілька слів). Приклад роботи даного додатку наведений на Рисунку 1.1 - Приклад роботи веб-додатку naurok.com.ua.

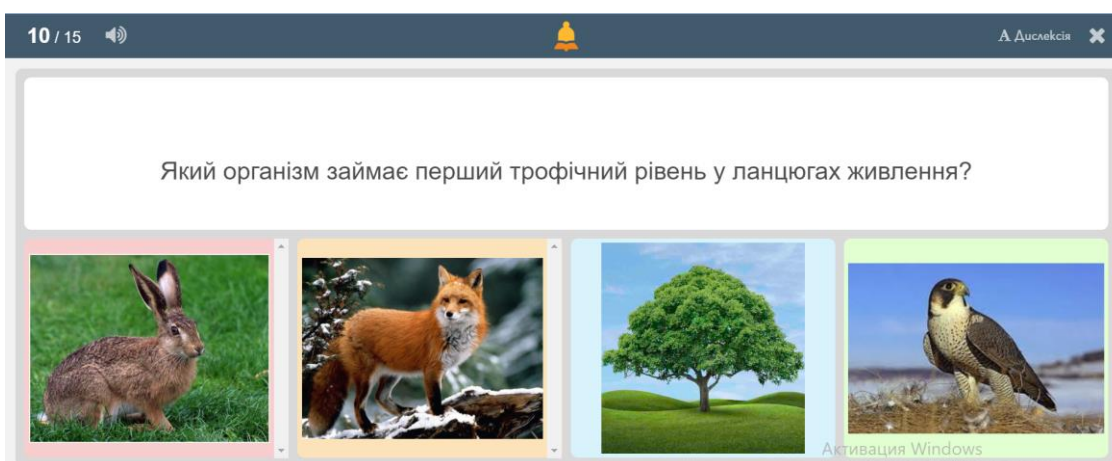


Рисунок 1.1 – Приклад роботи веб-додатку naurok.com.ua

ukrerudyt.com – веб-додаток з тестами з біології для різних класів. Має досить обмежений набір тестів без можливості для вчителів додавати власні. Крім цього, тести наявні тільки починаючи з 8 класу. Основним недоліком даного додатку є те, що питання передбачають тільки один варіант відповіді, немає питань відкритого характеру. Також відсутні будь-які фотографії, рисунки, схеми. Учень може дізнатися свої результати тільки після проходження тестування. В питаннях, на які було дано неправильну

відповідь, не відображається правильна відповідь, тобто немає можливості «роботи над помилками».

Приклад роботи даного веб-додатку наведений на Рисунку 1.2 - Приклад роботи веб-додатку ukrerudyt.com

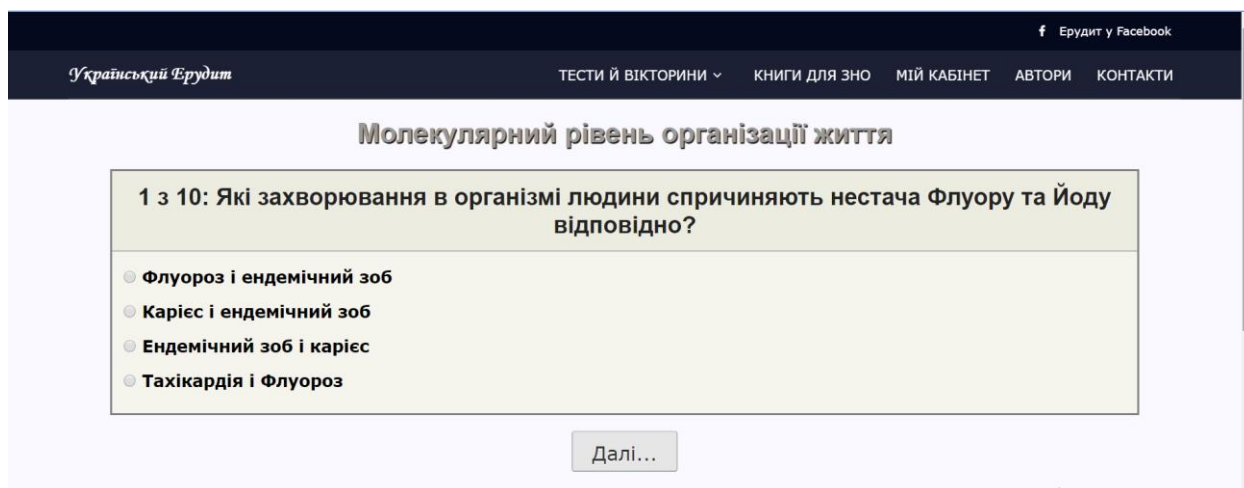


Рисунок 1.2 – Приклад роботи веб-додатку ukrerudyt.com

onlinetestpad.com – веб-додаток з декількома тестами лише для 10 та 11 класів. Відсутній розподіл на теми, тому кожен із тестів складається із занадто великої кількості питань. Майже кожне питання містить ілюстрацію, але, як і в попередніх додатках, немає можливості доповнювати схеми чи таблиці. Дизайн додатку є досить непривабливим, немає можливості відразу побачити, чи правильною є відповідь на питання. Приклад роботи даного веб-додатку наведений на Рисунку 1.3 - Приклад роботи веб-додатку onlinetestpad.com.

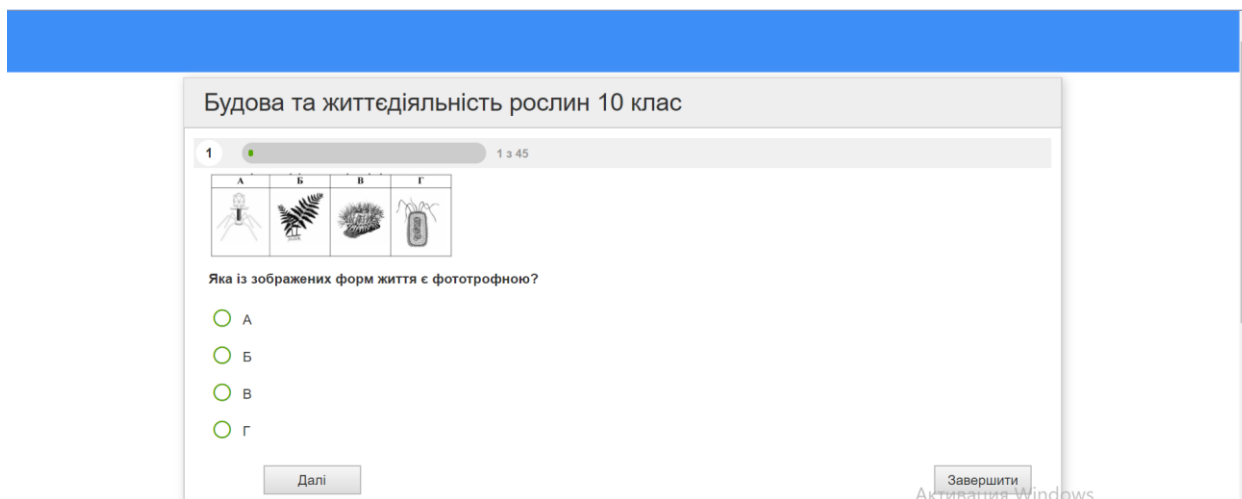


Рисунок 1.3 – Приклад роботи веб-додатку onlinetestpad.com

distance.edu.vn.ua – веб-додаток із дуже спрощеним дизайном та функціоналом. Немає можливості додавання своїх тестів. Перевірка відповіді на коректність відбувається тільки в кінці тестування, причому не відображається кількість правильних та неправильних – це можна дізнатися тільки порахувавши кількість «+» та «-», які проставляються біля кожного питання після завершення тестування. Відсутня можливість дізнатися правильну відповідь у випадку, якщо відповів неправильно. Приклад роботи даного веб-додатку наведений на Рисунку 1.4 - Приклад роботи веб-додатку distance.edu.vn.ua.



Рисунок 1.4 – Приклад роботи веб-додатку distance.edu.vn.ua.

testy-bio.pp.ua – веб-додаток зі спрощеним дизайном, але досить приємною анімацією. Безсумнівною перевагою даного додатку є можливість після кожного питання дізнатися, чи правильна була дана відповідь, а в кінці тесту побачити свій результат у 12-бальній системі та у відсотках. Більшість питань мають формат «питання з однією правильною відповіддю», «питання із декількома правильними відповідями», «питання із розгорнутою відповіддю» (декілька слів). Зустрічаються й інтерактивні питання, проте вони обмежуються перетягуванням об'єктів. Приклад роботи веб-додатку testy-bio.pp.ua наведений на Рисунку 1.15 – Приклад роботи веб-додатку testy-bio.pp.ua

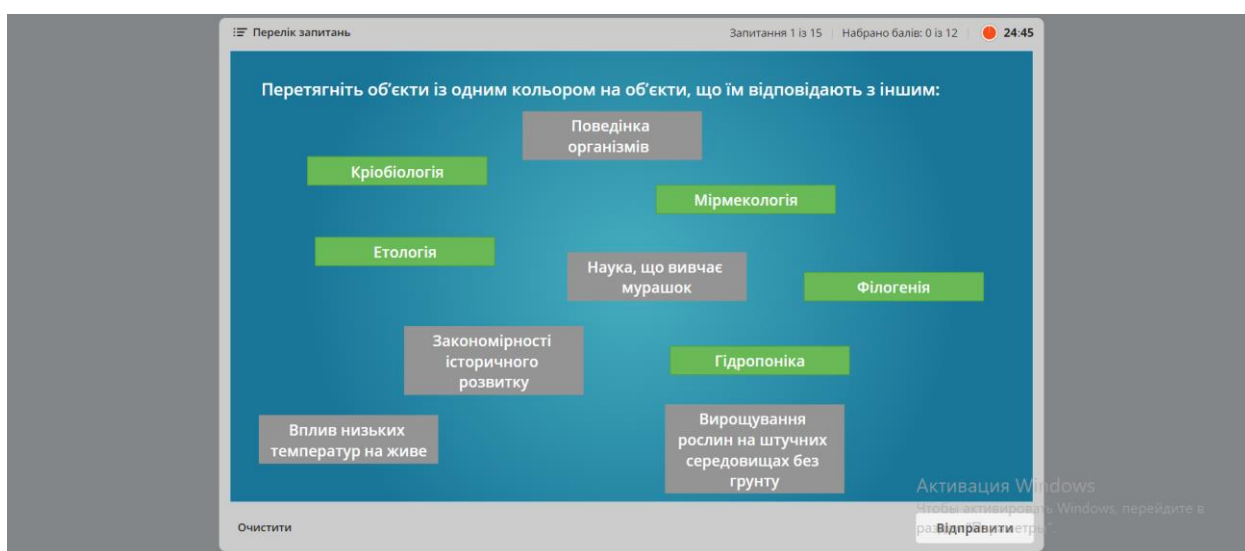


Рисунок 1.5 – Приклад роботи веб-додатку testy-bio.pp.ua.

testorium.net – веб-додаток із приємним оригінальним дизайном. У питаннях наявні ілюстрації, що робить тестування більш цікавим. Присутня оригінальна функція візуального відображення, на які питання була дана відповідь, а які були пропущені (з обох боків від питання відображений список усіх питань. Питання, на які була дана відповідь виділені жовтим, а ті, які залишилися без відповіді, - білим), проте свій результат можна дізнатися лише після закінчення тестування. Інтерактивні питання не передбачені

взагалі. Приклад роботи веб-додатку testorium.net наведений на Рисунку 1.6 –
 Приклад роботи веб-додатку testorium.net.

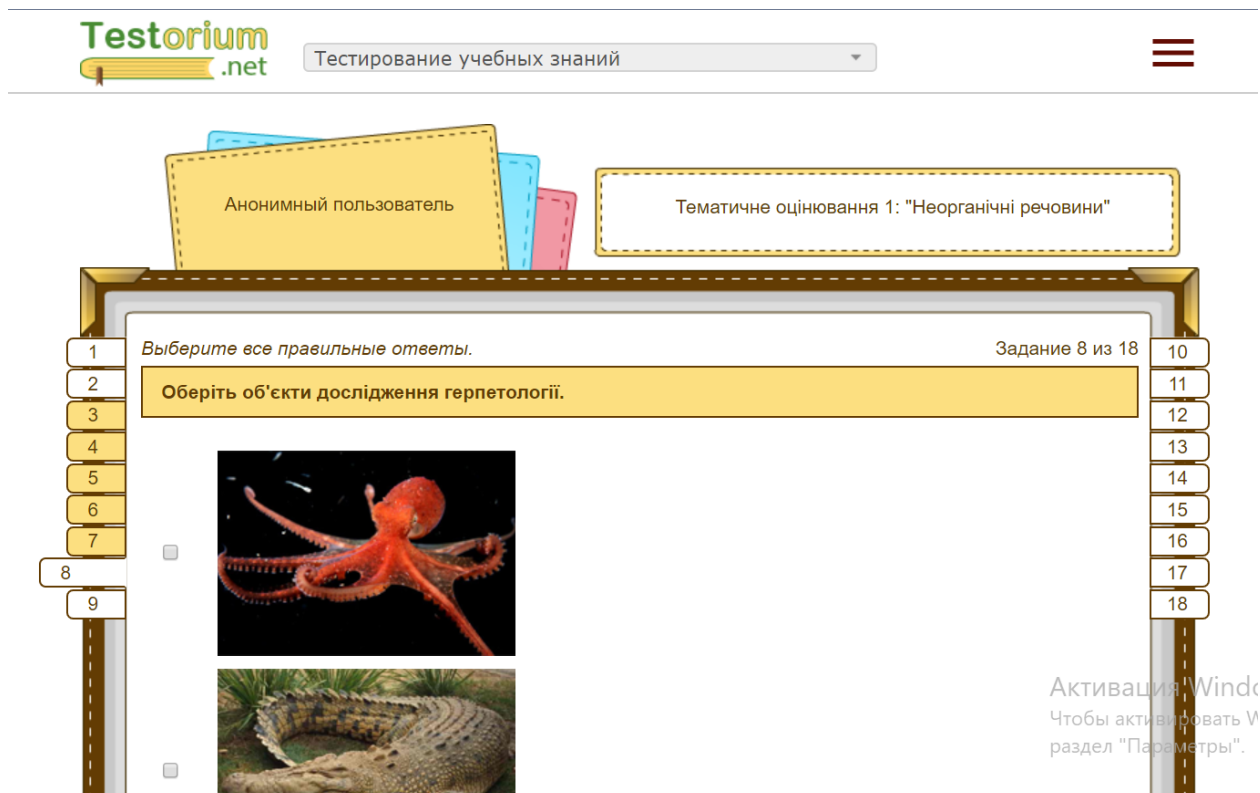


Рисунок 1.6 – Приклад роботи веб-додатку testorium.net.

Отже, наразі є актуальним створення саме веб-додатку, який, крім стандартних (можливість проходження тестів у вигляді «питання – одна відповідь» або «питання – декілька відповідей», відображення кількості правильних відповідей, відображення правильних відповідей у випадку неправильної відповіді), буде реалізовувати інші корисні функції, а саме:

- збереження результатів усіх пройдених тестів та відображення у відсотках і 12-бальній системі;
- відображення журналу з оцінками учнів класу для вчителів;
- робота зі схемами, таблицями (їх доповнення, редагування);
- розподіл тестів за темами та класами;
- забезпечення цікавого та пізнавального тестування (наприклад, шляхом додавання підбадьорливих написів та малюнків).

1.4 Постановка задачі

Розробити веб-додаток для автоматизації тестування у середніх навчальних закладах, який виконує наступні функції:

- авторизація користувачів з однією із трьох ролей: учень, учитель або адміністратор;
- класифікація тестів за класами;
- перевірка правильності відповіді та відображення результату відразу після того, як була дана відповідь;
- автоматичний розрахунок оцінки за 12-бальною системою та у відсотках;
- відображення оцінок учнів за всі пройдені тести в особистому кабінеті (для учнів);
- перегляд журналу класу з оцінками за пройдені тести (для вчителів);
- перегляд/додавання/редагування/видалення учнів та вчителів (для адміністратора);
- можливість відкривати/закривати доступ до тесту (для вчителів) для забезпечення проходження тесту учнями тільки у відведений для цього час.

Результатом роботи має бути веб-додаток, який повинен мати простий та ненав'язливий дизайн, бути інтуїтивно зрозумілим та цікавим для школярів.

2 ВИБІР МЕТОДІВ ВИРІШЕННЯ ЗАДАЧІ

2.1 Вибір методів розробки

Розробка повністю самописного веб-сайту зазвичай вимагає багато зусиль та часу. В більшості випадків вони витрачатимуться на реалізацію функцій, які вже були неодноразово створені іншими розробниками, тому такі витрати не є обґрунтованими. Цю проблему допомагають обійти фреймворки. Вони надають основу, на яку можна опиратися при створенні власного сайту, адже містять велику кількість попередньо реалізованого функціоналу, який також можна розширювати, додаючи власний код. Це значно збільшує не лише швидкість розробки, а й продуктивність та надійність додатку. При прийнятті рішення, чи використовувати фреймворк, потрібно враховувати індивідуальні особливості майбутнього проекту, та все ж розглянемо основні переваги та недоліки використання фреймворків при розробці веб-додатків.

Оскільки в основі фреймворків лежать кращі практики розробки та проектування, зменшується вірогідність виникнення проблем та помилок, підвищується безпека та ефективність додатку. При розв'язанні типових задач розробнику не потрібно придумувати власне рішення, він може скористатися готовою якісною реалізацією, над якою працювали багато розробників і яка неодноразово була протестована. Бекенд-фреймворки містять готову реалізацію майже всіх основних функцій, таких як проектування бази даних, побудова модулів системи, створення адміністративної частини тощо, що значно скорочує використовувані ресурси. Фронтенд-фреймворки містять готові розмітку та стилі, на основі яких дуже легко розробити власний дизайн та зробити сайт адаптивним з мінімальною кількістю коду. Фреймворки зазвичай мають відкритий код для використання, тому завжди можуть бути доповнені для власних потреб. Крім

цього, додатки на базі фреймворків легше підтримувати. Структура таких додатків визначена фреймворком, тобто є стандартною. Тому розробникам, які не брали участь в процесі проектування та розробки, знадобиться набагато менше часу, щоб розібратися в архітектурі, зрозуміти принцип роботи додатку, знайти компонент, у якому реалізований той чи інший функціонал, особливо, якщо вони вже знайомі із основними принципами використовуваного фреймворку. Більшість фреймворків мають дуже розгорнуту документацію, в якій легко знайти необхідну інформацію. Та навіть якщо на проблему, що виникла, не вдалося знайти вичерпну відповідь, можна звернутися до спільноти розробників, які також використовують даний фреймворк, та отримати пораду.

Серед недоліків використання фреймворків – те, що освоєння деяких із них вимагає досить багато зусиль. Потрібно витратити чимало часу на вивчення та осмислення документації. Проте бувають ситуації, коли відмова від використання фреймворків є виправданою. Наприклад, для невеликих проектів доцільніше їх не використовувати взагалі або використовувати тільки фронтенд-фреймворки. Якщо додаток має нескладну логіку, то використання фреймворку навпаки може його ускладнити та збільшити об'єм коду, частина якого буде надлишковою.

2.2 Вибір засобів програмування

Веб-додатки складаються із двох частин: фронтенд (візуальна частина, яка працює в браузері) та бекенд (серверна частина). До серверної частини відносяться такі функції, як авторизація, зберігання та обробка даних. До клієнтської частини в основному відноситься розмітка сторінки, застосування стилів, тобто все, з чим безпосередньо взаємодіє користувач. Оскільки фронтенд та бекенд відповідають за різні функції, то мови програмування та фреймворки для них теж використовуються різні.

Найпоширенішими мовами для серверної частини є PHP, JavaScript, Java, Python, для клієнтської – JavaScript, а також мова розмітки HTML і стилі CSS. На сьогоднішній день існує багато фреймворків як для фронтенда, так і для бекенда. Розглянемо найвідоміші фронтенд-фреймворки.

Bootstrap – один із найвідоміших безкоштовних HTML, CSS і JavaScript фреймворків з великою кількістю компонентів для веб-додатків. Він надає реалізацію таких інструментів, як сітка, кнопки, форми, таблиці, модальні вікна, навігаційні елементи меню, пагінація, каруселі, спливаючі підказки, випадаючі списки тощо. Контент стилізується за допомогою класів, наявних у Bootstrap. Крім скорочення кількості часу та зусиль, витрачених на розробку, Bootstrap має безліч переваг, серед яких можливість створення адаптивних крос-браузерних та крос-платформених веб-додатків, що є наразі дуже важливим, адже користувачі можуть відвідувати сайти зі смартфонів, ноутбуків, комп'ютерів, планшетів та інших пристроїв, кожний із яких має різні розміри екрану. Різними можуть бути й браузери та операційні системи, всі вони мають свої особливості. Дуже важливо, щоб сайт виглядав красивим та був зручним для використання на будь-якому пристрої та в будь-якому сучасному браузері. Даний фреймворк якраз надає можливість реалізувати це без великих зусиль, адже всі компоненти були протестовані на різних пристроях величезною кількістю розробників по всьому світу. Bootstrap дуже зручний у використанні та має зрозумілий код, тому його можуть легко освоїти та використовувати всі, навіть ті, хто є новачками у веб-розробці. До того ж Bootstrap має дуже розгорнуту документацію, в Інтернеті можна знайти безліч уроків та інструкцій для роботи з ним, або ж звернутися до спільноти розробників, яка є досить великою. Використання Bootstrap робить дизайн сайту більш гармонійним, легко підтримується єдиний стиль на всіх сторінках. При необхідності можна додати власні стилі поверх стандартних, щоб зробити свій сайт більш унікальним. Звідси випливає і головний недолік – шаблонність. Якщо використовувати лише стандартні інструменти, сайт

втрачає будь-яку оригінальність та стає схожим на безліч інших сайтів, тому Bootstrap може бути використаним не для будь-яких проєктів. Також може виникнути проблема із його підтримкою в більш старих браузерах. Іншим недоліком є те, що кінцеві файли CSS та JavaScript мають занадто великий розмір через вміст стилів і коду JavaScript для різноманітних компонентів, більшість із яких навіть не використовуються на сайті.

Foundation – безкоштовний багатофункціональний адаптивний HTML, CSS і JavaScript фронтенд-фреймворк із відкритим кодом. Допомогає однаково швидко й легко розробити дизайн як для мобільних пристроїв з найменшою роздільною здатністю, так і для телевізорів – з набагато більшою. Містить велику кількість адаптивних класів, які дозволяють реалізувати ретельно налаштований під будь-яку роздільну здатність додаток. Підтримує семантичну верстку (верстка, при якій HTML-теги використовуються у відповідності до їх семантики). Як і Bootstrap, має багато стилізованих компонентів інтерфейсу користувача: кнопки, таблиці, підказки, форми, головна панель з підтримкою трьохрівневої випадаючої навігації. Особливою перевагою даного фреймворку варто виділити можливість налаштування під власні потреби, тобто відключення невикористовуваних елементів. Проте, велика кількість компонентів водночас є суттєвим недоліком – таке різноманіття значно ускладнює роботу. Спільнота розробників є невеликою через складність освоєння.

Skeleton – досить простий фронтенд-фреймворк, заснований на CSS і JavaScript. Надає стилі лише для базових елементів: кнопки, списки, таблиці, форми тощо. Великої кількості компонентів, як у Bootstrap чи Foundation, немає, що полегшує сприйняття та зменшує час, необхідний на освоєння. Тому цей фреймворк може успішно використовуватися для невеликих нескладних проєктів із простим дизайном, для яких різноманіття Bootstrap чи Foundation буде надлишковим. Skeleton дуже добре підходить для адаптивної верстки – адаптивна сітка підлаштовується під будь-яку роздільну здатність

будь якого-якого дисплея, а також під будь-який браузер. Як і Foundation, підтримує семантичну верстку, що також полегшує освоєння даного фреймворку. Skeleton є зручним для створення власного дизайну, оскільки не має заданого стилю, під який потрібно підлаштовуватися. Цим він вигідно відрізняється від Bootstrap, стиль якого є легко розпізнати.

Materialize – сучасний адаптивний CSS і JavaScript фронтенд-фреймворк з відкритим кодом від компанії Google. Він використовує принципи та особливості Material Design та підходить для веб-додатків у традиційному стилі Google. Хоча цей фреймворк є досить новим, він має розгорнуту документацію з прикладами коду для нових користувачів, є простим для сприйняття. Фреймворк пропонує велику кількість стилів для різних елементів, компонентів, які можна змінювати під власні потреби. Крім цього, він має високу продуктивність та швидкість завантаження сторінок. Серед недоліків – шаблонність, яка властива майже всім фронтенд-фреймворкам, а також великий об'єм, що є зворотним боком різноманіття стилів.

Yii – універсальний компонентний об'єктно-орієнтований PHP бекенд-фреймворк з високою продуктивністю, який підходить для розробки практично будь-якого виду веб-додатків. Фреймворк підтримується великою кількістю розробників та є ретельно протестованим у різних ситуаціях користувачами зі всього світу. Він використовує загальноприйнятий архітектурний шаблон MVC, на основі якого і відбувається організація всього коду. Yii забезпечує досить високий рівень надійності (функції для хешування паролів, запобігання SQL-ін'єкціям, додавання CSRF-токена у форми, створені за допомогою ActiveForm). Даний фреймворк підтримує кешування, вбудовану валідацію форм. Для генерації кода, наприклад, Model, View, Controller передбачений графічний модуль Gii, який створює відповідні класи, код яких можна за потреби змінити. Крім готових реалізацій великої кількості стандартних для більшості додатків функцій, таких, як аутентифікація, авторизація, реєстрація, Yii має дуже розгорнуту

документацію, причому перекладену багатьма мовами, зокрема й українською. Це сприяє більш глибокому розумінню принципів роботи фреймворку та скороченню часу на його освоєння. При виникненні проблем чи питань, які докладно не висвітлені в документації, можна завжди звернутися до спільноти розробників, яка є й в Україні. Приємним бонусом є підтримка локалізації – всі повідомлення можна автоматично перекласти за допомогою спеціального компонента `i18n`. Її використовує вбудований фронтенд-фреймворк `Bootstrap`. Це дуже зручно у випадку, якщо цей фреймворк підходить для проекту, над яким ведеться робота. Якщо ж розробник має намір використати інший фронтенд-фреймворк, то вбудованість `Bootstrap` перетворюється на значний недолік, адже повністю відмовитися від його використання навряд чи вдасться. Через велику кількість компонентів у розробника-початківця можуть виникнути проблеми з розумінням їх призначення на етапі освоєння. Синтаксис даного фреймворка також іноді є складним. Оновлення Її виходять дуже рідко, тому на даний момент він є дещо застарілим.

Найголовнішим конкурентом Її є [Laravel](#). Цей безкоштовний PHP-фреймворк з відкритим кодом успішно використовується для будь-яких видів веб-додатків. Він швидко розвивається – оновлення виходять часто. Фреймворк містить реалізацію всіх основних функцій – аутентифікація користувачів, управління сесіями, кешування тощо та легко масштабується. Він має дуже високу швидкість роботи, продуктивність та стабільність. `Laravel` дає можливість швидкого створення панелі адміністратора із врахуванням власних потреб. У якості архітектурного шаблону використовується MVC. Фреймворк має розгорнуту документацію, проте лише англійською мовою, адже він є більш популярним у західних країнах. Крім неї, в Інтернеті можна знайти безліч навчальних матеріалів, які можуть стати в нагоді при вивченні та роботі із фреймворком, але більшість із них також розраховані на англійськомовну аудиторію. Синтаксис `Laravel` є досить

простим та зрозумілим, проте розташування файлів у директоріях дуже заплутане та не завжди логічне.

Zend – популярний безкоштовний об'єктно-орієнтований PHP-фреймворк, який використовує архітектуру MVC та має відкритий код. Переважно використовується при розробці великих проектів. Відмінною рисою є можливість завантаження тільки потрібних компонентів у якості окремих бібліотек, що дозволяє повторно використовувати великий об'єм коду. Передбачена інтеграція із зовнішніми бібліотеками. Zend також дає можливість легко розширювати стандартні функції, код є простим та зрозумілим. Фреймворк є гнучким, надійним (наявне хешування паролів, вбудована валідація форм тощо), стабільним та продуктивним, при цьому не дуже складний для освоєння. В Інтернеті можна знайти багато навчальних матеріалів, проте більшість із них – англійською мовою. Zend підтримує такі бази даних як Microsoft SQL Server, MySQL, Oracle Database, PostgreSQL тощо, забезпечує можливість локалізації. Недоліком даного фреймворку є те, що він дещо повільний, проте у більшості випадків ця проблема не є помітною.

Symfony – безкоштовний гнучкий стабільний фреймворк з великою кількістю не пов'язаних між собою компонентів та готових реалізацій корисних функцій, які можна обирати та конфігурувати за власними потребами. Деякі його компоненти були використані при розробці іншого дуже відомого фреймворка – Laravel. Symfony підтримує інтеграцію із зовнішніми бібліотеками, пакетами та компонентами. Цей PHP-фреймворк є досить поширеним серед розробників та має детальну документацію з прикладами коду та рекомендаціями, яка постійно оновлюється. Проте він є складним для освоєння та не підходить для новачків. Symfony не використовує MVC чи якийсь інший архітектурний шаблон, тому розробникові самотійно доводиться займатися проектуванням. Фреймворк дає можливість перекладу повідомлень про помилки різними мовами.

Підтримується робота з багатьма базами даних (PostgreSQL, MySQL тощо), але вбудовані функції роботи з ними відсутні (потрібно використовувати окремі інструменти). Symfony є відносно продуктивним та підходить для використання великими проектами, він часто оновлюється та є ретельно протестованим, що значно зменшує кількість помилок при розробці. Крім того, фреймворк надає вбудовані можливості тестування. Працює на принципах ООП.

CodeIgniter – бекенд-фреймворк з високою продуктивністю, написаний мовою PHP. Цей фреймворк є популярним завдяки своїй простоті та інтуїтивній зрозумілості, чому сприяє також відмінна документація, відповідно спільнота розробників також є великою. Навіть новачок із невеликим досвідом може швидко освоїти фреймворк та почати розробляти невеликі веб-додатки. Для більш масштабних проектів даний фреймворк не дуже добре підходить, адже має не надто високі стандарти безпеки та дуже рідко оновлюється. Він використовує відомий архітектурний шаблон MVC, а також пропонує різні компоненти, які за потреби можна додатково підключити, зокрема є компонент для кешування сторінок, який значно прискорює їх завантаження. Фреймворк має відкритий зрозумілий код з простими реалізаціями деяких основних функцій, його конфігурація також є незначною.

Phalcon – ще один простий фреймворк, який може бути легко освоєний новачками. Він вирізняється з-поміж вищенаведених фреймворків тим, що написаний мовою C та часто оновлюється. Ця унікальна особливість робить його швидким та ефективним, він має дуже високу продуктивність та низьку ресурсомісткість. Проте, даний фреймворк є набагато менш популярним, ніж вищезгадані, його документацію не можна назвати розгорнутою, пошук навчальних матеріалів також може викликати труднощі, причому для його освоєння бажано мати хоча б базові знання мови C. Проте, його простота, зрозумілість та лаконічність згладжують цей недолік. Даний фреймворк, як і

багато інших, використовує архітектурний шаблон MVC. Ще однією перевагою даного фреймворку є дуже проста інсталяція, яка не вимагає додаткових знань.

SakePHP – досить популярний MVC-фреймворк мовою PHP. Він є дуже продуктивним та надійним, значно спрощує розробку веб-додатків, має відкритий код. Фреймворк легкий в освоєнні, має простий та елегантний код. SakePHP помітно вирізняється великою кількістю різноманітних компонентів, деякі з яких містять функції, що в інших фреймворках вимагають реалізації власноруч. Незважаючи на різноманіття вбудованих функцій, вони можуть бути дуже легко розширені за власними потребами. Фреймворк також пропонує вбудовані генератори коду, в тому числі SQL-запитів. Він надає можливості для прискорення процесу тестування, а також локалізацію. Ще однією відмінною рисою даного фреймворку є дотримання набору угод при кодуванні. Таким чином, код, написаний одним розробником, легше сприймається іншим, що прискорює розробку та спрощує підтримку. Проте, ця особливість може сприйматися і як недолік – накладаються досить суворі обмеження.

На етапі вибору засобів програмування була розглянута велика кількість фронтенд- та бекенд-фреймворків, проаналізовані їх особливості, переваги та недоліки. При виконанні даної роботи було вирішено використовувати бекенд-фреймворк Yii, при цьому були взяті до уваги такі фактори:

- наявність попереднього досвіду роботи із фреймворком;
- наявність розгорнутої документації та навчальних матеріалів із прикладами коду;
- наявність вбудованого фронтенд-фреймворка (наприклад, Bootstrap, який має детальну документацію із прикладами готових стилів, реалізацій деяких функцій мовою JavaScript);
- наявність генераторів коду для прискорення розробки;
- швидке створення адмінпанелі, використовуючи можливості

фреймворку;

- використання архітектурного шаблону MVC (він є дуже широко використовуваним, що дасть можливість легко знайти навчальні матеріали з прикладами коду);
- наявність локалізації (оскільки веб-додаток розробляється для учнів українських шкіл, повинна бути можливість повного перекладу додатку державною мовою).

Ще одним важливим засобом програмування є бази даних. При виконанні даної роботи була використана СУБД на базі MySQL. Це реляційна СУБД, яка є простою у використанні, має розширений функціонал, є досить швидкою, продуктивною та відносно надійною, її зручно використовувати при розробці додатків мовою PHP. Усі ці якості повністю задовольняють потребам додатку, розроблюваного в межах даної роботи. Вирішальним фактором вибору цієї СУБД стала можливість використання phpMyAdmin. phpMyAdmin – це веб-інтерфейс, який дозволяє працювати із СУБД MySQL. Цей веб-додаток дає можливість виконувати основні функції адміністрування через браузер. Він інтегрований у веб-сервер XAMPP, який також використовувався при виконанні даної роботи. Основні функції, які забезпечує phpMyAdmin:

- створення та видалення баз даних, а також робота з ними: створення, видалення, копіювання, модифікація таблиць. Є можливість виконати всі ці операції через графічний інтерфейс, повністю відмовившись від SQL-запитів;
- створення, видалення та редагування полів таблиці також можна виконати повністю через графічний інтерфейс;
- ключові слова, які відповідають за маніпулювання даними, винесені як кнопки, які можна натиснути для отримання згенерованих запитів. Ці запити можна редагувати за власними потребами;
- є можливість завантаження текстових файлів зі скриптами та експорту

даних у різних форматах;

- для більш складних запитів є стандартне поле для їх введення, широко застосовуються підказки;
- можна в будь-який момент переглянути структуру бази, адже вона завжди знаходиться зліва від поля для введення запитів.

Ураховуючи всі ці переваги, СУБД MySQL була обрана для використання під час виконання даної роботи.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1 Проектування структури сайту

Фреймворк Yii, за допомогою якого реалізований веб-додаток, використовує архітектурний шаблон MVC. Як уже зазначалося, його основний сенс у розподілі задач між компонентами Model, View та Controller. Компонент Controller оброблює зміни, які вносить користувач, його запити. Сюди входить, наприклад, натискання на елементи інтерфейсу. Потім Controller оброблює ці зміни та викликає відповідні ресурси (Model та View).

Model відповідає за дані та правила роботи з ними. У відповідь на виклик Controller'а надає йому необхідні дані для відображення за допомогою View.

Саме View відображає дані у кінцевому вигляді, який бачить користувач. Це шаблон, який заповнюється даними, що отримуються від Model.

Компоненти MVC не мають прямого зв'язку між собою, але постійно взаємодіють. Схема цієї взаємодії наведена на Рисунку 3.1 – Схема взаємодії компонентів архітектурного шаблону MVC.

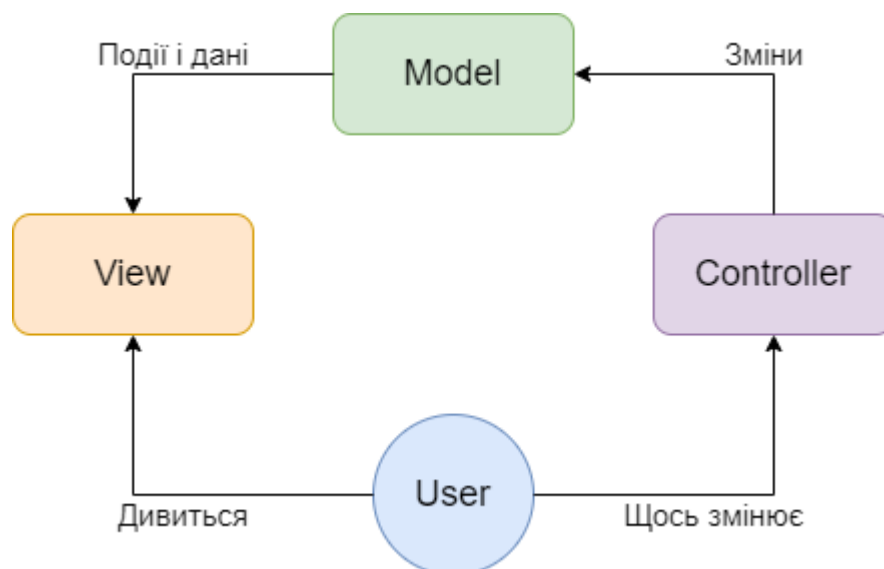


Рисунок 3.1 – Схема взаємодії компонентів архітектурного шаблону MVC

Проект має стандартну структуру, передбачену фреймворком Yii, а саме використовувався шаблон basic. Структура проекту зображена Рисунок 3.2 – Структура проекту за шаблоном Yii basic

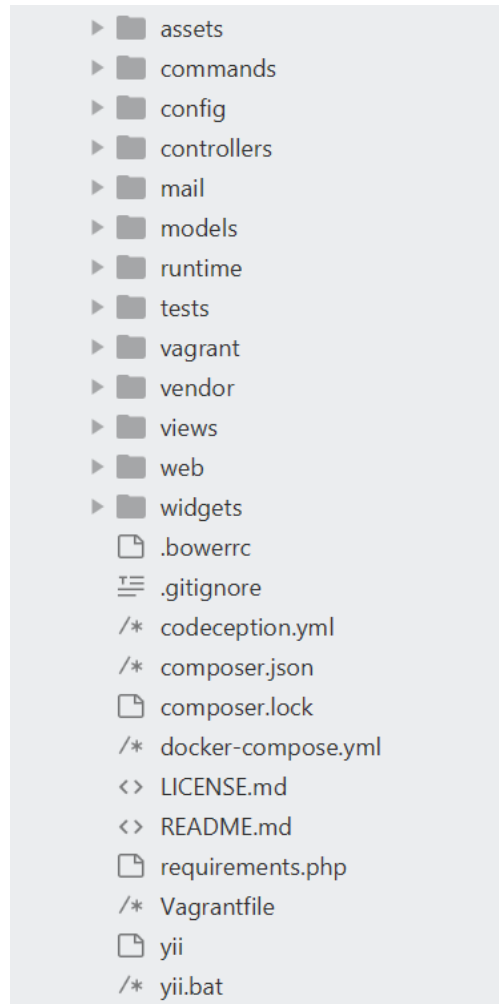


Рисунок 3.2 — Структура проекту за шаблоном Yii basic

Папка `assets` являє собою комплект ресурсів, тобто файлів `css` та `js`, які можна підключати. Можна також додавати свої файли, але в рамках даної роботи в цьому не було необхідності.

Папка `commands` містить консольні команди, вона так само залишилася незмінною.

Папка `config` містить файли конфігурації, зокрема файл `db.php`, в якому прописані параметри підключення до бази даних: її назва, користувач та пароль.

Папка `controllers` містить класи контролерів: головний `SiteController.php`, а також всі інші: `QuestionsController.php`, `TestsController.php`, `UserController.php`.

Папка `mail` містить файли з представленнями для електронної пошти. В рамках розробки даного додатку не було потреби змінювати ці файли.

Папка `models` містить класи `Model`. Для даного додатку це файли: `Gradebook.php`, `LoginForm.php`, `Questions.php`, `Results.php`, `Tests.php`, `TestsSearch.php`, `Users.php`, `UsersSearch.php`.

Папка `runtime` містить файли, що були створені під час роботи додатку (логи та кеш).

Папка `tests` містить тести для веб-додатку. Файли цієї папки не було потреби змінювати.

Папка `vagrant` містить файли з налаштуваннями для роботи з віртуальною машиною.

Папка `vendor` містить власне фреймворк `Yii`, а також зовнішні бібліотеки.

Папка `views` містить папки: `layouts` (шаблони), а також `site`, `questions`, `tests`, `users`, кожна з яких містить файли з представленнями для окремих сторінок.

Папка `web` є головною папкою веб-додатку, вона містить різні стилі, скрипти тощо. Зокрема, тут знаходиться файл `index.php`, який є точкою входу.

Папка `widgets` містить віджети. В рамках виконання даної роботи власні віджети не додавалися.

3.2 Реалізації основних компонентів

Веб-додаток використовує базу даних “`biology`”, яка складається із трьох таблиць: “`users`” – таблиця, що містить дані про всіх користувачів додатку,

“tests” – таблиця, що містить дані про всі тести, “results” – таблиця, що містить дані про результати проходження тестів. ER-діаграма бази даних наведена на Рисунку 3.3 – ER-діаграма бази даних.

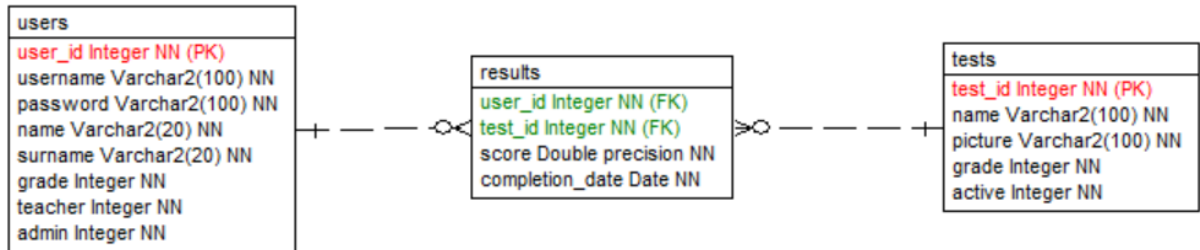


Рисунок 3.3 – ER-діаграма бази даних

Веб-додаток передбачає такий розподіл ролей користувачів за набором прав:

- гість;
- учень;
- учитель;
- адміністратор.

Для ілюстрації функцій, доступних для кожної з ролей користувачів наведена діаграма варіантів використання – Рисунок 3.4 – Діаграма варіантів використання.

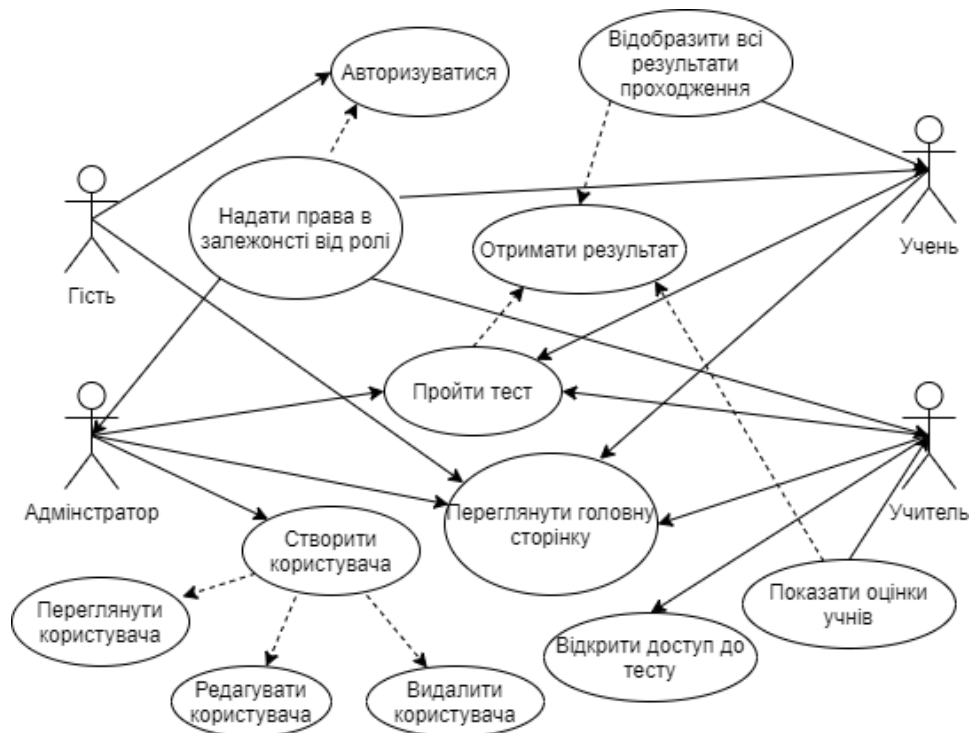


Рисунок 3.4 – Діаграма варіантів використання

Для користувача, який ще не авторизувався на сайті, доступні дві вкладки: «Увійти» та «Головна». Самостійна реєстрація користувачів не передбачена – це задача адміністратора. Вигляд головної сторінки наведений на Рисунку 3.5 – Головна сторінка веб-додатку.

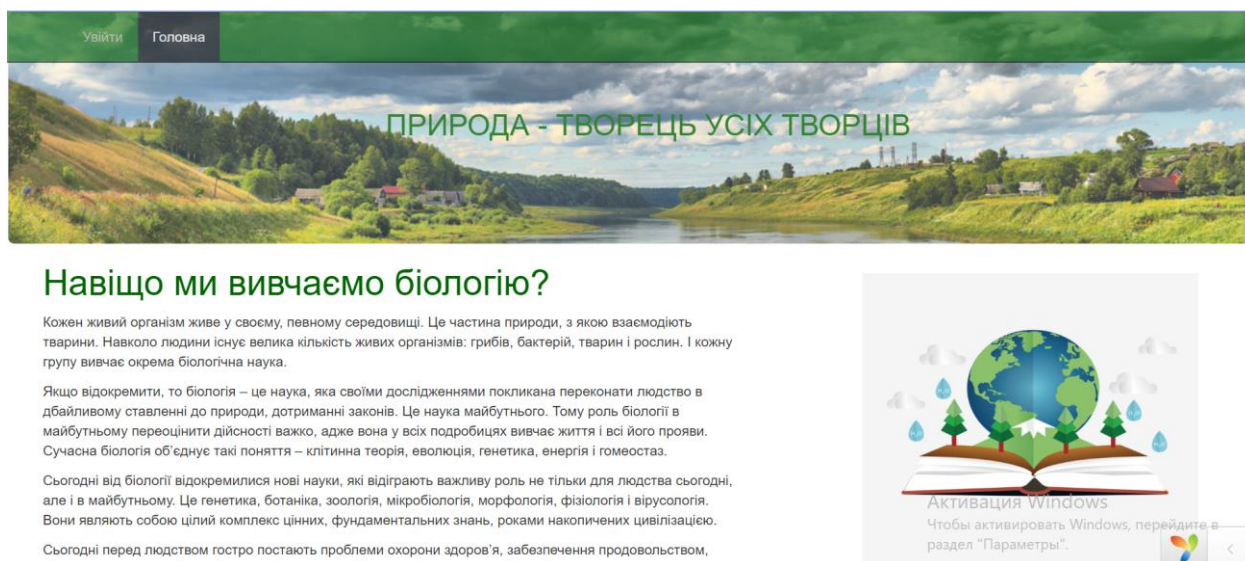


Рисунок 3.5 – Головна сторінка веб-додатку

Доступ до вкладок регулюється у файлі `views/layouts/main.php`. Приклад

коду:

```
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']]
)
),
```

На вкладці «Увійти» користувач повинен ввести свій логін та пароль, після чого натиснути кнопку «Увійти». Логін та пароль для користувачів обирає адміністратор (логін – справжні ім'я та прізвище користувача). Адміністратор надає вчителям та учням їхні вхідні дані, які вони самостійно змінити не можуть. Вигляд сторінки авторизації наведений на Рисунку 3.6 – Сторінка авторизації

A white rectangular form with a green border. At the top center is the title 'Авторизація' in green. Below it is the text 'Будь ласка, заповніть указані поля для входу в систему:'. There are two input fields: the first is labeled 'Логін' and the second is labeled 'Пароль'. At the bottom left of the form is a green button labeled 'Увійти'.

Рисунок 3.6 – Сторінка авторизації

Користувач, що авторизувався як адміністратор, має доступ до таких вкладок: «Вийти (username)», «Головна», «Адміністрування», «Тести». Спеціальною вкладкою, яка доступна лише йому, є «Адміністрування». Вона дозволяє виконувати з користувачами такі дії, як перегляд, пошук, додавання, редагування, видалення. Вигляд сторінки перегляду користувачів наведений на Рисунку 3.7 – Адміністративна сторінка перегляду користувачів

#	ID	Логін	Пароль	Ім'я	Прізвище	Клас	Учитель	Адміністратор	
	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
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 3.7 – Адміністративна сторінка перегляду користувачів

На сторінці відображені всі дані, які є про користувачів в базі. Для кожного стовпця є поле, куди можна ввести дані для пошуку. Приклад пошуку за прізвищем наведений на Рисунку 3.8 – Пошук користувача за прізвищем.

#	ID	Логін	Пароль	Ім'я	Прізвище	Клас	Учитель	Адміністратор	
	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text" value="Воло"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
1	5	anna_volova	12	Анна	Волова	11	0	0	

Рисунок 3.8 – Пошук користувача за прізвищем

Приклад коду, файл views/users/_search.php:

```
<?php $form = ActiveForm::begin([
    'action' => ['index'],
    'method' => 'get',
]); ?>
```

```

<?= $form->field($model, 'id') ?>
<?= $form->field($model, 'username') ?>
<?= $form->field($model, 'password') ?>
<?= $form->field($model, 'name') ?>
<?= $form->field($model, 'surname') ?>
<?= $form->field($model, 'grade') ?>

```

Для створення нового користувача є форма, яка відкривається після натискання кнопки «Створити користувача». Вигляд цієї форми наведений на Рисунку 3.9 – Сторінка створення нового користувача.

The screenshot shows a web page with a green header containing navigation links: 'Вийти (admin)', 'Головна', 'Адміністрування', and 'Тести'. Below the header is a large green title 'Створити користувача'. The form consists of several input fields, each with a label above it: 'Логін' (Login), 'Пароль' (Password), 'Ім'я' (Name), 'Прізвище' (Surname), 'Клас' (Class), 'Учитель' (Teacher), and 'Адміністратор' (Administrator). Each field contains a placeholder text: 'Введіть логін користувача', 'Введіть пароль користувача', 'Введіть ім'я користувача', 'Введіть прізвище користувача', 'Введіть клас, у якому навчається користувач (для учнів)', 'Введіть 1, якщо користувач є вчителем', and 'Введіть 1, якщо користувач є адміністратором'. At the bottom of the form is a green button labeled 'Зберегти' (Save).

Рисунок 3.9 – Сторінка створення нового користувача

Приклад коду, файл `views/users/_form.php`:

```

<?php $form = ActiveForm::begin(); ?>
    <?= $form->field($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']) ?>

```

Передбачена валідація даних, введених користувачем, приклад коду,
файл `models/Users.php`:

```

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!'],
    ];
}

```

На форму редагування користувача можна перейти, натиснувши кнопку-олівець на сторінці перегляду користувачів. Вигляд цієї форми наведений на Рисунку 3.10 – Сторінка редагування користувача

Вийти (admin) Головна Адміністрування Тести

Редагувати користувача: Анна Волова

Логін

Пароль

Ім'я

Прізвище

Клас

Учитель

Адміністратор

Рисунок 3.10 – Сторінка редагування користувача

Для видалення користувача потрібно натиснути на кнопку-кошик на сторінці перегляду користувачів, а потім підтвердити видалення. Повідомлення при видалення наведено на Рисунку 3.11 – Повідомлення при видаленні користувача.

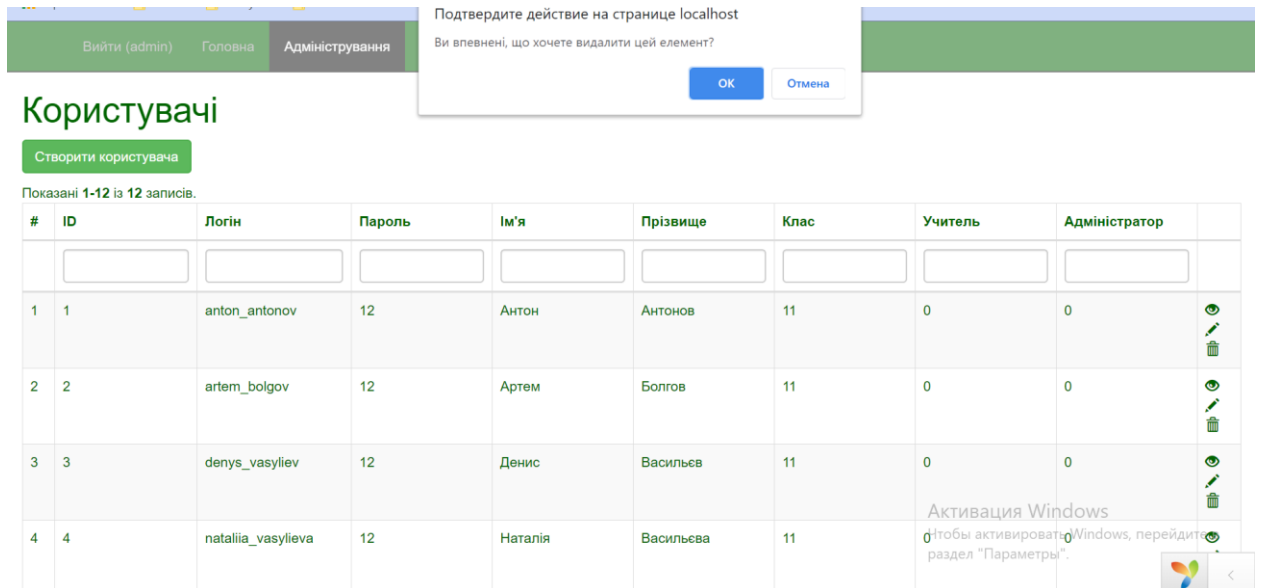


Рисунок 3.11 – Повідомлення при видаленні користувача

Окрім перегляду списку користувачів, є також можливість перегляду кожного користувача окремо. Вигляд сторінки перегляду одного користувача наведений на Рисунку 3.12 – Сторінка перегляду користувача.

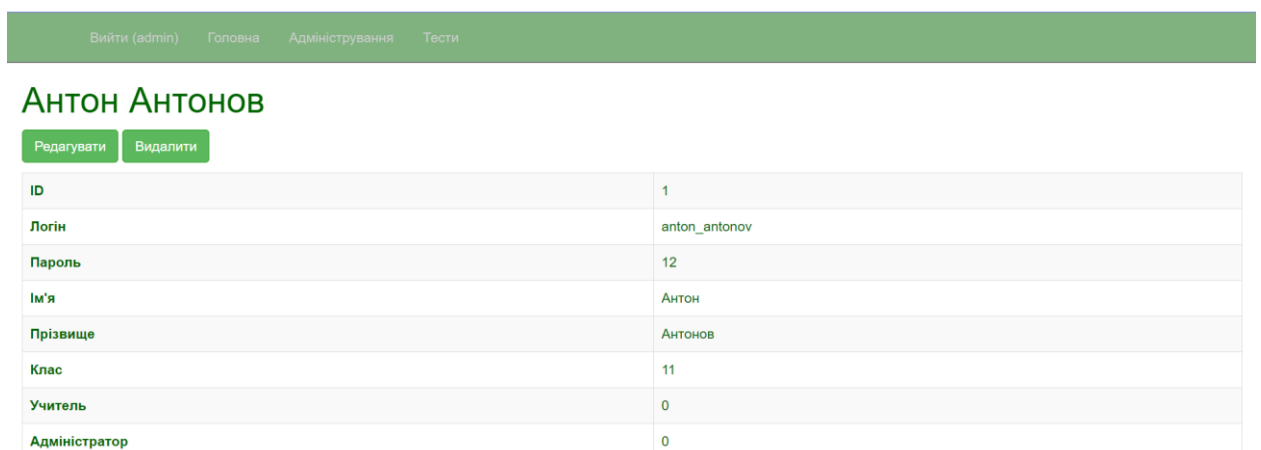


Рисунок 3.12 – Сторінка перегляду користувача

Користувач, що авторизувався як Учитель, має доступ до таких вкладок: «Вийти (username)», «Головна», «Адміністрування», «Тести», «Журнал». Спеціальними вкладками, доступними тільки вчителю, є «Адміністрування» та «Журнал». Сторінка адміністрування має схожий вигляд зі сторінкою адміністрування для адміністратора. Вона виконує такі самі функції (крім створення та видалення), але не з користувачами, а з тестами. Головне призначення сторінки – надати можливість учителю відкривати та закривати доступ до тестів задля унеможливлення проходження тесту в невідведений для цього час. Варто зазначити, що на самого вчителя, а також на адміністратора це правило не поширюється – вони мають доступ до усіх тестів у будь-який час. Вигляд сторінки перегляду тестів наведений на Рисунку 3.13 – Адміністративна сторінка перегляду тестів.

#	ID	Назва	Зображення	Клас	Доступ
1	1	Популяція. Екосистема. Біосфера	test1.jpg	11	1

Рисунок 3.13 – Адміністративна сторінка перегляду тестів

Вкладка «Журнал» дає можливість перегляду оцінок учнів обраного класу за всі тести. Після переходу на вкладку відкривається сторінка, вигляд якої наведений на Рисунку 3.14 – Вкладка «Журнал»

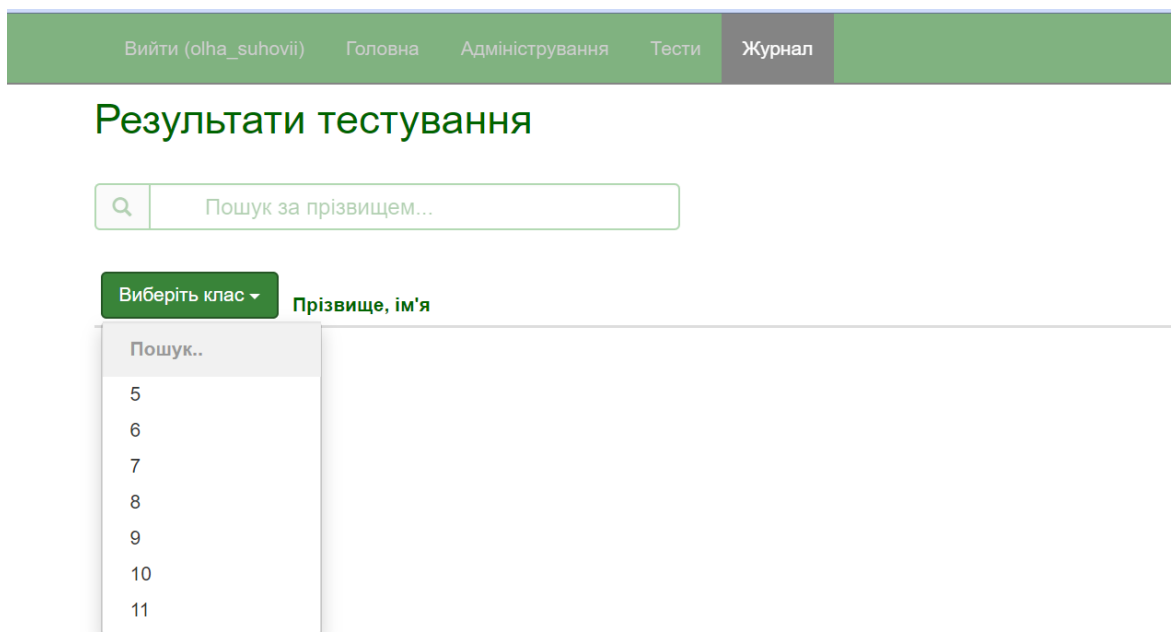


Рисунок 3.14 – Вкладка «Журнал»

Після обрання певного класу вчитель зможе побачити оцінки учнів у 12-бальній системі, причому відображається тільки найуспішніша спроба. Є можливість пошуку за прізвищем учня. Тести в таблиці відображаються у вигляді «Тема ...». Для того, щоб побачити назву тесту, потрібно навести мишкою на тему. Якщо потрібно дізнатися дату проходження тесту, потрібно навести мишкою на оцінку. Вигляд цієї сторінки наведений на Рисунку 3.15 – Сторінка перегляду оцінок учнів

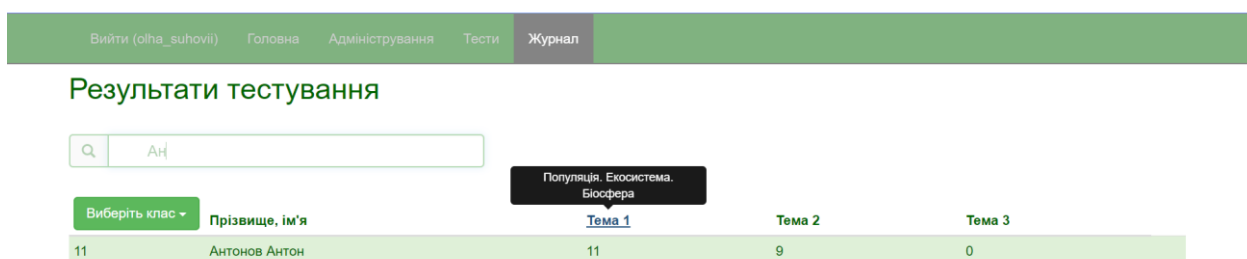


Рисунок 3.15 – Сторінка перегляду оцінок учнів

Приклад коду:

```

$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' =>
$_GET['grade']])->all(); ?>
foreach ($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; ?>

```

Користувач, який авторизувався як Учень, має доступ до таких вкладок: «Вийти (username)», «Головна», «Мої результати», «Тести». Вкладка «Мої результати» створена спеціально для учня. На цій сторінці можна переглянути свої результати за всі тести, причому відображені всі спроби. Є можливість пошуку за назвою тесту. Всі стовпці фільтруються за зростанням та спаданням (для цього потрібно натиснути на назву стовпця). Рядки таблиці мають різні кольори в залежності від рівня оцінки. Якщо оцінка високого рівня – зелений колір, середнього – жовтий, низького – червоний. Вигляд сторінки наведений на Рисунку 3.16 – Персональна сторінка учня із результатами тестування

Назва теста	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 3.16 – Персональна сторінка учня із результатами тестування

Приклад коду:

```

$sql_name = 'SELECT name, score, completion_date from tests
join results on id = test_id where pupil_id = :id';
$res_name = Tests::findBySQL($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';

```

```

$res_score = Results::findBySQL($sql_score, [':id' =>
Yii::$app->user->identity->id])->all();
foreach ($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; ?>

```

На вкладці «Тести» учень може бачити всі тести (із відкритим і закритим доступом), але тільки свого класу. Для кожного тесту відображається останній результат його проходження. Сторінка із тестами наведена на Рисунку 3.17 – Сторінка з тестами.

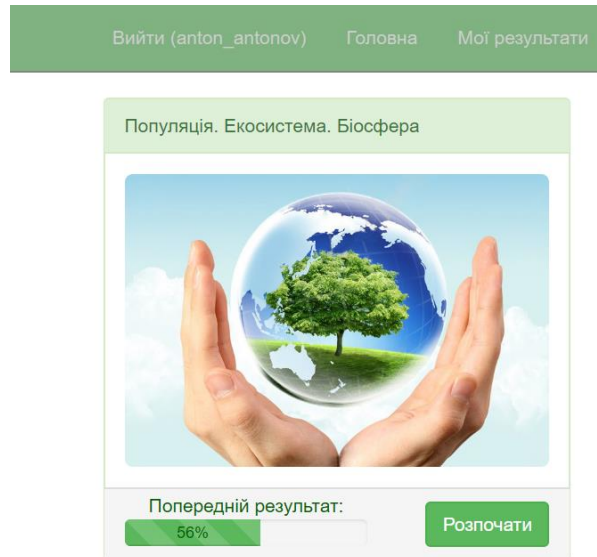


Рисунок 3.17 – Сторінка з тестами










Після натискання кнопки «Розпочати» користувач переходить на перше питання обраного тесту. Користувачеві пропонується питання, даючи відповідь на яке, він має перетягти зображення у відповідні стовпці. Для кожного зображення вказане його значення. Його можна побачити при наведенні мишкою. Код наведений у Додатку А. Питання наведене на Рисунку 3.18 – Перше питання тесту

Вийти (anton_antonov) Головна Мої результати Тести

Тестування: 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

Дати відповідь Далі










Рисунок 3.18 – Перше питання тесту

Після того, як користувач натиснув кнопку «Дати відповідь», він може переглянути, чи є його відповідь правильною. Правильно розміщені зображення виділяються зеленим, неправильно – червоним. Перевірене питання наведено на Рисунку 3.19 – Перевірене перше питання

Вийти (anton_antonov) Головна Мої результати Тести

Тестування: 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
  	  	  

Дати відповідь Далі

Рисунок 3.19 – Перевірене перше питання

Після натискання кнопки «Далі», яка стає доступною лише після того, як була дана відповідь, відкривається сторінка другого питання. Користувачеві пропонується відповісти на питання, поєднавши відповідні блоки. Є можливість скасувати всі з'єднання (але лише до натиснення кнопки «Дати відповідь»). Відповіді також перевіряються та виділяються

кольором відповідно до правильності. Код наведений у Додатку Б. Друге питання наведене на Рисунку 3.20 – Друге питання тесту

Вийти (anton_antonov) Головна Мої результати Тести

Тестування: Тема 1 Популяція. Екосистема. Біосфера

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

ЕРОЗІЯ ҐРУНТІВ	ІНТЕНСИВНА ДІЯЛЬНІСТЬ ПРОМИСЛОВОСТІ ТА ТРАНСПОРТУ
ЗАСОЛЕННЯ ҐРУНТІВ	ВИКИДИ В АТМОСФЕРУ ХЛОРОФТОРВУГЛЕЦЕВИХ СПОЛУК
УТВОРЕННЯ ОЗОНОВИХ ДІРОК	МАСОВЕ ВИРУБУВАННЯ ЛІСІВ
ТЕПЛИЧНИЙ ЕФЕКТ	НЕРАЦІОНАЛЬНЕ ПОЛИВАННЯ ҐРУНТІВ

Скасувати Дати відповідь Далі

Рисунок 3.20 – Друге питання тесту

У третьому питанні тесту користувач має розфарбувати книги, які спочатку є сірими, в один із кольорів відповідно до опису (описані документи: Червона книга, Зелена книга, Чорна книга). Відповіді перевіряються. Код наведений у Додатку В. Вигляд третього питання наведений на Рисунку 3.21 – Третє питання тесту

Вийти (anton_antonov) Головна Мої результати Тести

Тестування: Тема 1 Популяція. Екосистема. Біосфера

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

	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 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.

Активізація Windows
Дати відповідь Далі
Windows, перейдіть в розділ "Параметри".

Рисунок 3.21 – Третє питання тесту

У четвертому питанні користувач має шляхом перетягування розмістити зображення заповідників у відповідні блоки в залежності від їх місцезнаходження. Для кожного зображення шляхом наведення мишкою можна дізнатися назву заповідника, про який іде мова. Є можливість скасувати розміщення зображень та почати відповідати заново. Код наведений у Додатку Г. Вигляд четвертого питання після перевірки відповіді наведений на Рисунку 3.22 – Четверте питання тесту.

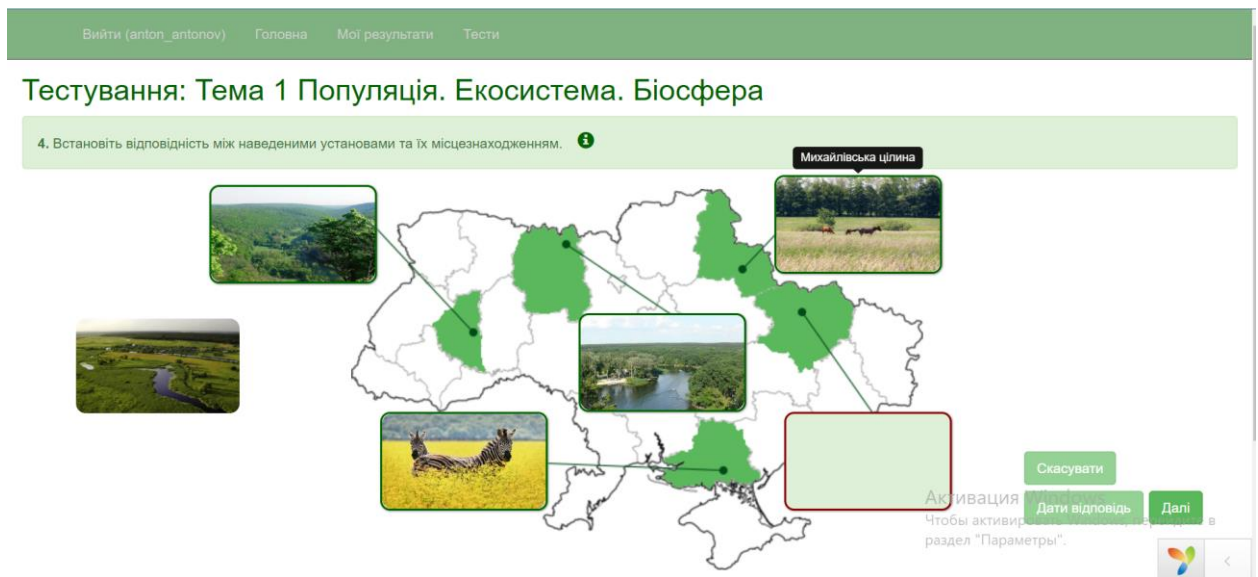


Рисунок 3.22 – Четверте питання тесту

П'яте питання – кросворд. Питання наведені у вигляді списку. Їх можна по черзі переглядати, натискаючи на потрібне. Є можливість перемикатися між клітинками за допомогою кнопки Tab. У виділених клітинках зашифроване слово. Якщо його відгадати, з'являються допоміжні букви для інших слів. Код наведений у Додатку Г. Перевірене питання наведено на Рисунку 3.23 – П'яте питання тесту.

Вийти (anton_antonov) Головна Мої результати Тести

Тестування: Тема 1 Популяція. Екосистема. Біосфера

5. Розв'яжіть кросворд.

ПИТАННЯ

Питання 1

Питання 2

Питання 3

Питання 4

Питання 5

Питання 6

Угруповання популяцій організмів різних видів, які населяють ділянку біосфери з однорідними умовами існування і пов'язані між собою різноманітними взаємозв'язками.

1	А	А	А	А	А	А	А				
	Д	У	Ц	Е	Н	Т	И				
4											
5			П	О	П	У	Л	Я	Ц	І	Я
6	Н	О	О	С	Ф	Е	Р	А			

Активация Windows [Дати відповідь](#) [Далі](#)

Чтобы активировать Windows, перейдите в раздел "Параметры".


Рисунок 3.23 – П'ять питань тесту

Відповівши на всі питання, учень отримує можливість побачити свій остаточний результат. Вигляд сторінки з результатом наведений на Рисунку 3.24 – Сторінка з результатом тестування.

Вийти (olha_suhovii) Головна Адміністрування Тести Журнал

Тестування: Тема 1 Популяція. Екосистема. Біосфера

Вітаємо, тестування закінчене! Ваш результат: 10 (83,3%)



Активация Windows [Дати відповідь](#) [Далі](#)

Чтобы активировать Windows, перейдите в раздел "Параметры".

Рисунок 3.24 – Сторінка з результатом тестування.

ВИСНОВКИ

У рамках виконання даної роботи було проведене дослідження на предмет використання інформаційних технологій в середніх навчальних закладах України. Був зроблений висновок, що ІТ технології використовуються неефективно, хоча більшість шкіл забезпечені необхідним обладнанням. Ураховуючи, що комп'ютери зазвичай є застарілими, було прийняте рішення розробити веб-додаток, що спеціалізується на автоматизації тестування учнів з біології. Пошук та порівняння аналогічних додатків показали, що майже всі додатки є спрощеними та нецікавими для учнів у контексті питань. У відкритому доступі був знайдений лише один додаток, який, крім стандартних питань із варіантами відповідей, пропонував інтерактивні питання, в яких учням необхідно встановити відповідність шляхом перетягування блоків. Проте, більша частина питань були однакової концепції – питання з одним або декількома варіантами відповідей.

Ураховуючи результати проведених досліджень, було вирішено головною концепцією розроблюваного додатку обрати максимальну інтерактивність та дружність до учня. В результаті вдалося спроектувати та розробити веб-додаток із такими основними функціями: авторизація користувачів з трьома ролями: учень, учитель, адміністратор, реалізація адміністративних функцій для вчителя (відкриття доступу до тесту, перегляд інформації про тест), та адміністратора (створення, перегляд, редагування та видалення користувачів), перегляд учнями своїх оцінок за всі спроби (з можливістю пошуку за назвою тесту та фільтрації стовпців), перегляд учителями журналу оцінок учнів вибраного класу (з можливістю пошуку за прізвищем). Крім цього, розроблені концепції п'яти інтерактивних питань, які були успішно реалізовані.

Даний веб-додаток може бути розширений іншими тестами, після чого може бути використаний у середніх навчальних закладах.

СПИСОК ЛІТЕРАТУРИ

1. Уайнсет Дж., Сафронов М. Разработка веб-приложений в Yii 2. – Пер. с англ. – ДМК Пресс, 2015. – 392с.
2. Bogdanov A., Eliseev D., Makarov A. Yii2 Application Development Cookbook - Third Edition – 2016, 584 с.
3. Ullman L. The Yii Book (2nd ed.). – 2015. – 300 с.
4. Caldarelli F. Yii2 By Example. – 2015. 290 с.
5. Myers M. A Smarter Way to Learn HTML & CSS: Learn it faster. Remember it longer. – 2015, 258 с.
6. Кириченко А.В, Хрусталёв А.А HTML5+CSS3. Основы современного Web-дизайна – СПб.: «Наука и техника», 2018 г. – 352 с., ил.
7. МакГрат, Майк. PHP7 для начинающих с пошаговыми инструкциями / Майк МакГрат. – Москва : Издательство «Э», 2018. 256 с. – (Программирование – это просто).
8. Kyrnin J, Meloni J. HTML, CSS, and JavaScript All in One. – 2018. – 800 с.
9. Brown T. CSS Master 2nd edition— 2018. — 351 с
10. Felke-Morris T. Basics of Web Design: HTML5 & CSS (5th Edition). — 2019. — 475 с.
11. Carey P. New Perspectives of HTML 5 and CSS: Comprehensive – 2019. – 796 с.
12. Скляр Д. Изучаем PHP 7: руководство по созданию интерактивных веб-сайтов. Пер. с англ. – СПб.: ООО «Альфа книга», 2017. – 464 с., ил.
13. Самоучитель PHP 7 / М.В. Кузнецов, И.В. Симдянов. – СПб.: БХВ-Петербург, 2018. – 448 с.:ил.

14. Веллинг Л, Томсон Л. Разработка веб-приложений с помощью PHP и MySQL. Руководство. Пер. с англ. – Москва: «Альфа-книга», 2016. – 768 с.
15. Зандстра М. PHP: Объекты, шаблоны и методики программирования. – Москва: «Вильямс», 2015. – 577 с.
16. Локхарт Дж. Современный PHP. Новые возможности и передовой опыт. – Москва: «ДМК Пресс», 2016. – 304 с.

ДОДАТКИ

Додаток А

У даному додатку наведений код першого питання

```
$(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 >= col1_x1 && therm_x <= col1_x2 &&
therm_y >= col1_y1 && therm_y <= col1_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>АБИОТИЧНИ</strong></
header></div>
    <div class="column" id="column2" draggable="false"
ondragenter="dragEnter(event)" ondrop="dragDrop(event)"
ondragover="dragOver(event)"><header><strong>БИОТИЧНИ</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 Б

У даному додатку наведений код другого питання

```
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 ГРҮНТІВ</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>ЗАКОЛЕЕННЯ ГРУНТИВ</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 ОЗООНОВИХ
ДИПОК</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>

```

```

    };
</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;
            }
        }
    </script>

```

```

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>ХЛОРФТОРВУГЛЕЦЕВИХ<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
ЛIICIB</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>НЕПАЦИОНАЛЬНЕ
ПОЛИВАННЯ<br>ГРУНТІВ</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;  
        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);
}
else {

}
};
</script>
```

Додаток В

У даному додатку наведений код третього питання

```

<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>
<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 Г

У даному додатку наведений код четвертого питання

```

<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>
</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_l');
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 Г

У даному додатку наведений код п'ятого питання

```
<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

Питання 3

Питання 4

Питання 5

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 == 'c' &&
```

```
    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{
'darkred';
    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{
'darkred';
    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{
        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>

```