

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
Державний університет “Житомирська політехніка”  
Кафедра інженерії програмного забезпечення

## **ЗВІТ 3**

# **ПЕРЕДДИПЛОМНОЇ ПРАКТИКИ**

**ЦВІКА Олександра Сергійовича**

студента 4 курсу

групи ІПЗ-20-3

галузь знань 12 «Інформаційні технології»

спеціальність 121 «Інженерія програмного  
забезпечення»

освітня програма «Інженерія програмного  
забезпечення»

освітній ступінь «бакалавр»

Керівник: КРАВЧЕНКО Світлана Миколаївна

## РЕФЕРАТ

Звіт з переддипломної практики складається зі вступу, чотирьох розділів, висновків, списку використаної літератури та додатків.

Текстова частина викладена на 26 сторінках друкованого тексту.

Список використаних джерел містить 13 найменувань і займає 2 сторінки. Звіт не має додатків. У роботі наведено 6 рисунків. Загальний обсяг роботи – 28 сторінок.

У першому розділі було викладено розгорнутий перелік задач, що необхідно вирішити під час опрацювання роботи та реалізувати у вигляді програмного комплексу.

У другому розділі описано аналогічні програмні продукти, що наявні на ринку із визначенням їх основних функціональних можливостей.

У третьому розділі описано оптимальну архітектурну модель для забезпечення ефективності та масштабованості системи. Розроблено діаграму розгортання, схему слоїв та описано взаємодію між компонентами системи.

У четвертому розділі описано технології для розробки вебдодатку та визначено вимоги до апаратного забезпечення.

Висновки містять ключові аспекти для успішної реалізації системи електронного навчання, включаючи вибір архітектури та технологій розробки.

**КЛЮЧОВІ СЛОВА:** ASP.NET CORE, EF CORE, ORM, SPA, БД, ВЕБДОДАТОК, СИСТЕМА ЕЛЕКТРОННОГО НАВЧАННЯ, ЧИСТА АРХІТЕКТУРА.

					Житомирська політехніка.24.121.19.080 - ПП			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Цвік О. С.			Переддипломна практика  Звіт	Літ.	Арк.	Аркушів
Перевір.							2	28
Керівник		Кравченко С. М.				ФІКТ Гр. ІПЗ-20-3		
Н. контр.								
Зав. каф.								

## ЗМІСТ

ВСТУП .....	3
РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ.....	6
Висновки до першого розділу.....	8
РОЗДІЛ 2. АНАЛІЗ АНАЛОГІВ ПРОГРАМНОГО ПРОДУКТУ .....	9
2.1. Google Classroom .....	9
2.2. Moodle.....	10
2.3. Classdojo .....	13
Висновки до другого розділу .....	14
РОЗДІЛ 3. ВИЗНАЧЕННЯ АРХІТЕКТУРИ ВЕБДОДАТКУ .....	15
Висновки до третього розділу.....	21
РОЗДІЛ 4. ОБҐРУНТУВАННЯ ВИБОРУ ІНСТРУМЕНТАЛЬНИХ ЗАСОБІВ ТА ВИМОГИ ДО АПАРАТНОГО ЗАБЕЗПЕЧЕННЯ .....	22
Висновки до четвертого розділу .....	25
ВИСНОВКИ.....	26
ЛІТЕРАТУРА .....	27

## ВСТУП

**Актуальність теми.** У сучасному світі, де технології швидко розвиваються, інтернет відіграє важливу роль у житті людей, він дає безліч можливостей та способів для розваг та розвитку людей, це місце де можна отримати якісну освіту не виходячи з дому. Дистанційна освіта стає все популярнішою через збільшення доступності та простоти у використанні, вона дозволяє економити час та гроші, адже не потрібно кудись збиратись та їхати.

Системи електронного навчання стають необхідним елементом для забезпечення доступу до освіти в будь-якому місці та у будь-який час. Впровадження даних систем дає можливість полегшити доступ до навчальних матеріалів, сприяє ефективній взаємодії студентів та викладачів, а також оптимізує процеси оцінювання.

Студенти мають можливість працювати з різними типами навчальних матеріалів, наприклад, презентаціями, тестами, лабораторними завданнями чи посиланнями на інші джерела, при цьому навчаючись у власному темпі та в зручний для них час. Викладачі можуть створювати різні курси та наповнювати їх різноманітними матеріалами, виставляти оцінки тощо. Така система не лише допомагає у засвоєнні матеріалу, а й стимулює до саморозвитку всіх учасників навчального процесу.

**Метою** даної роботи є проектування системи електронного навчання. Система має забезпечувати функціонал, який спрямований на оптимізацію процесів навчання та надасть студентам і викладачам можливість ефективно взаємодіяти між собою.

Поставлена мета обумовлює необхідність виконання наступних **завдань**:

- виконати аналіз функціональних потреб системи;
- обрати оптимальну в сучасних умовах архітектуру функціонування програмного комплексу;
- визначити технології та інструменти для реалізації.

		Цвік О. С.			Житомирська політехніка.24.121.19.080 - ПП	Арк.
		Кравченко С. М.				4
Змн.	Арк.	№ докум.	Підпис	Дата		

**Об'єктом** дослідження є система електронного навчання. **Предметом** дослідження є використання сучасних технологій розробки веб-додатків та організації їх інформаційної взаємодії для створення системи електронного навчання.

Базою практики є ФОП Бернацький М.В., що зареєстрований за адресою вул. Чуднівська, 103Б, м. Житомир, 10005. До основних видів діяльності відносяться: комп'ютерне програмування; консультування з питань інформатизації; діяльність із керування комп'ютерним устаткуванням; оброблення даних, розміщення інформації на веб-вузлах і пов'язана з ними діяльність тощо. Завданням практики було виконати аналіз напрямків використання інформаційних технологій для реалізації системи електронного навчання.

		Цвік О. С.			Житомирська політехніка.24.121.19.080 - ПП	Арк.
		Кравченко С. М.				5
Змн.	Арк.	№ докум.	Підпис	Дата		

## РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ

Система електронного навчання – це цифрова платформа, яка дозволяє створювати навчальні матеріали, керувати ними та розповсюджувати їх за допомогою інтернету. Це середовище та інструмент для передачі знань. Вона пройшла кілька десятків років трансформацій та досягнула свого кращого вигляду завдяки розвитку інтернет-технологій. Це платформа для централізованого створення та організації навчального контенту. На ній викладачі можуть завантажувати матеріали курсу, створювати тести та оцінювати роботи студентів. Студенти можуть переглядати матеріали курсів, виконувати завдання та отримувати зворотний зв'язок [13].

Зараз дистанційні засоби навчання стають необхідною складовою сучасної освіти, забезпечуючи студентам та викладачам нові можливості для ефективної взаємодії та покращення якості освіти. Основною метою розробки системи електронного навчання є створення зручного та інтерактивного середовища для навчання та обміну інформацією між учасниками освітнього процесу.

У сучасних умовах розвитку інформаційних технологій електронне навчання є одним з найбільш ефективних способів надання освітніх послуг. Система електронного навчання дозволяє студентам отримувати освіту в будь-який час і в будь-якому місці, що робить її особливо актуальною для студентів, які проживають у віддалених регіонах або мають обмежені можливості для відвідування навчальних закладів.

Дана система має забезпечувати доступ до широкого спектру освітніх ресурсів, включаючи різноманітні навчальні матеріали, тести, та можливість взаємодії з викладачами та іншими студентами. Використання такої системи дозволить підвищити ефективність навчання, зробити його більш індивідуальним і адаптованим до потреб кожного студента.

Для того, щоб успішно завершити створення системи електронного навчання потрібно виконати наступні завдання:

		Цвік О. С.			Житомирська політехніка.24.121.19.080 - ПП	Арк.
		Кравченко С. М.				6
Змн.	Арк.	№ докум.	Підпис	Дата		

- проаналізувати аналоги програмних продуктів для електронного навчання;
- визначити функціональні вимоги до системи;
- обрати оптимальну архітектуру для функціонування системи;
- спроектувати базу даних, яку буде легко масштабувати;
- розробити методи взаємодії з системою для студентів та викладачів, включаючи перегляд оцінок, матеріалів курсів, проходження тестів, обмін повідомленнями тощо;
- створити адаптивний, зрозумілий та простий дизайн для веб-додатка;
- розробити потрібні компоненти, сторінки та функціонал;
- провести тестування роботи додатку.

Першим важливим кроком є аналіз аналогів, тобто додатків, що мають подібний функціонал. На цьому етапі потрібно визначити переваги та недоліки інших систем, які варто врахувати при розробці власної, а також знайти корисні та додаткові можливості. Виконання даного кроку дозволить уникнути помилок, які наявні в інших системах й отримати якісний та унікальний продукт.

Наступним кроком варто визначити функціональні та технічні вимоги до системи електронного навчання, зокрема можливості перегляду та завантаження навчальних матеріалів, взаємодії між користувачами, проведення тестів, оцінювання студентів тощо.

Загалом розробку даної системи можна розділити на 3 частини – проектування бази даних, створення інтерфейсу та розробки RESTful API, який забезпечує спосіб звернення до функцій додатку через визначені URL-адреси та HTTP-методи, такі як GET, POST, PUT, PATCH, DELETE та ін.

Першою частиною у створенні проєкта є вибір СУБД та проектування бази даних. Це дуже важлива частина розробки, адже до БД буде відправлятися багато запитів для виконання різних операцій з даними. Наприклад, отримання успішності студентів з конкретного курсу та в подальшому генерування електронного журналу на клієнті, або додавання до журналу порожнього

		Цвік О. С.			Житомирська політехніка.24.121.19.080 - ПП	Арк.
		Кравченко С. М.				7
Змн.	Арк.	№ докум.	Підпис	Дата		

стовпця, в клітинки якого можна вводити оцінки та виставляти присутність тощо.

Після проєктування БД потрібно розробити серверну частину додатку, де необхідно налаштувати автентифікацію та авторизацію, адже на сайті передбачається користувачі трьох типів: адміністратор, викладач та студент. Важливо налаштувати чіткий контроль доступу до різних частин сайту, щоб наприклад студент не мав змогу змінити собі або іншим студентам оцінки в журналі, а викладач не мав доступу змінювати оцінки в тих курсах, які йому не належать, тому важливо налаштувати захищену систему. Також на сервері потрібно виконувати перевірку даних, які надходять від користувачів, перед тим як виконувати CRUD операції. Якщо дані некоректні, сервер повинен відправити відповідне повідомлення в якому буде вказано інформацію про помилку.

Та третьою частиною є створення дизайну для головної частини сайту та адмін-панелі. Оскільки на сайті будуть присутні адміністратори, які матимуть доступ до всіх частин системи з розширеним функціоналом, отже для них інтерфейс буде відрізнятися від основного. Крім того, вони зможуть переглядати статистику по різних критеріям. Головна частина сайту включає в себе сторінку особистого кабінету, де розміщено інформацію про авторизованого користувача, а також інші сторінки такі як курси, тести, чати, журнал успішності та інші.

Результатом успішної реалізації поставленого завдання є створена система електронного навчання, яка забезпечує доступ до широкого спектру освітніх ресурсів, включаючи різноманітні навчальні матеріали, тести та забезпечує можливість взаємодії між студентами та викладачами. Використання цієї системи сприяє підвищенню ефективності навчання, роблячи його більш індивідуальним та адаптованим до потреб кожного студента.

### **Висновки до першого розділу**

Проаналізовано вимоги до програмного продукту та предметної області. Визначено завдання, які необхідно вирішити для успішної реалізації проєкту.

		Цвік О. С.			Житомирська політехніка.24.121.19.080 - ПП	Арк.
		Кравченко С. М.				8
Змн.	Арк.	№ докум.	Підпис	Дата		



## РОЗДІЛ 2. АНАЛІЗ АНАЛОГІВ ПРОГРАМНОГО ПРОДУКТУ

В умовах швидкого розвитку технологій та широкого використання інтернету, системи електронного навчання стають важливою складовою сучасної освіти. Вони не лише дозволяють створювати та управляти навчальним контентом, але й відкривають нові горизонти для дистанційного навчання та забезпечують централізовану платформу для обміну знаннями.

Після детального аналізу і перегляду подібних систем стало зрозуміло, що світовий ринок має дуже багато схожих продуктів, адже інтернет в наш час дуже стрімко розвивається і є популярним. Зараз існує безліч різних систем, які пропонують різні функціональні можливості, які іноді є подібними, а іноді представляють унікальні рішення.

Отже, розглянемо найбільш популярні системи для дистанційного навчання з метою виокремлення ключових особливостей та визначення аспектів, які можна впровадити у власний продукт.

### 2.1. Google Classroom

Google Classroom [4] є популярною платформою для електронного навчання, розробленою компанією Google. Цей інструмент значно спрощує процес організації та створення навчальних завдань для вчителів та студентів, забезпечуючи ефективний зворотний зв'язок. Цей продукт інтегрується з іншими Google-сервісами, такими як Google Drive, Google Docs та Gmail. Інтерфейс платформи дуже зручний і простий, тому складнощів у знайомстві з функціями не виникне.

Google Classroom навряд чи можна назвати класичною системою дистанційного навчання, це скоріше середовище для спільної роботи, де Google об'єднали всі необхідні інструменти для ефективної освіти в одному місці. Але дана система відмінно підійде для знайомства з онлайн-навчанням. Для того щоб скористатися функціями даного сервісу, досить створити обліковий запис в Google.

		Цвік О. С.			Житомирська політехніка.24.121.19.080 - ПП	Арк.
		Кравченко С. М.				9
Змн.	Арк.	№ докум.	Підпис	Дата		

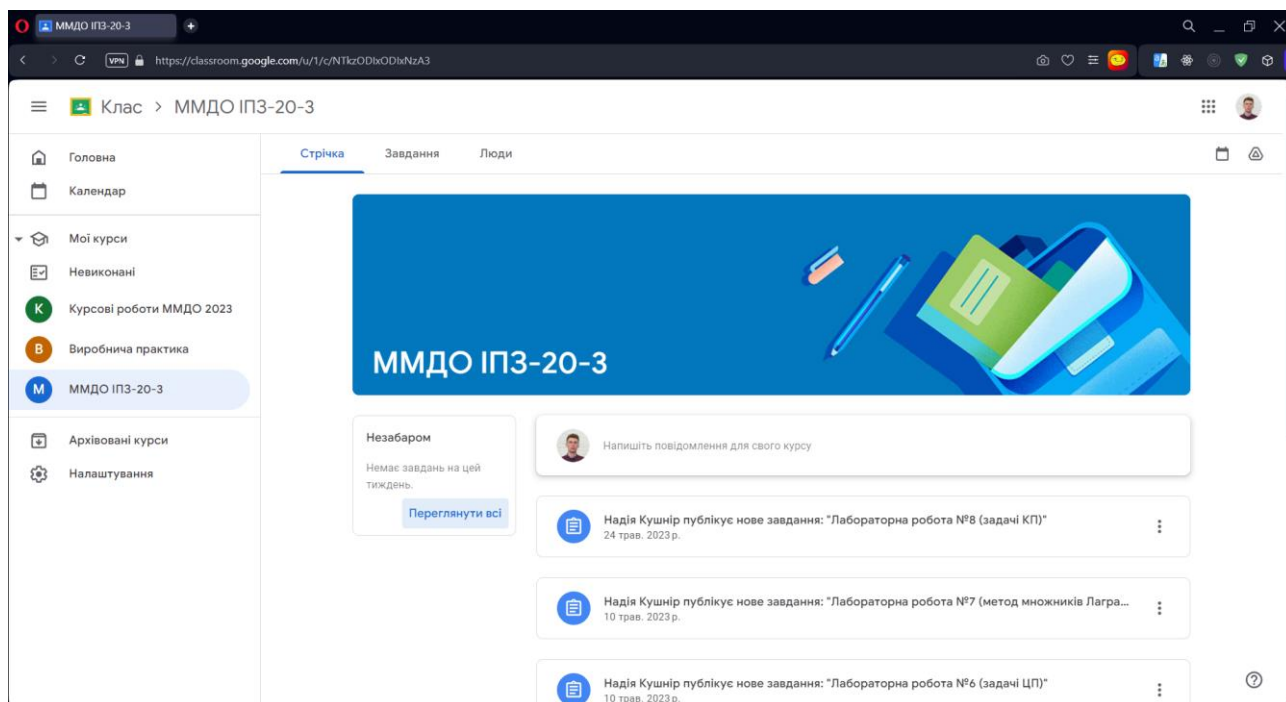


Рисунок 1.1 – Інтерфейс класу в Google Classroom

#### Переваги:

- викладачі можуть створювати курси, додавати студентів і ділитися різними навчальними матеріалами;
- над курсом можуть працювати декілька викладачів;
- можливість створення завдань та тестів;
- оцінювання виконаних студентами завдань і перегляд їх прогресу;
- викладач може залишати коментарі стосовно виконаного завдання;
- легка інтеграція з іншими інструментами Google для зручного обміну документами та іншими матеріалами.

#### Недоліки:

- обмежений набір навчальних засобів;
- використовувати можуть лише користувачі Google пошти.

## 2.2. Moodle

Moodle [5] (Modular Object-Oriented Dynamic Learning Environment, вимовляється «Мудл») – це модульне об'єктно-орієнтоване динамічне навчальне середовище, яке називають також системою управління навчанням (LMS),

		Цвік О. С.			Житомирська політехніка.24.121.19.080 - ПП	Арк.
		Кравченко С. М.				10
Змн.	Арк.	№ докум.	Підпис	Дата		

системою управління курсами (CMS), віртуальним навчальним середовищем (VLE) або просто платформою для навчання, яка надає викладачам, учням та адміністраторам дуже розвинутий набір інструментів для комп'ютеризованого навчання, в тому числі дистанційного. Moodle можна використовувати в навчанні школярів, студентів, при підвищенні кваліфікації, бізнес-навчанні, як в комп'ютерних класах навчального закладу, так і для самостійної роботи вдома [12].

Moodle надає багато різного функціоналу для створення онлайн-курсів та взаємодії зі студентами. Вона широко використовується багатьма закладами вищої та середньої освіти, адже є відмінним помічником в організації дистанційного навчання. Однак, на відміну від Google Classroom, ця платформа вимагає більш серйозного підходу і більш глибокого вивчення інструментів роботи.

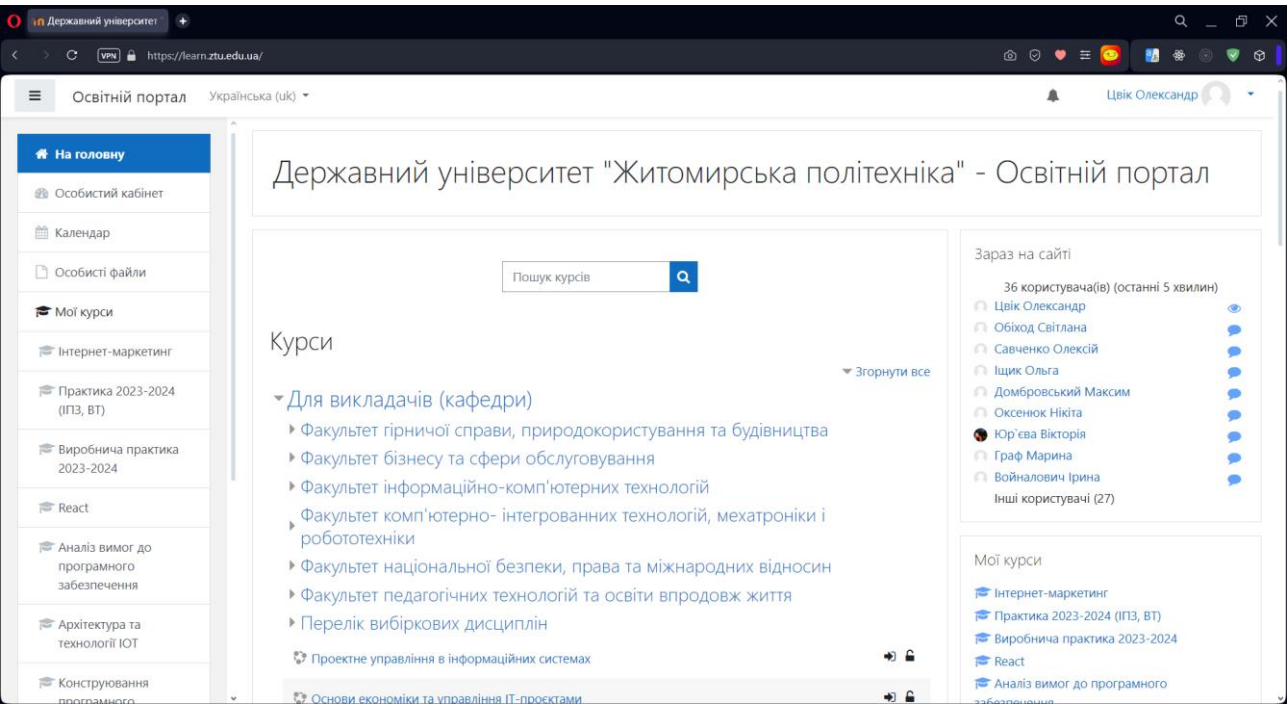


Рисунок 1.2 – Інтерфейс Moodle налаштований для Житомирської політехніки

Moodle повністю безкоштовна платформа, яку можна вільно завантажувати, встановлювати та змінювати. Вона відноситься до Open Source систем, тобто системам з відкритим вихідним кодом, що дозволяє багатьом програмістам створювати додаткові, дуже корисні розширення або модулі.

		Цвік О. С.			Житомирська політехніка.24.121.19.080 - ПП	Арк.
		Кравченко С. М.				11
Змн.	Арк.	№ докум.	Підпис	Дата		

#### Переваги:

- створення курсів та наповнення різноманітними матеріалами;
- можливість публікації навчального контенту різного формату – аудіо, відео, текст, різні формати файлів, каталоги тощо;
- велика кількість інструментів для управління курсами;
- містить багато функціоналу для створення різноманітних тестів;
- містить налаштування варіантів керування доступом користувачів до курсу та розміщених матеріалів;
- відстеження прогресу студентів;
- наявність особистого кабінету, який містить всі дані про користувача;
- користувачі мають можливість обмінюватися повідомленнями;
- перегляд активних користувачів на поточній сторінці;
- відкритий вихідний код.

Оскільки Moodle програма з відкритим вихідним кодом, платформа має велику кількість плагінів та доповнень до системи. Такі доповнення як правило безкоштовні, їх можна просто завантажити і встановити для своєї системи.

#### Прикладами таких плагінів є:

- модулі відеоконференції;
- аудіо та відео чати;
- масова розсилка повідомлень;
- засоби проєктної роботи;
- налаштування зовнішнього вигляду курсів;
- різні інтерактивні елементи гейміфікації та ігри (кросворд, вікторина, мільйонер, sudoku та інші);
- формування електронного портфоліо.

#### Недоліки:

- не всі функції є інтуїтивно зрозумілими;
- потрібно багато часу на засвоєння роботи з системою.

		Цвік О. С.			Житомирська політехніка.24.121.19.080 - ПП	Арк.
		Кравченко С. М.				12
Змн.	Арк.	№ докум.	Підпис	Дата		

## 2.3. Classdojo

Classdojo [2] - це сервіс, який допоможе забезпечити комунікацію між вчителями, учнями та їхніми батьками. Дана платформа дає змогу максимально відтворити шкільне середовище вдома й намагається зацікавити дітей молодшого і середнього шкільного віку за допомогою гейміфікації та заохочувальних бейджів. Кожен учень на платформі має свою власну анімовану аватарку, яка радіє, коли отримано похвалу від учителя і сумує, коли ставлять негативну оцінку.

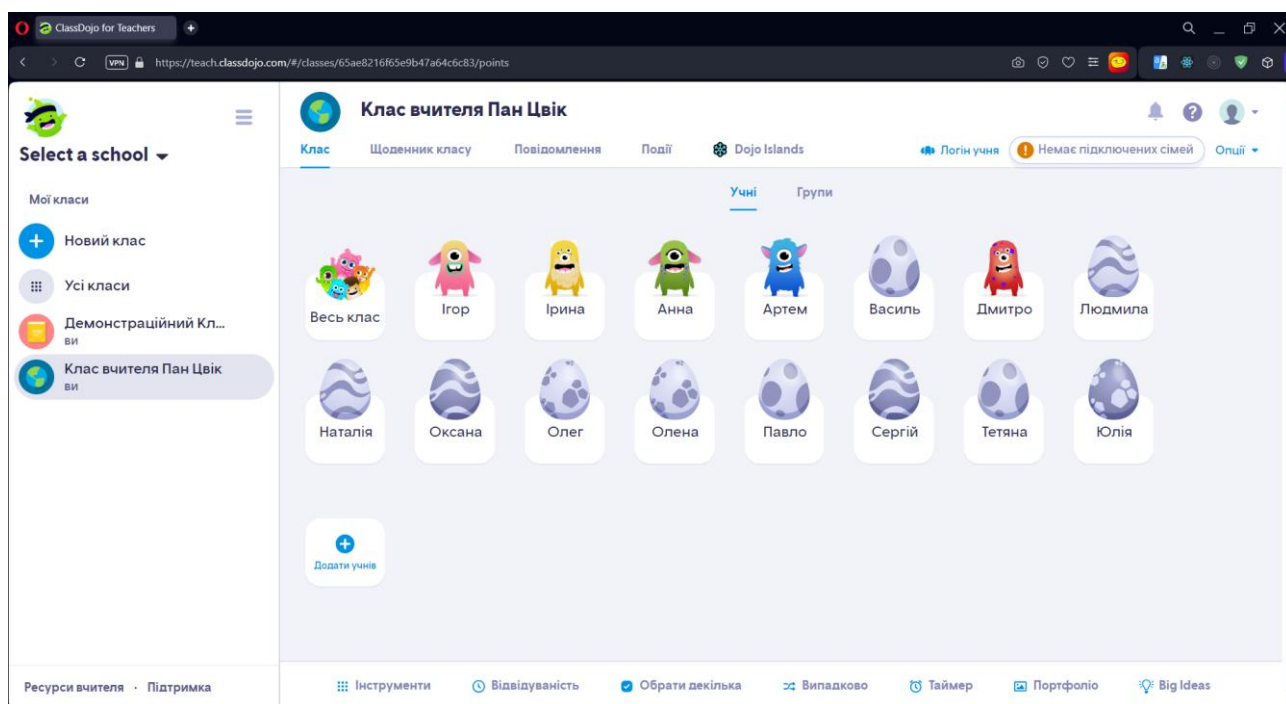


Рисунок 1.3 – Інтерфейс класу в Classdojo

Переваги:

- проста і зрозуміла реєстрація;
- легко запрошувати учнів і їх батьків, а також взаємодіяти з ними;
- яскрава анімація для дітей, яка привертає увагу;
- стрічка подій для інформування класу;
- бейджи з різними зображеннями для позитивних та негативних оцінок;
- генерація статистики прогресу кожного учня і всього класу для обраного періоду;
- батьки можуть переглядати успішність їхньої дитини.

		Цвік О. С.			Житомирська політехніка.24.121.19.080 - ПП	Арк.
		Кравченко С. М.				13
Змн.	Арк.	№ докум.	Підпис	Дата		

Недоліки:

– простий набір функцій (відсутність тестів, керування доступом до ресурсів тощо).

### **Висновки до другого розділу**

Після аналізу різних систем електронного навчання, представлених на ринку, було визначено їхні ключові функціональні можливості. Проаналізовано зовнішній вигляд інтерфейсів та перераховано переваги та недоліки подібних програмних продуктів.

Кожен з розглянутих продуктів відзначається своїми унікальними особливостями та функціональними можливостями. Важливо відзначити, що система з простим та зрозумілим інтерфейсом є ефективним інструментом для дистанційного навчання. Крім того, важливою рисою є гнучке налаштування, що включає в себе можливість зміни мови інтерфейсу. Впровадження елементів гейміфікації позитивно впливає на всіх учасників навчального процесу, адже стимулює їх бути більш активними.

		Цвік О. С.			Житомирська політехніка.24.121.19.080 - ПП	Арк.
		Кравченко С. М.				14
Змн.	Арк.	№ докум.	Підпис	Дата		

### РОЗДІЛ 3. ВИЗНАЧЕННЯ АРХІТЕКТУРИ ВЕБДОДАТКУ

Ефективним підходом для створення даного проєкту є використання клієнт-серверної архітектури. Модель такої системи полягає в тому, що клієнт відправляє запит на сервер, де він обробляється, і далі готовий результат відправляється клієнтові. Ця архітектура розділяє функціональні обов'язки між клієнтом та сервером, що робить її ідеальним вибором для створення ефективної, гнучкої та легко масштабованої системи.

Функції, які реалізуються на сервері:

- зберігання, доступ, захист і резервне копіювання даних;
- обробка клієнтського запиту;
- відправлення результату (відповіді) клієнту.

Функції, які реалізуються на стороні клієнта:

- надання користувальницького інтерфейсу;
- формулювання запиту до сервера і його відправка;
- отримання та відображення результатів запиту.

Діаграма розгортання призначена для аналізу апаратної частини системи. За допомогою даної діаграми можна провести аналіз необхідної апаратної конфігурації, на якій працюватимуть окремі процеси системи, і описати їх взаємодію між собою і іншими апаратними пристроями.

Вузол (англ. node) – це фізичний компонент системи, який має певні обчислювальні ресурси. Наприклад, вузлом може бути сервер, комп'ютер, пристрій зберігання даних або хмарна платформа тощо.

Крім вузлів на діаграмі розгортання вказуються відносини між ними. В якості відносин виступають фізичні з'єднання між вузлами і залежності між вузлами і компонентами, зображення яких теж можуть бути присутніми на діаграмах розгортання. З'єднання є різновидом асоціації і зображуються відрізками ліній без стрілок. Наявність такої лінії вказує на необхідність організації фізичного каналу для обміну інформацією між відповідними вузлами.

		Цвік О. С.			Житомирська політехніка.24.121.19.080 - ПП	Арк.
		Кравченко С. М.				15
Змн.	Арк.	№ докум.	Підпис	Дата		

Характер з'єднання може бути додатково специфікований приміткою, поміченим значенням або обмеженням.

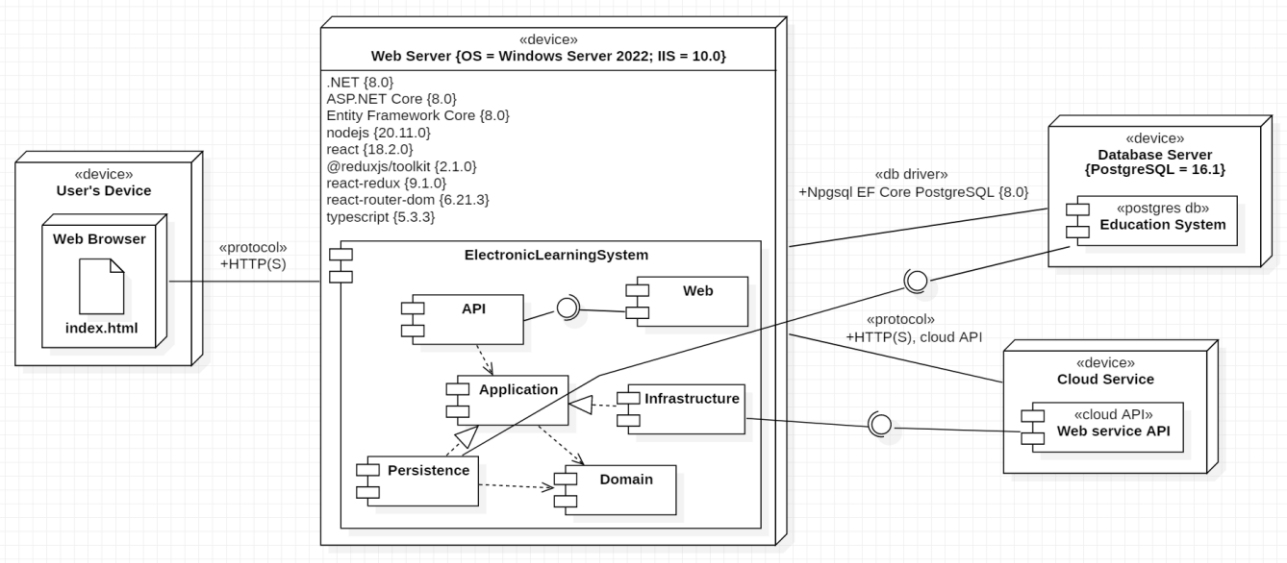


Рисунок 3.1 – Діаграма розгортання

Діаграми розгортання та діаграми компонентів можна зображувати окремо, також допускається поміщати діаграму компонентів в діаграму розгортання. Це доцільно робити, щоб показати які компоненти виконуються і на яких вузлах. У випадку діаграми на рис. 3.1, щоб уникнути збільшення її складності, вважаємо, що компонент ElectronicLearningSystem відображає діаграму компонентів зображену на рис. 3.3.

Система складається з чотирьох основних вузлів (див. рис. 3.1):

- User's Device – будь-який пристрій з якого користувач взаємодіє з системою (наприклад, комп'ютер, смартфон тощо).
- Web Server – сервер, на якому розміщується розроблений програмний код системи.
- Database Server – сервер, на якому розміщена база даних системи.
- Cloud Service – хмарний сервіс, на якому зберігаються файли різних типів.

Web Server можна розгорнути на віртуальній машині в хмарі. Це дозволяє масштабувати систему відповідно до попиту та забезпечити доступ до системи з будь-якого місця. Web Server взаємодіє з компонентом ElectronicLearningSystem,



тобто з кодом всієї системи, для отримання доступу до функціональності системи. Також, він може взаємодіяти з базою даних та хмарним сервісом для зберігання та отримання даних та файлів відповідно.

База даних Education System розгортається у виділеному сервері (Database Server), це дозволяє забезпечити надійність та продуктивність зберігання даних.

Користувачі можуть взаємодіяти з системою за допомогою User's Device через браузер (Web Browser) та артефакт (англ. artifact) index.html, тобто вебінтерфейс, який реалізований та можна отримати від вебсервера.

Для проектування системи було використано основні принципи «чистої архітектури» для розділення програмного забезпечення на різні рівні. Головним правилом, що приводить цю архітектуру в дію, є правило залежностей (англ. Dependency Rule): залежності у вихідному коді мають бути спрямовані лише всередину (див. рис. 3.2) [9, с. 215].

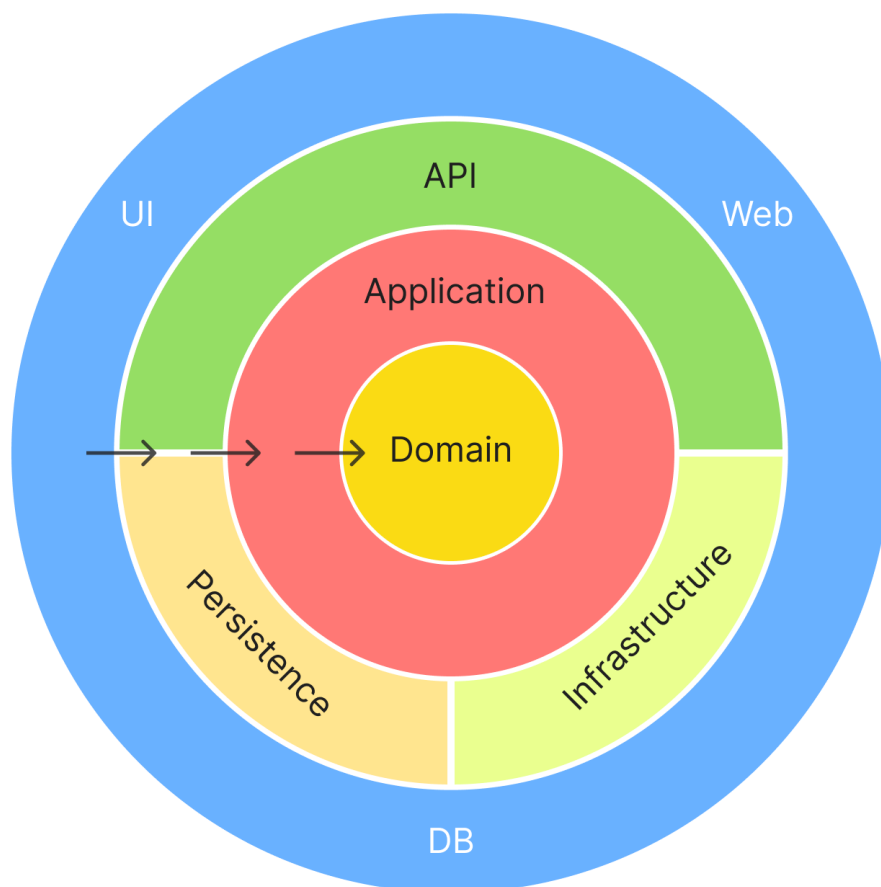


Рисунок 3.2 – Схема слоїв

Усе, що перебуває у внутрішньому колі, взагалі не може нічого знати про зовнішні кола. Наприклад, імена, оголошені в зовнішніх колах, не повинні згадуватися в коді, що міститься у внутрішніх колах. Сюди належать функції, класи, змінні та будь-які інші іменовані елементи програми.

Так само формати даних, оголошені у зовнішніх колах, не повинні використовуватися у внутрішніх, особливо якщо ці формати генеруються фреймворком у зовнішньому колі. Ніщо у зовнішньому колі не мусить впливати на внутрішні кола [9, с. 215-216].

Усі головні компоненти системи чітко розділені та інкапсульовані, утворюючи кілька шарів з односторонніми залежностями:

1. Domain. Компонент Domain – це фундамент системи, навколо нього будується весь проект. Цей шар немає жодних залежностей від зовнішніх і містить лише сутності, перерахування, константні дані і, можливо, деякі спеціальні винятки на рівні Domain.

2. Application. Разом із доменним рівнем (Domain) прикладний рівень (Application) утворює ядро рішення, яке повинно мати можливість працювати та забезпечувати бізнес-логіку незалежно від зовнішніх рівнів і залежати виключно від Domain. Він створює об'єкти сутностей, виконує бізнес-правила та повертає результати для API.

3. Infrastructure. Це рівень, у якому має бути реалізована вся логіка зв'язку із зовнішніми системами, як-от надсилання електронних листів, зв'язок із стороннім API тощо. Він залежить лише від Application та реалізує його інтерфейси.

4. Persistence. Порівняно з Infrastructure, рівень Persistence також містить логіку для зв'язку із зовнішніми системами, але його конкретна мета – зв'язуватися з базами даних. Цей рівень залежить від Application та Domain.

5. API. Це рівень взаємодії із зовнішнім світом, який дозволяє клієнтам отримувати видимі результати після запиту даних. Цей рівень може бути у будь-якій формі, наприклад: веб-API, консольна програма, графічний інтерфейс тощо. Він також залежить лише від Application, але також може використовувати

		Цвік О. С.			Житомирська політехніка.24.121.19.080 - ПП	Арк.
		Кравченко С. М.				18
Змн.	Арк.	№ докум.	Підпис	Дата		

елементи з Persistence. Наприклад, при першому запуску програми потрібно завантажити початкові дані (англ. seed data) в БД, які зберігаються в Persistence. Він не містить бізнес-логіки, його єдине завдання – перетворити запити користувача на зрозумілу для Application форму та повернути отримані результати у зручному форматі для компонента Web.

6. Web. Шар Web на самому верхньому рівні та відповідає за реалізацію клієнтської частини, яка може включати в себе HTML, CSS, JavaScript та будь-які інші фронтенд технології для створення вебінтерфейсу. Користувачі можуть використовувати даний компонент для взаємодії з системою за допомогою HTTP запитів.

Використовуючи клієнт-серверну архітектуру, клієнтську частину, за яку відповідає компонент Web, варто реалізувати у вигляді односторінкового додатку (англ. Single Page Application, SPA). SPA – це вебдодаток, який завантажує сторінку лише один раз та динамічно змінює її вміст без необхідності перезавантаження всієї сторінки. SPA використовує підхід AJAX (англ. Asynchronous JavaScript and XML) для обміну даними з сервером та оновлення лише необхідних частин сторінки. SPA забезпечує такі переваги, як:

1. Постійна взаємодія з користувачем, за допомогою динамічної зміни контенту й без завантаження нової сторінки з сервера. Це дає змогу зменшити навантаження на сервер, адже серверу достатньо відправити дані у JSON форматі, а не генерувати всю HTML сторінку.

2. При завантаженні нових модулів (сторінок) контент на них оновлюється лише частково, тому що немає необхідності повторно завантажувати елементи, які не було змінено.

3. Розробникам доступні різні фреймворки та бібліотеки, які спрощують створення архітектури проєкту та надають чимало готових елементів для роботи.

Односторінкові додатки мають і деякі недоліки, які можна виправити шляхом вивчення сторонніх інструментів та проведення додаткової оптимізації:

1. Проблеми з індексацією динамічного контенту, тобто складнощі з налаштуванням SEO.

		Цвік О. С.			Житомирська політехніка. 24.121.19.080 - ПП	Арк.
		Кравченко С. М.				19
Змн.	Арк.	№ докум.	Підпис	Дата		

2. Тривале завантаження під час першого відвідування, через велику кількість скриптів, які генерують контент на сторінці.

3. Важливо, щоб у користувачів було ввімкнено підтримку JavaScript у браузері.

Розуміння точної поведінки системи, яку надає кожна частина програмного забезпечення, є важливою для будь-якого програміста. Діаграми компонентів є потужним інструментом, який може допомогти в цьому.

Діаграми компонентів здатні допомогти іншим розробникам зрозуміти структуру конкретної системи, також ці діаграми можуть описувати програмні системи, реалізовані будь-якою мовою чи стилем програмування.

Головна мета цих діаграм – показати взаємозв'язок між різними компонентами в системі. Тут «компоненти» можуть означати модулі класів, які представляють незалежні системи або підсистеми з можливістю взаємодії з рештою системи, також у ролі компонента можуть виступати файли, бібліотеки, модулі, виконувані файли, пакети тощо.

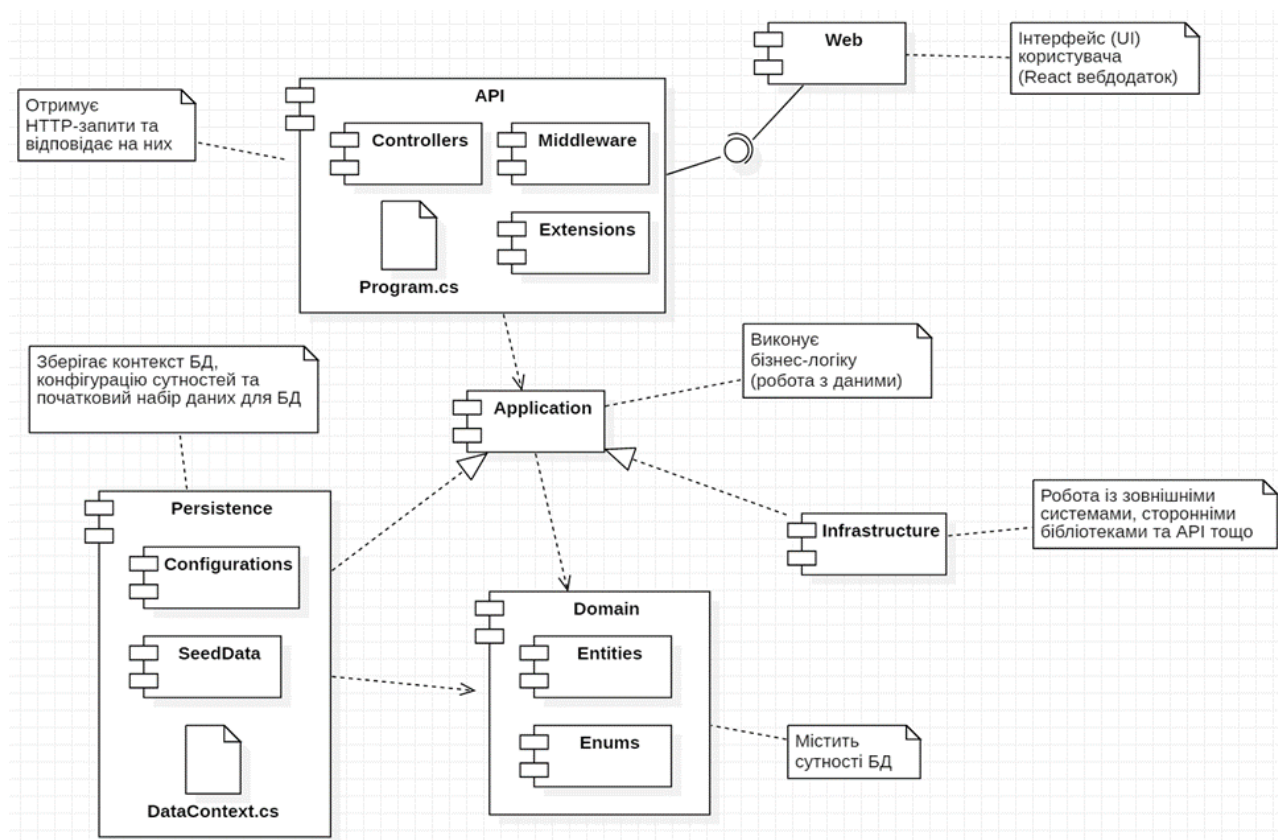


Рисунок 3.3 – Діаграма компонентів

Діаграма компонентів зображена на рис. 3.3 показує загальну структуру системи електронного навчання. Для зручності сприйняття діаграми компонентів, до кожного компонента було додано відповідні нотатки.

Компонент «Web» зв'язаний з компонентом «API» через інтерфейс тому, що взаємодія між ними відбувається через певний набір методів HTTP запитів. Ці методи визначають спосіб передачі даних та обмін інформацією між компонентами, що дозволяє взаємодіяти без прив'язки до конкретних реалізацій методів.

Компонент «API» використовує компонент «Application» для доступу до даних та основної функціональності системи. В свою чергу «Application» використовує «Domain», де зберігаються класи сутностей, що відповідають таблицям в базі даних. Компоненти «Persistence» та «Infrastructure» реалізують різні інтерфейси компонента «Application» для роботи з контекстом БД та зовнішніми системами відповідно. Також «Persistence» використовує сутності з «Domain» для того, щоб провести конфігурацію БД.

### Висновки до третього розділу

Обрано клієнт-серверну архітектуру для системи, в якій клієнтську частину буде реалізовано у вигляді односторінкового додатку (SPA), а сервер буде використовувати «чисту архітектуру» Роберта Мартіна.

Використання SPA дозволить забезпечити користувачам зручний, динамічний та швидкий інтерфейс, зменшуючи навантаження на сервер та забезпечуючи більш ефективну обробку запитів. Застосування «чистої архітектури» на сервері дозволить розділити компоненти системи на незалежні шари, що полегшить тестування, розширення та підтримку коду.

		Цвік О. С.			Житомирська політехніка.24.121.19.080 - ПП	Арк.
		Кравченко С. М.				21
Змн.	Арк.	№ докум.	Підпис	Дата		

## РОЗДІЛ 4. ОБҐРУНТУВАННЯ ВИБОРУ ІНСТРУМЕНТАЛЬНИХ ЗАСОБІВ ТА ВИМОГИ ДО АПАРАТНОГО ЗАБЕЗПЕЧЕННЯ

Для серверної частини обрано **ASP.NET Core** – це кросплатформний фреймворк для створення вебзастосунків на платформі .NET з відкритим вихідним кодом, де в якості мови програмування можна використовувати C#. Має всі необхідні інструменти для роботи сучасного вебдодатка: маршрутизація, конфігурація, логування, можливість роботи з різними системами баз даних, налаштування авторизації, інтеграція зі сторонніми API та інші. Він пропонує ефективний підхід до розробки програми, спеціально орієнтований на роботу в стилі REST (англ. Representation State Transfer, передача стану уявлення).

Проектування бази даних буде виконано за допомогою ORM (англ. object-relational mapping, реляційне відображення об'єктів), тобто відображення даних на реальних об'єктах, **Entity Framework Core** (EF Core) та підходу Code First. Тобто спочатку потрібно написати код на мові програмування C#, а потім з цього коду буде автоматично згенеровано таблиці для бази даних. Для цього підходу дуже важливо визначити класи сутностей, що будуть відповідати таблицям, які будуть зберігатися в базі даних.

Конфігурацію таблиць зручно налаштовувати за допомогою Fluent API, адже він надає більший діапазон параметрів конфігурації, ніж атрибути анотації даних. Конфігурація Fluent API також сприяє чистішому коду, оскільки конфігурацію можна зберігати в окремих файлах.

EF Core підвищує ефективність та швидкість розробки, оскільки тепер програмісти мають справу не з таблицями та полями, а з класами та об'єктами, що дозволяє не писати складних SQL-запитів. EF Core є одним з популярних ORM-фреймворків для .NET розробників, адже підтримує різні СУБД, включаючи PostgreSQL, MySQL, SQLite, Oracle, MongoDB та інші.

Бібліотека **MediatR** – спрощує реалізацію патернів CQRS та Mediator. Завдяки їй, можна вдосконалити розділення обов'язків та зробити код більш гнучким та масштабованим.

		Цвік О. С.			Житомирська політехніка.24.121.19.080 - ПП	Арк.
		Кравченко С. М.				22
Змн.	Арк.	№ докум.	Підпис	Дата		

**FluentValidation** – це бібліотека .NET для створення типізованих правил валідації даних. Допомагає уникнути помилок та виконання некоректних операцій з даними.

Обрано **PostgreSQL** як об'єктно-реляційну систему керування базами даних, адже вона є однією з найпотужніших та надійних СУБД з відкритим кодом. Має високу продуктивність, підтримує багато функцій та може бути використана в різних проектах – від невеликих вебдодатків до складних корпоративних систем. Забезпечує масштабованість та надійність, тобто практично необмежене зростання обсягів зберігання, що чудово підходить для створення системи електронного навчання, адже майже щодня будуть створюватися нові записи про відвідуваність та успішність студентів.

Для реалізації клієнтського додатку обрано бібліотеку **React** разом з мовою програмування **TypeScript**, адже строга типізація призводить до написання більш надійного коду та швидкої розробки. **Redux Toolkit** для керування станом додатку й **RTK Query** для відправки запитів до сервера. Бібліотеку компонентів **Ant Design** для стилізації компонентів, а також додатково бібліотеку **date-fns** - для роботи з датами та часом, й відображення їх на сторінці у різних форматах.

Redux є популярною бібліотекою управління станом для React додатків. Його використання дозволяє легко управляти станом додатка, адже Redux зберігає стан додатку в одному об'єкті, який можна легко оновлювати за допомогою виклику функції `dispatch`. Особливо його зручно використовувати, коли однаковим та актуальним станом потрібно користуватися у різних компонентах або коли є необхідність у взаємодії між декількома компонентами.

Наприклад, у стані додатка можна зберігати інформацію про автентифікованого користувача, для того щоб відкривати лише ті частини сайту до яких користувач має доступ. При першому завантаженні сторінки відбувається запит до сервера для отримання даних про поточного користувача, поки користувач чекає можна показати індикатор завантаження, компонент з бібліотеки Ant Design.

		Цвік О. С.			Житомирська політехніка.24.121.19.080 - ПП	Арк.
		Кравченко С. М.				23
Змн.	Арк.	№ докум.	Підпис	Дата		

Для навігації та маршрутизації в React додатках використовується бібліотека **React Router** (react-router-dom). Вона дозволяє легко визначати шляхи, обробляти параметри URL і ефективно керувати навігацією між сторінками. Має можливість гнучкого налаштування перенаправлення користувачів на інші сторінки, якщо вони намагаються перейти до неіснуючої сторінки або сторінки, яка вимагає авторизації.

Щоб покращити якість написання коду варто використати **Prettier** та **ESLint** – це інструменти, які допомагають дотримуватися загальних стандартів у процесі розробки програмного забезпечення. Prettier використовується для форматування коду, працює за допомогою правил, які можна налаштувати відповідно до потреб проєкта. ESLint допомагає виявити помилки у коді, підтримує широкий спектр правил, включаючи правила стилю, продуктивності та інші.

У якості середовища розробки обрано **Visual Studio Code** (VS Code). Цей редактор швидко аналізує структуру коду, показує підказки під час написання коду, підтримує автодоповнення і рефакторинг, що сприяє уникненню помилок та прискорює процес розробки. Має великий список доступних розширень, які дозволяють зручно працювати з великим переліком технологій, таких як C#, JavaScript, TypeScript, HTML, CSS тощо. Крім того, за допомогою VS Code можна легко та зручно відлагоджувати власний код.

### Вимоги до апаратного забезпечення

#### 1. Вимоги до середовища виконання:

- 1.1. Наявність стабільного інтернет-з'єднання для взаємодії з серверними та іншими віддаленими ресурсами.
- 1.2. Комп'ютер на якому буде розгортатися система повинен підтримувати та виконувати вимоги наступних технологій:
  - .NET 8.0;
  - Node.js 20.11.0;
  - React 18.2.0.

		Цвік О. С.			Житомирська політехніка.24.121.19.080 - ПП	Арк.
		Кравченко С. М.				24
Змн.	Арк.	№ докум.	Підпис	Дата		



2. Вимоги до СУБД та доступу до даних:

2.1. Для забезпечення надійного та ефективного доступу до даних система має використовувати реляційну СУБД PostgreSQL 16.1.

3. Вимоги до клієнтського пристрою:

3.1. Наявність стабільного інтернет-з'єднання.

3.2. Сумісність з будь-яким браузером, який підтримує виконання JavaScript, наприклад Opera, Chrome, Firefox, Microsoft Edge тощо.

### **Висновки до четвертого розділу**

Було обрано основні інструменти для розробки серверної частини, такі як ASP.NET Core, Entity Framework Core, MediatR та FluentValidation. У якості СУБД обрано PostgreSQL, а Visual Studio Code як середовище розробки.

Визначено головні технології для реалізації клієнтського додатку: React, TypeScript, Redux Toolkit, RTK Query, React Router, Ant Design, date-fns, Prettier та ESLint.

Також було сформовано мінімальні вимоги до апаратного забезпечення для справної роботи системи.

		Цвік О. С.			Житомирська політехніка.24.121.19.080 - ПП	Арк.
		Кравченко С. М.				25
Змн.	Арк.	№ докум.	Підпис	Дата		

## ВИСНОВКИ

На основі аналізу вимог до програмного продукту, було визначено завдання, які необхідно вирішити для успішної реалізації проєкту.

Було проведено порівняльний аналіз аналогічних програмних продуктів, з урахуванням їх переваг та недоліків. Кожна з розглянутих систем має свої унікальні особливості та функціональні можливості. Визначено, що система з простим та зрозумілим інтерфейсом, гнучким налаштуванням і елементами гейміфікації є ефективним інструментом для дистанційного навчання.

Обрано клієнт-серверну архітектуру з використанням SPA для клієнтської частини, даний підхід дозволить забезпечити зручний та ефективний інтерфейс для користувачів. Застосування «чистої архітектури» на сервері спростить тестування та підтримку коду.

Визначено основні інструменти для розробки як серверної, так і клієнтської частини, зокрема ASP.NET Core, React, TypeScript, PostgreSQL, які забезпечать стабільність та ефективність розробки.

Сформовано мінімальні вимоги до апаратного забезпечення для справної роботи системи, що дозволить користувачам з різних пристроїв отримувати доступ до навчального контенту без перешкод.

Визначені завдання, архітектура та інструменти для реалізації системи мають практичну цінність для подальшої реалізації проєкту. Для майбутнього вдосконалення програмного комплексу можна впроваджувати нові технології, такі як машинне навчання та штучний інтелект, для підвищення ефективності системи, або додавати нові елементи гейміфікації для стимулювання активності та зацікавленості користувачів у навчальному процесі.

		Цвік О. С.			Житомирська політехніка.24.121.19.080 - ПП	Арк.
		Кравченко С. М.				26
Змн.	Арк.	№ докум.	Підпис	Дата		

## ЛІТЕРАТУРА

1. 5 платформ для організації дистанційного навчання [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: <https://buki.com.ua/news/5-platform-dlya-orhanizatsiyi-dystantsiynoho-navchannya/>.
2. Classdojo [Електронний ресурс] – Режим доступу до ресурсу: <https://www.classdojo.com/uk-ua/>.
3. Component Diagram Tutorial [Електронний ресурс] – Режим доступу до ресурсу: <https://www.lucidchart.com/pages/uml-component-diagram>.
4. Google Classroom [Електронний ресурс] – Режим доступу до ресурсу: <https://classroom.google.com/>.
5. Moodle [Електронний ресурс] – Режим доступу до ресурсу: <https://moodle.org/>.
6. SPA у програмуванні: занурення у світ односторінкових додатків [Електронний ресурс]. – 2023. – Режим доступу до ресурсу: <https://foxminded.ua/spa-u-prohramuvanni/>.
7. Діаграма розгортання [Електронний ресурс] – Режим доступу до ресурсу: [https://vuzlit.com/1009782/diagrama\\_rozgortannya](https://vuzlit.com/1009782/diagrama_rozgortannya).
8. Клієнт-серверна архітектура [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: <https://training.qatestlab.com/blog/technical-articles/client-server-architecture/>.
9. Мартін Р. Чиста архітектура: Мистецтво створення програмного забезпечення / Роберт Мартін. – Харків: Фабула, 2019. – 368 с. – (Видання друге).
10. Моралес Д. Повне розуміння діаграми компонентів UML за допомогою легкого методу [Електронний ресурс] / Джейд Моралес. – 2023. – Режим доступу до ресурсу: <https://www.mindonmap.com/uk/blog/uml-component-diagram/>.
11. Сервіси та платформи для дистанційного навчання [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: [https://24tv.ua/tech/navchannya-vdoma-spisok-program-platform-dlya-distantsiynogo-navchannya\\_n1416110](https://24tv.ua/tech/navchannya-vdoma-spisok-program-platform-dlya-distantsiynogo-navchannya_n1416110).

		Цвік О. С.			Житомирська політехніка. 24.121.19.080 - ПП	Арк.
		Кравченко С. М.				27
Змн.	Арк.	№ докум.	Підпис	Дата		

12. Що таке Moodle [Електронний ресурс]. – 2023. – Режим доступу до ресурсу: <https://moodle.org/mod/page/view.php?id=8174>.

13. Що таке система дистанційного навчання? [Електронний ресурс]. – 2024. – Режим доступу до ресурсу: <https://softbook.app/blog/uk/shho-take-sistema-distancijnogo-navchannya/>.

		Цвік О. С.			Житомирська політехніка.24.121.19.080 - ПП	Арк.
		Кравченко С. М.				28
Змн.	Арк.	№ докум.	Підпис	Дата		