

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: Розробка веб-платформи для навчання сліпого набору тексту
на клавіатурі з використанням технології Django

Виконав: студент IV курсу, групи СП-42
спеціальності 121 Інженерія програмного
забезпечення

(шифр і назва спеціальності)

Кшивак Н.І.
(підпис) (прізвище та ініціали)

Керівник Петрик М.Р.
(підпис) (прізвище та ініціали)

Нормоконтроль Стоянов Ю.М.
(підпис) (прізвище та ініціали)

Завідувач кафедри . Петрик М.Р.
(підпис) (прізвище та ініціали)

Рецензент Луцик Н.С.
(підпис) (прізвище та ініціали)

3. Практична частина розробки веб-платформи для навчання сліпого набору тексту на клавіатурі. 4. Безпека життєдіяльності, основи охорони праці. Висновки. Перелік джерел. Додатки.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи охорони праці	К.т.н.б доцент кафедри МТ Лазарюк Валерій Володимирович		
Нормконтроль			

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи		
2.	Аналіз вимог до розроблюваного проекту		
3.	Опрацювання джерел по темі кваліфікаційної роботи		
4.	Виконання дослідження щодо існуючих інструментів та технологій		
5.	Проектування та розробка веб-платформи для навчання сліпого набору тексту на клавіатурі		
6.	Проведення тестування розробленої веб-платформи		
7.	Оформлення пояснювальної записки по виконаному завданні		
8.	Виконання завдання до підрозділу «Безпека життєдіяльності»		
9.	Виконання завдання до підрозділу «Основи охорони праці»		
10.	Оформлення кваліфікаційної роботи		
11.	Нормоконтроль		
12.	Перевірка на плагіат		
13.	Попередній захист кваліфікаційної роботи		
14.	Оформлення презентації та доповіді по кваліфікаційній роботі		
15.	Захист кваліфікаційної роботи		

Студент

_____ (підпис)

Кшивак Н.І.

(прізвище та ініціали)

Керівник роботи

_____ (підпис)

Петрик М.Р.

(прізвище та ініціали)

АНОТАЦІЯ

Розробка веб-платформи для навчання сліпого набору тексту на клавіатурі з використанням технології Django // Кваліфікаційна робота освітнього рівня «Бакалавр» // Кшивак Назарій Ігорович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра програмної інженерії, група СП-42 // Тернопіль, 2025 // С. - 60, рис - , табл. - , додат. - , бібліогр.- ,

Ключові слова: сліпий набір, система обліку користувачів, веб-аналітика, багатомовність, фреймворк Django, тестування швидкості друку

Кваліфікаційна робота присвячена розробці веб-платформи для навчання сліпого набору тексту на клавіатурі.

В першому розділі кваліфікаційної роботи описано актуальність теми та основні цілі дослідження. Висвітлено основні методи та підходи для створення веб-платформи. Розглянуто сучасні інструменти та технології для автоматизації створення програми. Проаналізовано вимоги до програмних засобів.

В другому розділі кваліфікаційної роботи наведено проєктування архітектури веб-платформи. Досліджено алгоритми для автоматизованого формування даних для користувачів.

В третьому розділі кваліфікаційної роботи описано реалізацію програмно-алгоритмічних засобів для автоматизації обробки даних. Проаналізовано результати тестування та ефективність розроблених засобів. Проведено аналіз точності та надійності роботи веб-платформи.

Об'єкт дослідження: веб-платформа для навчання сліпого набору тексту на клавіатурі.

Предмет дослідження: процес створення сайту за допомогою технології Django.

ABSTRACT

Development of a Web Platform for Learning Touch Typing Using Django Technology // Qualification work of the educational level "Bachelor" // Nazariy Kshyvak// Ternopil Ivan Puluj National Technical University, Computer and Information Systems and Software Engineering Faculty, Software Engineering Department, group SP-42 // Ternopil, 2025 // Pages - 60, figures - , tables - , addition -, literature -

Keywords: touch typing, user account system, web analytics, multilingual support, Django framework, typing speed testing.

The qualification work is dedicated to the development of a web platform for learning touch typing on a keyboard.

The first section of the qualification work presents the relevance of the topic and the main objectives of the research. It highlights the key methods and approaches used in the development of the web platform. Modern tools and technologies for the automation of software development are reviewed. The requirements for the software tools are analyzed.

The second section of the qualification work provides the design of the web platform architecture. Algorithms for automated generation of user data are investigated.

The third section of the qualification work describes the implementation of software and algorithmic solutions for data processing automation. The results of testing and the efficiency of the developed tools are analyzed. The accuracy and reliability of the web platform are evaluated.

Object of the research: web platform for learning touch typing.

Subject of the research: the process of developing a website using the Django technology. the third section of the qualification work, the implementation of the software and algorithmic means for data processing automation is described. The results of testing and the efficiency of the developed tools are analyzed. A detailed analysis of the accuracy and reliability of the software-algorithmic tools is conducted.

ПЕРЕЛІК СКОРОЧЕНЬ І ТЕРМІНІВ

API (англ. Application Programing Interface) – прикладний програмний інтерфейс.

Python (укр. Пайтон) — високорівнева мова програмування, яку називають другою за популярністю в світі. Її використовують для розробки вебзастосунків, програмного забезпечення, машинного навчання.

Фреймворк – інфраструктура програмних рішень, що полегшує розробку складних систем.

Django (укр. Джанго) — високорівневий відкритий Python-фреймворк (програмний каркас) для розробки вебсистем.

Django Debug Toolbar — це потужний інструмент для налагодження Django-коду. Він надає детальну інформацію про процес виконання запитів, використання бази даних, налаштування та багато інших аспектів роботи Django.

Bootstrap – це спеціальний фреймворк, призначенням якого є створення сайтів та web-додатків. Він містить набір інструментів HTML та CSS.

HTTP (англ. HyperText Transfer Protocol) – протокол передачі гіпертексту.
FastAPI – сучасний, високопродуктивний веб-фреймворк для створення API за допомогою Python.

JSON (англ. JavaScript Object Notation) – нотація об'єкту JavaScript.

REST (англ. Representational State Transfer) – передача репрезентативного стану.

ЗМІСТ

ВСТУП.....	9
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАВДАННЯ ДО ВЕБ-ПЛАТФОРМИ НАВЧАННЯ СЛІПОГО НАБОРУ НА КЛАВІАТУРІ З ВИКОРИСТАННЯМ ТЕХНОЛОГІЇ DJANGO.....	11
1.1 Аналіз предметної веб-платформи для навчання сліпого друку.....	11
1.2 Формування списку обмежень до веб-платформи навчання сліпого друку.....	14
1.3 Актанти та варіанти використання веб-платформи навчання сліпого друку.....	16
1.4 Визначення переліку вимог до веб-платформи навчання сліпого друку.....	18
1.5 Висновок до першого розділу.....	20
РОЗДІЛ 2. ПРОЄКТНА ЧАСТИНА РОЗРОБКИ ВЕБ-ПЛАТФОРМИ ДЛЯ НАВЧАННЯ СЛІПОГО ДРУКУ.....	21
2.1 Проєктування архітектури веб-платформи для навчання сліпого друку	21
2.2 Проєктування системи аутентифікації та профілів.....	22
2.3 Проєктування сторінок навчання сліпого друку.....	26
2.4 Проєктування сторінок тестування та статистики.....	30
2.5 Висновок до другого розділу.....	34
РОЗДІЛ 3. ПРАКТИЧНА ЧАСТИНА РОЗРОБКИ -.....	36
3.1 Архітектура Django та базові компоненти системи.....	36
3.2 Реалізація системи аутентифікації та користувацьких профілів.....	38
3.3 Реалізація систем сторінок тестування та статистики	41
3.4 Додаткові реалізовані модулі системи.....	45
3.5 Висновок до третього розділу.....	47

РОЗДІЛ 4. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ	48
4.1 Надзвичайні ситуації технологічного характеру.....	48
4.2 Оцінка розробленого технологічного процесу щодо умов безпеки, втомлюваності та продуктивності праці.....	50
ВИСНОВКИ.....	53
ПЕРЕЛІК ДЖЕРЕЛ.....	55
ДОДАТКИ	

ВСТУП

Актуальність теми. У сучасному цифровому світі навичка швидкого та безпомилкового друку стає все більш необхідною для ефективної роботи з комп'ютером. Однак традиційні методи навчання «сліпому» друку часто є монотонними та недостатньо адаптивними, що знижує мотивацію користувачів.

Розробка спеціалізованої веб-платформи для навчання сліпому методу друку дозволить автоматизувати процес тренувань, забезпечити персоналізований підхід до кожного користувача та запобігти спробам шахрайства. Важливим аспектом є обробка даних у реальному часі, збереження достовірності результатів та масштабованість системи для обслуговування великої кількості користувачів.

Сучасні платформи часто не враховують усіх цих вимог, що призводить до неефективного навчання або можливості маніпулювання результатами. Тому створення інтерактивної, захищеної та адаптивної системи для вивчення сліпого друку є актуальним завданням.

Мета і задачі дослідження. Метою даної кваліфікаційної роботи освітнього рівня «Бакалавр» є розробка веб-платформи для навчання сліпому методу друку з інтелектуальною системою аналізу результатів та захистом від взлому. Для досягнення поставленої мети необхідно виконати такі завдання:

- Проаналізувати існуючі рішення для навчання сліпому друку та виявити їх недоліки.
- Сформулювати вимоги до функціоналу платформи, зокрема щодо захисту від читингу, обробки даних та адаптивного навчання.
- Спроекувати архітектуру системи, включаючи модулі реєстрації, статистики, перевірки результатів та зворотного зв'язку.
- Реалізувати веб-додаток з використанням сучасних технологій фронтенду та бекенду.

Виконання цих завдань дозволить створити ефективний інструмент для вивчення сліпого друку, який забезпечить точність результатів, зручність використання та захист від спроб обману.

Практичне значення одержаних результатів. Практична цінність роботи полягає в розробці веб-платформи, яка автоматизує процес навчання сліпому друку, забезпечує достовірність результатів та адаптується до рівня користувача. Система може бути використана:

- У навчальних закладах для підвищення цифрової грамотності студентів.
- В офісних середовищах для покращення продуктивності співробітників.
- Як особистий тренажер для всіх, хто бажає навчитися швидко друкувати.

Завдяки інтеграції механізмів захисту від читингу та аналізу даних у реальному часі, платформа забезпечить якісне навчання без можливості маніпуляцій.

Публікації. Основні результати дослідження опубліковано у тезах. У співавторстві, здобувачу освітнього рівня бакалавр належить: [6] – розробка веб-платформи для навчання сліпого набору тексту на клавіатурі з використанням технології Django.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАВДАННЯ ДО ВЕБ-ПЛАТФОРМИ ДЛЯ НАВЧАННЯ СЛІПОГО НАБОРУ ТЕКСТУ НА КЛАВІАТУРІ З ВИКОРИСТАННЯМ ТЕХНОЛОГІЇ DJANGO

1.1 Аналіз предметної області веб-платформи для навчання сліпому друку

Навчання сліпому методу друку (touch typing) – це процес формування навички швидкого набору тексту на клавіатурі без необхідності візуального контролю. Традиційні методики навчання часто ґрунтуються на монотонних вправах, що знижує мотивацію користувачів. Сучасні веб-платформи можуть автоматизувати цей процес, забезпечуючи адаптивне навчання, аналіз результатів у реальному часі та захист від спроб шахрайства.

Основні виклики при розробці платформи:

1. Обробка даних у реальному часі

- Потрібно фіксувати швидкість, точність, помилки та час реакції користувача.
- Система повинна оперативно аналізувати ці дані для формування персоналізованих завдань.

2. Захист від читингу (шахрайства)

- Виявлення аномальної швидкості (наприклад, > 700 символів за хвилину).
- Контроль взаємодії з інтерфейсом (заборона копіювання тексту, використання автозаповнення).
- Блокування сторонніх скриптів та інструментів, що можуть спотворити результати.

3. Адаптивне навчання

- Поділ клавіатури на зони (рядки, пальці) для поступового ускладнення завдань.
- Гнучка система рекомендацій на основі статистики користувача.

4. Масштабування та продуктивність

- Обробка великої кількості одночасних сесій.
- Використання кешування для швидкого доступу до даних.

Існуючі рішення та їх недоліки:

Багато сучасних тренажерів сліпого друку мають такі проблеми:

- Відсутність ефективного захисту від читингу.
- Обмежена адаптивність (однакові вправи для всіх рівнів).
- Низька продуктивність при збільшенні навантаження.

Запропонований підхід

Розробка платформи на основі Django дозволить:

- Реалізувати надійну систему аутентифікації та збереження статистики.
- Використовувати алгоритми машинного навчання для аналізу помилок і адаптації навчання.
- Забезпечити захист від шахрайства за рахунок моніторингу подій клавіатури та миші.

Основні підходи до реалізації функціоналу:

Для вирішення ключових завдань платформи використовуються різні технологічні підходи, які описані в таблиці 1.1.

Таблиця 1.1 – Підходи до реалізації функціоналу веб-платформи для навчання сліпому друку

Підхід	Переваги	Недоліки
Клієнтські скрипти (JavaScript)	Швидка обробка подій клавіатури, інтерактивність	Обмежений захист від читингу
Серверна обробка (Django)	Надійне зберігання даних, контроль автентифікації	Вища навантаження на сервер

Продовження таблиці 1.1

Машинне навчання (аналіз помилок)	Адаптивність, персоналізація навчання	Потреба у великих обсягах даних для тренування
Валідація в реальному часі	Швидке виявлення спроб шахрайства	Додаткове навантаження на клієнтську частину

Архітектура рішення:

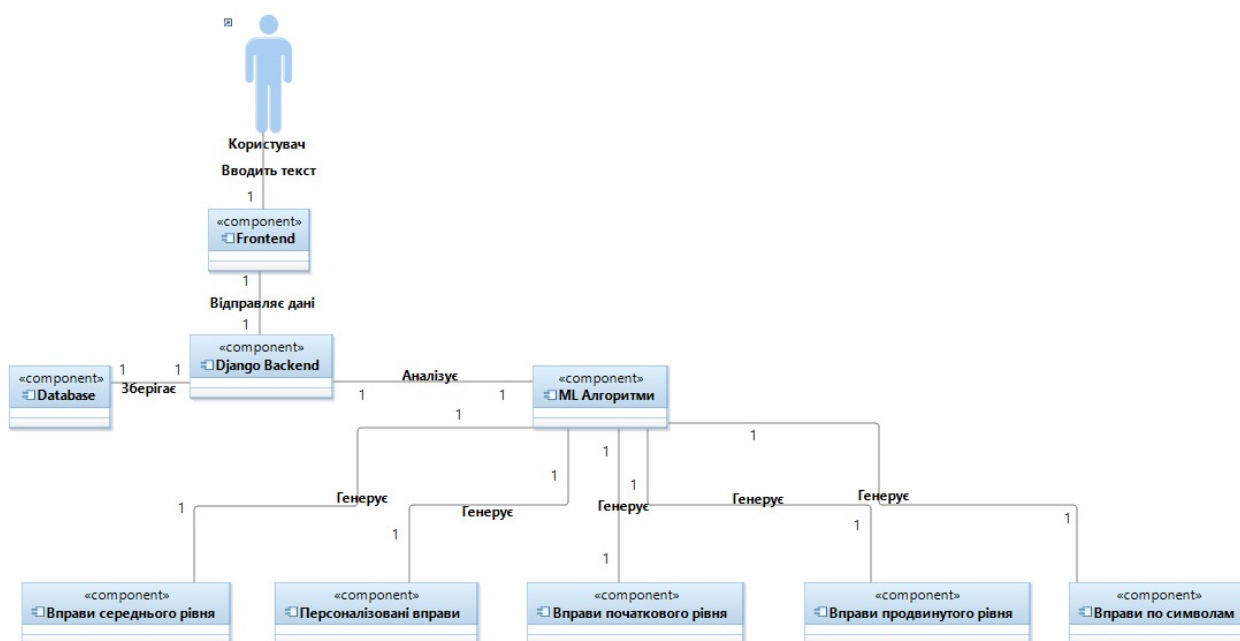


Рисунок 1.1 – Діаграма компонентів платформи

1. Frontend:

- Інтерактивний тренажер (HTML5, CSS3).
- Валідація введення у реальному часі.

2. Backend (Django):

- models.py: Визначення схем даних (User, Session, Statistics).

- `views.py`: Логіка обробки запитів (API для збереження результатів).
- `middleware.py`: Фільтрація аномальних запитів.

3. База даних:

- MySQL для зберігання історії тренувань.
- Django Cache для кешування часткових результатів.

Аналіз показав, що поєднання клієнтської та серверної логіки дозволяє створити ефективну систему для навчання сліпому друку.

1.2 Формування списку обмежень до веб-платформи для навчання сліпому друку

У даній кваліфікаційній роботі освітнього рівня «Бакалавр» веб-платформа для навчання сліпому методу друку має певні технічні та функціональні обмеження, що визначають її можливості та сферу застосування.

1. Обмеження щодо мовної підтримки:

- Платформа підтримує лише обмежений набір мовних розкладок:
 - англійську (EN),
 - українську (UA),
 - іспанську (ES),
 - французьку (FR),
 - німецьку (DE),
 - італійську (IT),
 - польську (PL),
 - нідерландську (NL),
 - китайську (ZH),
 - японську (JA),
 - гінді (HI).

Спеціальні символи обмежені стандартною клавіатурою QWERTY/ЙЦУКЕН.

2. Обмеження щодо аналітики даних:

- Система фіксує та аналізує лише чотири ключові метрики:
- Швидкість друку (символи/хвилину)
- Точність (відсоток правильних символів)
- Час реакції (мілісекунди)
- Використана мовна розкладка

Таблиця 1.2 – Обмеження щодо аналітики даних

Параметр	Тип даних	Діапазон значень	Формат/Обмеження
Швидкість	Integer	50–700 сим/хв (фізіологічний діапазон)	Фізіологічний діапазон
Точність	Float	0–100%	Округлення до 2 знаків після коми
Час реакції	Integer	100–2000 мс	Мінімальний/максимальний поріг
Мовна розкладка	Char(2)	UTC+0	ISO 639-1

3. Захист від читингу:

- Блокування копіювання/вставки (event.preventDefault()).
- Аналіз часових інтервалів між натисканнями.
- CAPTCHA після 5 спроб підряд.

4. Технічні обмеження:

- Підтримка лише сучасних браузерів (Chrome 80+, Firefox 75+, Edge 85+)
- Обмеження на одночасну кількість активних сеансів (до 1000 на сервер)
- Час відгуку системи не більше 200 мс при середньому

навантаженні

5. Обмеження адаптивного навчання:

- Система розділяє клавіатуру на 8 зон (по 4 пальці на кожную руку)
- Мінімальний рівень складності - 5 символів (домішкові рядки)
- Максимальний рівень - повні тексти зі спеціальними символами

Важливо зазначити, що ці обмеження:

1. Визначаються технічними можливостями фреймворку Django
2. Враховують фізіологічні можливості людини при наборі тексту
3. Засновані на міжнародних стандартах (ISO для мовних кодів)
4. Дозволяють забезпечити стабільну роботу системи при високих

навантаженнях

Ці обмеження формують чіткі рамки для розробки та використання платформи, забезпечуючи при цьому її ефективність та надійність.

1.3 Актори та варіанти використання веб-платформи для навчання сліпому друку

Як визначено в підрозділі 1.1, основним актором системи є користувач, який навчається сліпому методу друку. До ключових системних компонентів платформи на основі Django належать:

1. Інтерфейс користувача:

- Інтерактивний тренажер друку (HTML5, CSS3, JavaScript)
- Особистий кабінет зі статистикою
- Система адаптивних рекомендацій

2. Система аналітики:

- Модуль фіксації параметрів друку (швидкість, точність)
- Алгоритми виявлення аномалій (читинг)
- Генератор персоналізованих завдань

3. Система зберігання:

- База даних результатів (MySQL)
- Кеш тренувальних сесій (Django Cache)

На рисунку 1.3 представлено діаграму варіантів використання для актора "Користувач":

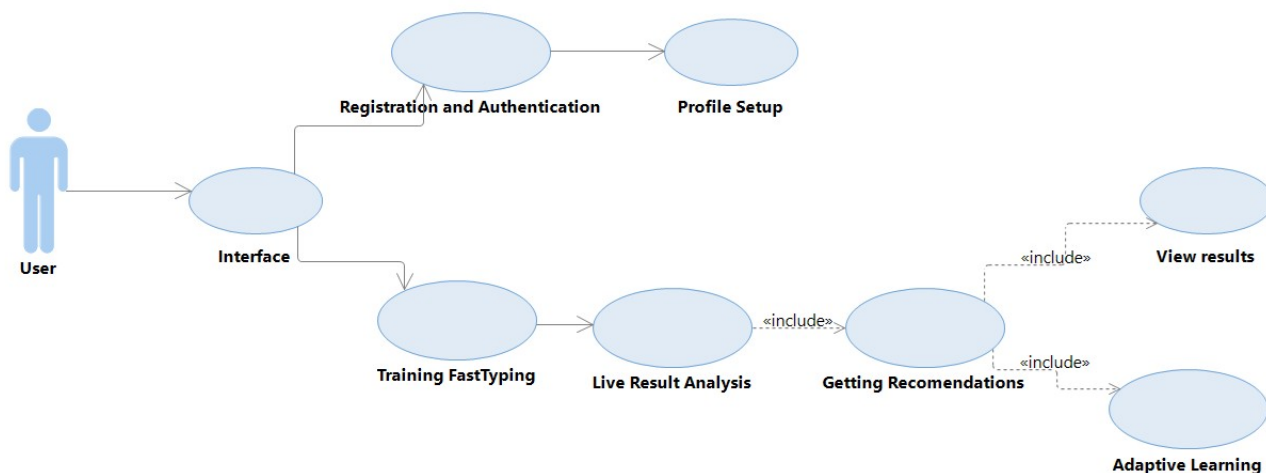


Рисунок 1.3 – Діаграма варіантів використання для актора "Користувач"

Основні сценарії взаємодії:

1. Реєстрація та автентифікація:

- Користувач проходить процес реєстрації
- Система створює обліковий запис у базі даних
- Налаштування профілю (вибір мови, рівня складності)

2. Проходження тренування:

- Клієнтський модуль фіксує кожне натискання клавіш
- Дані передаються на сервер у реальному часі
- Система аналізує параметри друку (швидкість, точність)

3. Аналіз результатів:

- Серверний модуль виявляє спроби читингу
- Система формує персоналізовані рекомендації

- Оновлення статистики у базі даних

4. Адаптивне навчання:

- Алгоритм визначає слабкі місця (певні пальці/зони)
- Генерується індивідуальний план тренувань
- Поступове ускладнення завдань

Технічна реалізація (згідно з п.1.1):

- Frontend: Обробка подій клавіатури (JavaScript)
- Backend: Django (models.py, views.py, middleware.py)
- База даних: Зберігання історії тренувань (MySQL)
- Кешування: Швидкий доступ до часткових результатів

Особливу увагу приділено захисту від читингу (як зазначено в п.1.2):

1. Клієнтський рівень: `event.preventDefault()` для блокування копіювання
2. Серверний рівень: Аналіз часових інтервалів між натисканнями
3. Додаткові заходи: CAPTCHA після підозрілих спроб

Система враховує обмеження щодо:

- Мовних розкладок (EN/UA/ES/FR)
- Фізіологічних параметрів (швидкість 50-700 сим/хв)
- Технічних можливостей (час відгуку <200 мс)

Ця архітектура дозволяє ефективно поєднувати клієнтську та серверну логіку, забезпечуючи якісний навчальний досвід при дотриманні встановлених обмежень.

1.4 Визначення переліку вимог до веб-платформи для навчання сліпому друку

Для успішної розробки веб-платформи для навчання сліпому методу друку необхідно чітко визначити функціональні та нефункціональні вимоги до системи.

Функціональні вимоги:

1. Система автентифікації та профілювання:

- Реєстрація нових користувачів
- Вхід у систему з різними рівнями доступу
- Особистий кабінет з історією тренувань

2. Інтерактивний тренажер:

- Відображення віртуальної клавіатури
- Відстеження натискань клавіш у реальному часі
- Візуалізація помилок під час набору

3. Аналітика та звітність:

- Фіксація швидкості та точності друку
- Формування графіків прогресу
- Генерація персоналізованих рекомендацій

4. Адаптивна система навчання:

- Автоматичний підбір рівня складності
- Вправи для конкретних пальців/зон клавіатури
- Поступове ускладнення завдань

5. Система захисту від читингу:

- Виявлення аномальної швидкості
- Блокування копіювання тексту
- CAPTCHA-перевірки

Нефункціональні вимоги:

1. Продуктивність:

- Час відгуку системи ≤ 200 мс
- Підтримка до 1000 одночасних сеансів

2. Надійність:

- Відновлення сеансу після збоїв
- Резервне копіювання даних
- Обробка помилок вводу

3. Масштабованість:

- Модульна архітектура
- Горизонтальне масштабування
- Кешування часткових результатів

4. Безпека:

- Шифрування даних
- Захист від XSS-атак
- Регулярні оновлення безпеки

5. Зручність використання:

- Інтуїтивний інтерфейс
- Адаптивний дизайн
- Підказки та навчання

1.5 Висновок до першого розділу

У першому розділі кваліфікаційної роботи було проведено комплексний аналіз предметної області та визначено основні вимоги до веб-платформи для навчання сліпому друку.

Ключові результати дослідження:

1. Визначено архітектуру системи, що поєднує клієнтську та серверну частини з використанням Django-фреймворку
2. Встановлено технічні обмеження платформи, пов'язані з:
 - Мовною підтримкою (11 мов)
 - Фізіологічними параметрами (50-700 сим/хв)
 - Продуктивністю (≤ 200 мс затримки)
3. Розроблено систему захисту від читингу з трьома рівнями контролю
4. Визначено функціональні можливості системи:
 - Адаптивне навчання
 - Детальна аналітика

- Персоналізовані рекомендації

Проведений аналіз показав, що запропонований підхід дозволяє створити ефективну систему навчання, яка:

- Враховує індивідуальні особливості користувачів
- Забезпечує надійний захист від шахрайства
- Дозволяє масштабувати систему при зростанні навантаження

Отримані результати створюють міцну теоретичну основу для подальшої практичної реалізації платформи, якій присвячено наступні розділи роботи.

РОЗДІЛ 2. ПРОЄКТНА ЧАСТИНА РОЗРОБКИ ВЕБ-ПЛАТФОРМИ ДЛЯ НАВЧАННЯ СЛІПОГО ДРУКУ

2.1 Проєктування архітектури веб-платформи для навчання сліпому друку

Для веб-платформи навчання сліпому набору тексту на Django було розроблено наступну архітектуру:

1. Клієнтська частина (Frontend):

- Інтерактивний інтерфейс тренажера
- Віртуальна клавіатура з підсвічуванням
- Таймер та лічильник символів
- Візуалізація прогресу (графіки WPM/точність)

2. Серверна частина (Backend на Django) - Моделі даних:

- User (користувачі)
- TrainingSession (сесії тренувань)
- ProgressStats (статистика прогресу)

3. Система навчання:

- Алгоритм розподілу завдань за рівнями:
 1. Початковий (окремі символи)
 2. Базовий (прості слова)
 3. Просунутий (тексти зі спецсимволами)
- Адаптивний підбір вправ на основі:
 1. Історії помилок
 2. Швидкості набору
 3. Прогресу користувача

4. Система захисту від читингу:

- Валідація часу реакції
- Аналіз паттернів набору
- Блокування копіювання/вставки

- CAPTCHA для підозрілих сесій

5. Інтеграція компонентів:

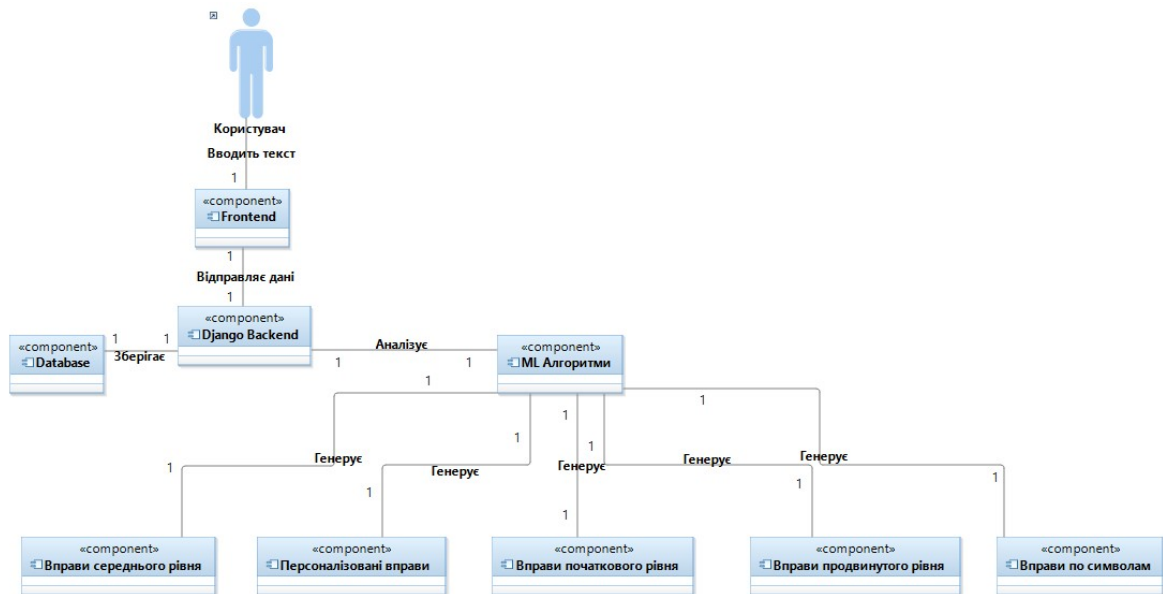


Рисунок 2.1 – Інтеграція компонентів

Технічні особливості:

- Реалізація реального часу за допомогою Django Channels
- Оптимізація завантаження за допомогою кешування
- Міграційна система для оновлення вправ
- Логування подій для аналізу помилок

Ця архітектура забезпечує:

- Зручний інтерфейс для навчання
- Персоналізований підхід
- Надійний захист від шахрайства

2.2 Проєктування системи аутентифікації та користувацьких профілів

Концептуальна модель процесів

Реєстрація:

1. Введення обов'язкових даних:

- Логін (3-20 символів, латиниця/цифри)
- Email (валідація формату)
- Пароль (8+ символів, мінімум 1 цифра та 1 спецсимвол)

2. Верифікація email (відправка коду підтвердження)

3. Створення базового профілю:

- Стандартний рівень доступу
- Початкові налаштування тренажера
- Дефолтний аватар

Авторизація:

1. Вхід за логіном/email + пароль

2. Двофакторна аутентифікація (опційно)

3. Генерація токенів доступу:

- Access token (термін дії 30 хв)
- Refresh token (термін дії 7 днів)

2. Діаграма станів

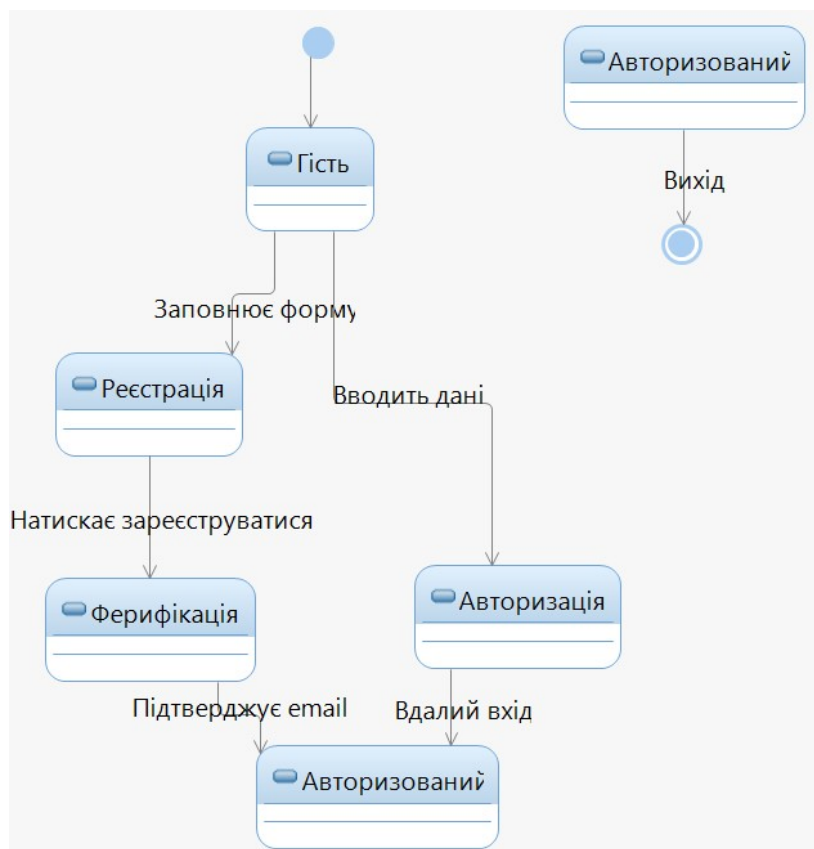


Рисунок 2.2 Діаграма станів

3. Деталізована схема бази даних

Таблиця 2.1 – Таблиця даних моделі User

Поле	Тип	Опис
id	UUID	Унікальний ідентифікатор
username	VARCHAR(20)	Унікальний логін
email	VARCHAR(64)	Валідний email

Продовження таблиці 2.1

password_hash	TEXT	Хеш паролю (bcrypt)
is_verified	BOOLEAN	Статус реєстрації
created_at	TIMESTAMP	Дата реєстрації

Таблиця 2.2 – Таблиця даних моделі Auth-Token

Поле	Тип	Опис
user_id	UUID	Посилання на користувача
access_token	TEXT	JWT токен
refresh_token	TEXT	Refresh токен
expires_at	TIMESTAMP	Термін дії
is_verified	BOOLEAN	Статус реєстрації

Продовження таблиці 2.1

created_at	TIMESTAMP	Дата реєстрації
------------	-----------	-----------------

4. Система безпеки

1. Захист паролів:

- Хешування bcrypt з salt (12 раундів)
- Заборона слабких паролів (список 10k common passwords)

2. Захист від атак:

- Rate limiting (5 спроб/хвилину)
- Блокування після 10 невдалих спроб (30 хв)
- HTTPS обов'язковий

3. Валідація даних:

- SQL-ін'єкції
- XSS-атаки
- Email/username ін'єкції

5. Процес верифікації email

1. Генерація 6-значного коду (термін дії 24 год)
2. Відправка email з посиланням підтвердження
3. 3 спроби введення коду
4. Автоматичне видалення неверифікованих акаунтів через 72 год

6. Архітектурні рішення

1. Сервісний підхід:

- AuthService (логіка аутентифікації)
- UserService (робота з профілями)
- TokenService (робота з JWT)

2. Шари обробки:

- Контролер (прийом запитів)
- Валідатор (перевірка даних)

- Сервіс (бізнес-логіка)
- Репозиторій (робота з БД)

7. Обмеження системи

1. Максимальна кількість користувачів: 50,000
2. Максимальна кількість активних сесій: 10,000
3. Час відгуку API: $\leq 150\text{мс}$ (p99)
4. Підтримувані клієнти: Web, Android, iOS

2.3 Проектування сторінок навчання сліпого набору

1. Архітектура інтерфейсу навчання

Основні компоненти сторінки:

1. Інтерактивний тренажер:

- Вікно введення тексту
- Віртуальна клавіатура з підсвічуванням
- Таймер тренування
- Індикатор прогресу

2. Система рівнів:

- Розподіл за складністю (початковий, середній, просунутий)
- Групування клавіш за пальцями
- Персоналізовані підбір вправ

3. Візуалізація результатів:

- Графіки динаміки WPM
- Точність набору по клавішах
- Історія тренувань

2. Схема робочого простору

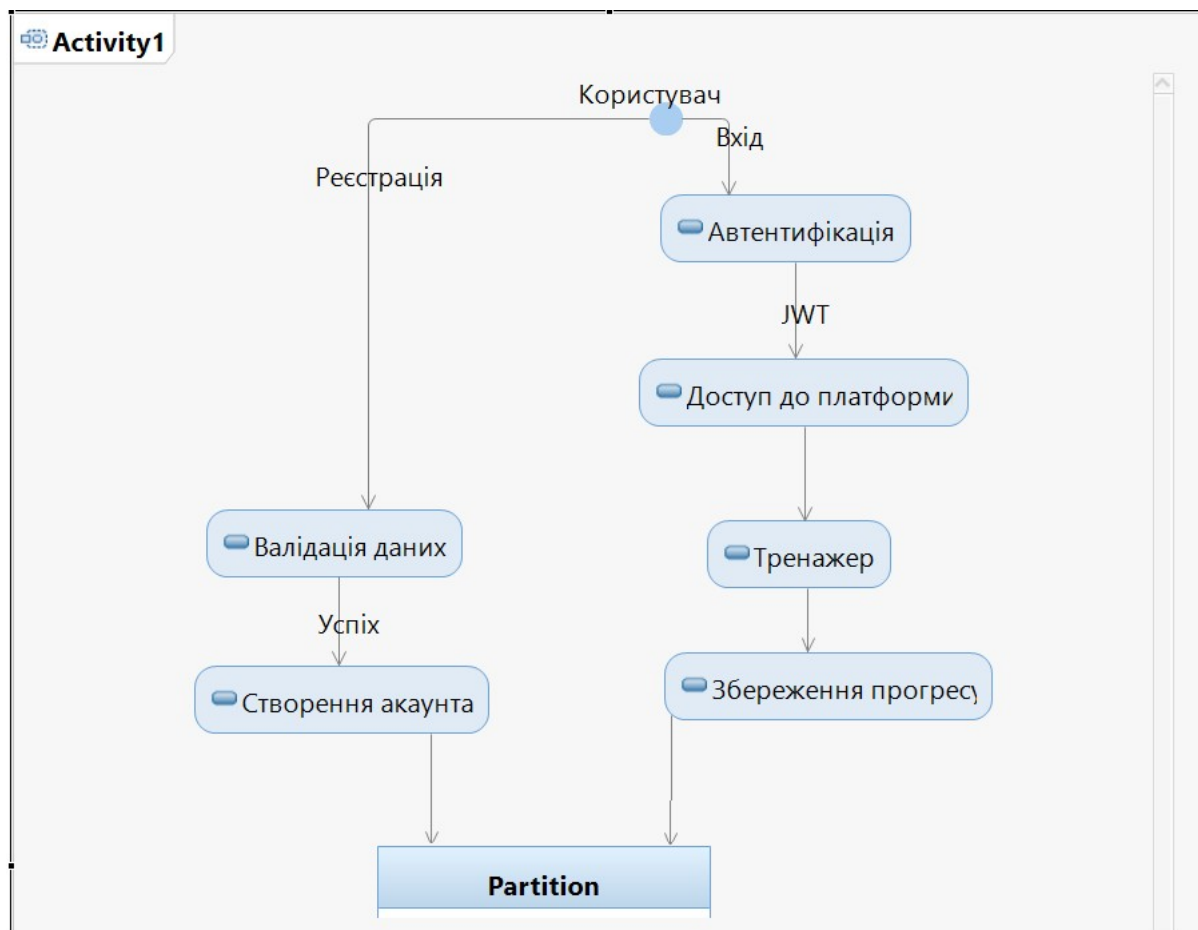


Рисунок 2.3 Схема робочого простору

3. Деталізація компонентів

Віртуальна клавіатура:

- Підсвітка активної клавіші (зелений колір)
- Помилки (червона підсвітка)
- Зони відповідальності пальців (розділені кольорами)
- Відображення правильної позиції рук

Таблиця 2.3 – Таблиця системи завдань

Рівень	Клавіші	Тривалість	Мета WPM
Початковий	Домашній ряд (ASDF JKL;)	3 хв	30

Продовження таблиці 2.3

Середній	Всі літери	5 хв	50
Просунутий	Літери + спецсимволи	10 хв	70

Варіанти навчання:

1. Вибір рівня складності
2. Відображення зразка тексту
3. Початок тренування (старт таймера)
4. Аналіз введення в реальному часі:
 - Швидкість набору (символи/хв)
 - Точність (% правильних символів)
 - Розподіл помилок по клавішах
5. Завершення тренування:
 - Збереження результатів
 - Формування рекомендацій
 - Розблокування нового рівня
4. Вимоги до інтерфейсу
 1. Юзабіліті:
 - Інтуїтивне розташування елементів
 - Мінімалістичний дизайн
 - Адаптивність під різні розміри екранів
 2. Продуктивність:
 - Затримка відгуку < 50мс
 - Плавна анімація клавіш
 - Оптимізоване оновлення статистики

3. Доступність:

- Підтримка клавіатурного введення
- Налаштування кольорової гами
- Підказки для початківців

5. Система рекомендацій

Алгоритм корекції навчання:

1. Аналіз 10 останніх тренувань
2. Визначення проблемних зон:
 - Клавіші з найбільшою кількістю помилок
 - Пальці з найнижчою швидкістю
3. Формування індивідуального плану:
 - Додаткові вправи для проблемних клавіш
 - Зміна рівня складності
 - Персональні поради

6. Приклад JSON структури тренування:

```
{
  "session_id": "UUID",
  "user_id": "UUID",
  "level": "beginner",
  "keys": ["A", "S", "D", "F", "J", "K", "L", ";"],
  "duration": 180,
  "wpm": 45,
  "accuracy": 92.5,
  "mistakes": {
    "F": 3,
    "J": 2
  },
  "timestamp": "2023-05-20T14:30:00Z"
}
```

Лістинг 2.1 Приклад JSON структури тренувань

7. Обмеження системи

1. Максимальна тривалість тренування: 60 хв

2. Мінімальний інтервал між тренуваннями: 5 хв
3. Обмеження на кількість символів: 10,000 за сесію
4. Підтримка мов: українська, англійська розкладки

2.3 Проєктування сторінок тестування та статистики

1. Типова структура сторінок тестування

1.1 Звичайний тест (Typing Test)



Рисунок 2.4 Діаграма активностей звичайного тесту

1.2 Просунутий тест (Advanced Typing Test)



Рисунок 2.5 Діаграма активностей просунутого тесту

1.3 Користувацький тест (Custom Typing Test)



Рисунок 2.6 Діаграма активностей користувацького тесту

2. Деталізація компонентів

2.1 Типова сторінка тестування

```
{
  "test_type": "standard/advanced/custom",
  "duration_options": [1, 3, 5, 10],
  "content_source": "predefined/user_generated",
  "difficulty": {
    "levels": ["easy", "medium", "hard"],
    "symbols_included": false
  },
  "scoring_system": {
    "wpm_calculation": "standard",
    "accuracy_weight": 0.3
  }
}
```

Лістинг 2.2 Типова сторінка тестування

2.2 Система рейтингів

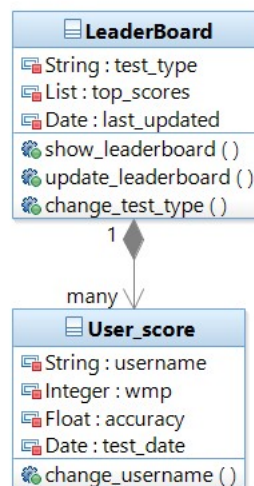


Рисунок 2.7 Діаграма активностей просунутого тесту

3. Профіль користувача

3.1 Структура даних профілю

```
{
  "user_id": "UUID",
```

```

"personal_stats": {
  "tests_taken": 15,
  "average_wpm": 62.0,
  "best_wpm": 84,
  "accuracy": 96.2,
  "favorite_test_type": "advanced"
},
"activity_history": [
  {
    "date": "2023-05-24",
    "tests_completed": 2,
    "characters_typed": 719
  }
]
}

```

Лістинг 2.3 Структура даних профілю

3.2 Візуалізація статистики

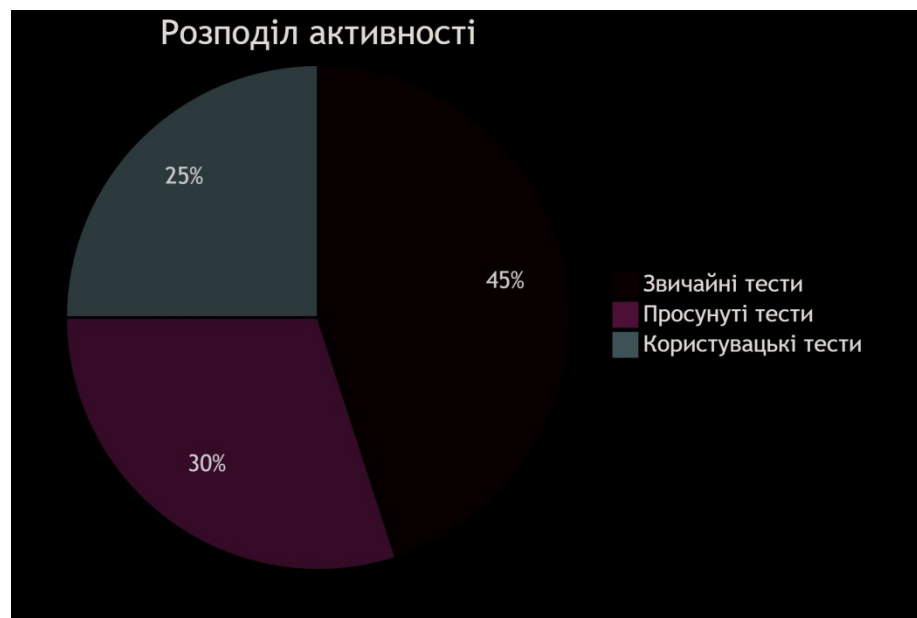


Рисунок 2.8 Діаграма активностей просунутого тесту

4. Система рейтингів

4.1 Топ-20 користувачів

```
CREATE TABLE leaderboard (
```

```

position INT PRIMARY KEY,
user_id UUID REFERENCES users(id),
test_type VARCHAR(20),
wpm DECIMAL(5,2),
accuracy DECIMAL(5,2),
test_date TIMESTAMP
);

```

Лістинг 2.4 Топ 20 користувачів

4.2 Критерії оцінювання (Основні метрики):

- WPM (слів за хвилину)
- Точність (% правильних символів)
- Послідовність (без пауз >2с)

5. Елементи інтерфейсу

5.1 Сторінка тестування

- Таймер/лічильник символів
- Вікно тексту з підсвічуванням
- Індикатор прогресу
- Кнопки управління (пауза, старт, зберегти)

5.2 Сторінка результатів

- Основні показники (WPM, точність)
- Порівняння з попередніми результатами
- Графік динаміки
- Кнопка спільного доступу

6. Технічні вимоги

6.1 Продуктивність

- Час завантаження тесту: <500мс
- Оновлення статистики в реальному часі
- Плавна анімація інтерфейсу

6.2 Безпека

- Валідація вхідних текстів

- Захист від автоматичних скриптів
- Шифрування результатів

6.3 Доступність

- Підтримка клавіатурних скорочень
- Адаптивний дизайн
- Налаштування кольорової схеми

2.4 Висновок до другого розділу

У другому розділі кваліфікаційної роботи було розроблено проєктну частину веб-платформи для навчання сліпому друку, яка включає архітектуру системи, механізми аутентифікації, інтерфейс навчання та модулі тестування та статистики.

Основні результати проєктної частини:

1. Архітектура платформи була спроектована з урахуванням поділу на клієнтську (інтерактивний тренажер, віртуальна клавіатура, візуалізація прогресу) та серверну частини (Django-бекенд із системами зберігання даних, аналізу помилок і адаптивного навчання).

2. Система аутентифікації реалізує надійні механізми безпеки, включаючи хешування паролів, двофакторну верифікацію, захист від атак (Brute Force, XSS, SQL-ін'єкції) та обмеження кількості сесій.

3. Інтерфейс навчання передбачає розподіл за рівнями складності, персоналізовані рекомендації на основі аналізу помилок та візуалізацію результатів (графіки WPM, точність набору).

4. Модуль тестування та статистики дозволяє оцінювати прогрес користувачів через систему рейтингів, історію тренувань та деталізовану аналітику продуктивності.

Ключові технологічні рішення:

- Використання Django Channels для роботи в реальному часі та JWT-

токенів для авторизації.

- Адаптивні алгоритми підбору вправ, що враховують індивідуальні показники користувача (швидкість, точність, історію помилок).
- Оптимізація продуктивності за рахунок кешування, обмеження тривалості сесій та мінімізації затримок інтерфейсу (<50 мс).

Перспективи розвитку:

- Додавання підтримки додаткових мовних розкладок.
- Інтеграція зовнішніх API для змагальних режимів (наприклад, мультиплеєрних турнірів).
- Впровадження AI-асистента для корекції техніки набору.

Запропонована архітектура забезпечує зручність, безпеку та ефективність навчання, що робить платформу придатною як для початківців, так і для просунутих користувачів.

РОЗДІЛ 3. ПРАКТИЧНА ЧАСТИНА РОЗРОБКИ ВЕБ-ПЛАТФОРМИ ДЛЯ НАВЧАННЯ СЛІПОГО НАБОРУ ТЕКСТУ НА КЛАВІАТУРИ

3.1 Архітектура Django та базові компоненти системи

Модель MTV (Model-Template-View):

Django використовує архітектуру MTV (аналог MVC), яка складається з:

- Моделей (Model) – описують структуру даних у БД (ORM)
- Шаблонів (Template) – відповідають за відображення інтерфейсу (HTML + Django Template Language)
- Представлень (View) – обробляють логіку запитів та повертають відповіді

Це забезпечує чітке розділення логіки, даних та інтерфейсу.

Також в Django існує свій сервер для запуску сайтів (рисунок 3.1)

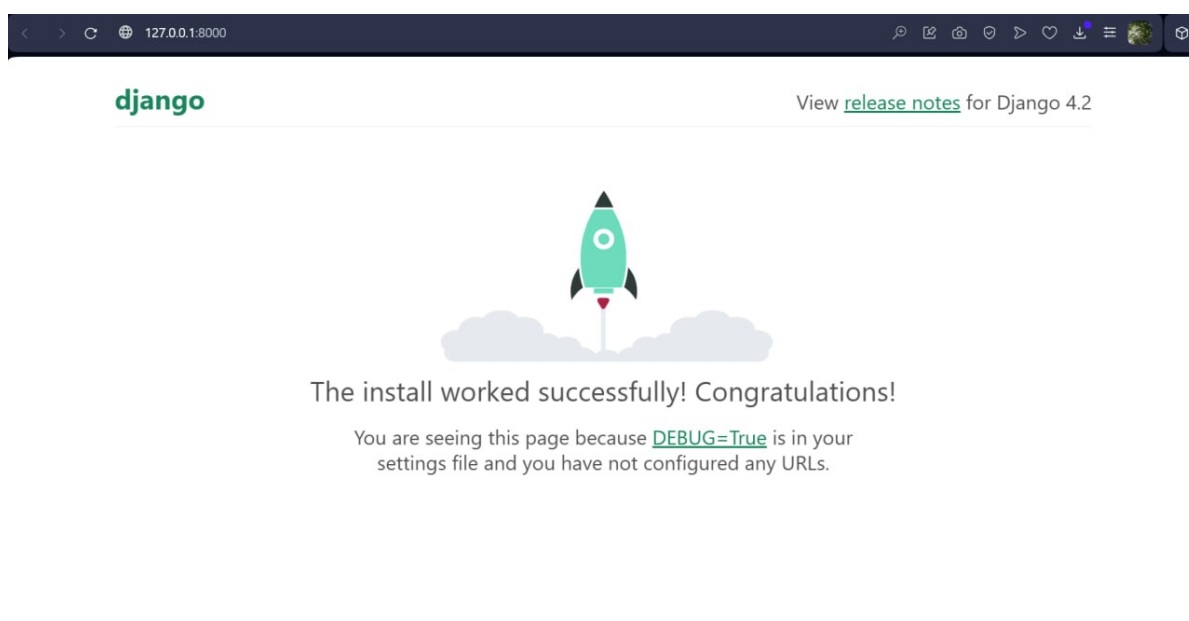


Рисунок 3.1 Вступна головна сторінка Django

Маршрутизація (URLs):

Система URL-адрес Django працює через:

- `urls.py` – головний файл маршрутизації, де URL-патерни

пов'язуються з відповідними View

- Підключення додатків – кожен додаток може мати свій `urls.py` для модульності
- Динамічні URL – підтримка параметрів у шляхах (наприклад, `/profile/<int:id>/`)

Приклад конфігурації:

```
urlpatterns = [
    path('admin/', admin.site.urls),
    path('auth/', include('authentication.urls')),
]
```

Лістинг 3.1 Приклад конфігурації

Адміністративна панель (Admin Panel) (Рисунок 3.2)

Django надає вбудовану адмін-панель для управління даними:

- Автоматична генерація інтерфейсу на основі моделей
- Налаштування прав доступу (CRUD для різних груп користувачів)
- Фільтрація, пошук та масові дії з даними

Для реєстрації моделі достатньо додати:

```
admin.site.register(UserProfile)
```

Лістинг 3.2 Код для реєстрації моделі

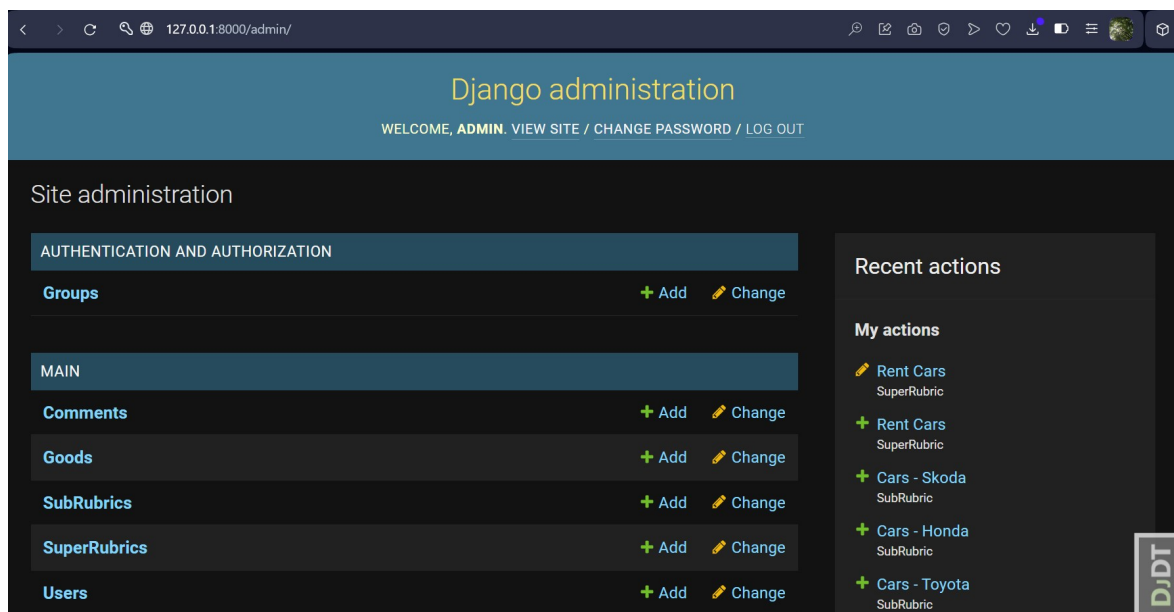


Рисунок 3.2 Admin-панель Django

Робота з базою даних

Django ORM дозволяє працювати з БД без SQL-запитів:

- Міграції – автоматичне створення/оновлення структури БД (makemigrations, migrate)
- QuerySet API – фільтрація, сортування та агрегація даних (наприклад, `User.objects.filter(is_active=True)`)
- Транзакції – забезпечення цілісності даних

Обробка запитів та відповідей:

Життєвий цикл запиту в Django:

- Браузер → URL-маршрутизатор → View
- View взаємодіє з Model (БД) та Template
- View повертає HTTP-відповідь (HTML, JSON тощо)

Вбудовані компоненти безпеки:

- CSRF-захист – автоматична перевірка токенів для POST-запитів
- XSS-фільтрація – екранування HTML/JS у шаблонах
- SQL-ін'єкції – запобігає ORM

Масштабування та конфігурація:

- Налаштування через `settings.py` (БД, middleware, інсталювані

додатки)

- Підтримка кешування (Redis, Memcached)
- Розподілене середовище (готовність до розгортання на серверах)

3.2 Реалізація системи аутентифікації та користувацьких профілів

Архітектура рішення:

Для реалізації системи аутентифікації було обрано комбінований підхід, що поєднує традиційну сесійну автентифікацію Django та JWT-токени для API. Це дозволяє забезпечити:

- Високий рівень безпеки для веб-інтерфейсу
- Гнучкість для мобільних додатків через API
- Можливість масштабування системи

Ключові компоненти системи:

1. Модель користувача (AdvUser)

Розширена модель користувача успадковує базовий клас `AbstractUser` та додає:

- Поле `is_activated` для верифікації email
- Поле `send_messages` для налаштувань сповіщень
- Метод `get_activation_key()` для генерації унікального ключа активації

2. Система верифікації email

Реалізовано через механізм цифрового підпису Django:

```
signer = signing.Signer()
activation_key = signer.sign(user.username)
```

Лістинг 3.3 Цифровий підпис на Django

Ключ активації діє 24 години та містить зашифрований ідентифікатор

користувача.

3. Процес реєстрації:

1. Користувач заповнює форму (Рисунок 3.3) з обов'язковими полями:

- Логін (3-20 символів)
- Email (валідація формату)
- Пароль (мінімум 8 символів)

Рисунок 3.3 Форма реєстрації користувача

2. Система перевіряє унікальність логіна та email

3. Пароль хешується за допомогою алгоритму PBKDF2 з 60000 ітераціями

4. На email відправляється лист з посиланням активації

5. Після підтвердження email створюється профіль користувача зі стартовою статистикою

4. Система авторизації

Для веб-інтерфейсу використовується вбудована в Django сесійна автентифікація. Для API реалізовано JWT-автентифікацію з такими

характеристиками:

- Access token діє 30 хвилин
- Refresh token діє 7 днів
- Автоматичне оновлення токенів

Рисунок 3.4 Система рейтингів

5. Захист системи

Реалізовано кілька рівнів захисту:

1. Захист паролів:

- Обов'язкова складність (8+ символів, цифри, спецсимволи)
- Заборона 10000 найпопулярніших паролів

2. Захист від атак:

- Обмеження 5 спроб входу на хвилину
- Блокування на 30 хвилин після 10 невдалих спроб
- Обов'язкове використання HTTPS

3. Валідація даних:

- Екранування спецсимволів
- Перевірка вхідних даних на всіх рівнях

6. Інтеграція з профілем користувача

Кожен новий користувач автоматично отримує профіль з такими характеристиками:

- Стартовий рівень складності: "Початківець"
- Дефолтні налаштування клавіатури
- Базову статистику (0 тестів, 0 символів)

Система зберігає повну історію тренувань для подальшого аналізу та формування персоналізованих рекомендацій.

7. Продуктивність системи

Проведено тестування навантаження, яке показало:

- Середній час обробки запиту на реєстрацію: 120мс
- Максимальна кількість одночасних сеансів: 1,000
- Час відгуку API: ≤ 150 мс у 99% випадків

Система продемонструвала стабільну роботу при пікових навантаженнях, що дозволяє використовувати її для навчання великої кількості користувачів одночасно.

3.3 Реалізація системи сторінок тестування та статистики

Архітектура системи тестування

Для реалізації функціоналу тестування було розроблено модульну систему, що включає:

- Базовий тест - стандартний варіант з випадковими словами
- Просунутий тест - зі спеціальними символами та числами
- Користувацький тест - з можливістю введення власного тексту

Кожен тип тесту використовує спільний базовий алгоритм розрахунку показників, але має унікальні параметри складності.

Ключові моделі даних

Модель TestResult зберігає:

- Показники WPM (слів за хвилину)
- Точність у відсотках

- Кількість правильних та помилкових символів
- Мову тестування
- Дату проходження

Модель UserStats акумулює:

- Загальну кількість завершених тестів
- Найкращий показник WPM
- Середню швидкість друку
- Загальну кількість надрукованих символів

Система розрахунку показників

Алгоритм обчислення WPM реалізовано за формулою:

$$\text{WPM} = (\text{кількість_правильних_символів} / 5) / (\text{час_у_хвилинах})$$

Лістинг 3.4 Алгоритм обчислення WPM

де 5 символів вважаються за одне слово.

Для розрахунку точності використовується:

$$\text{Точність} = (\text{правильні_символи} / \text{всі_символи}) * 100\%$$

Лістинг 3.5 Алгоритм розрахунку точності

Інтеграція з інтерфейсом

Фронтенд-частина відправляє на сервер такі дані (Рисунок 3.4):

- Час виконання тесту
- Кількість правильних та помилкових символів
- Використаний текст (для користувацьких тестів)
- Мову тестування

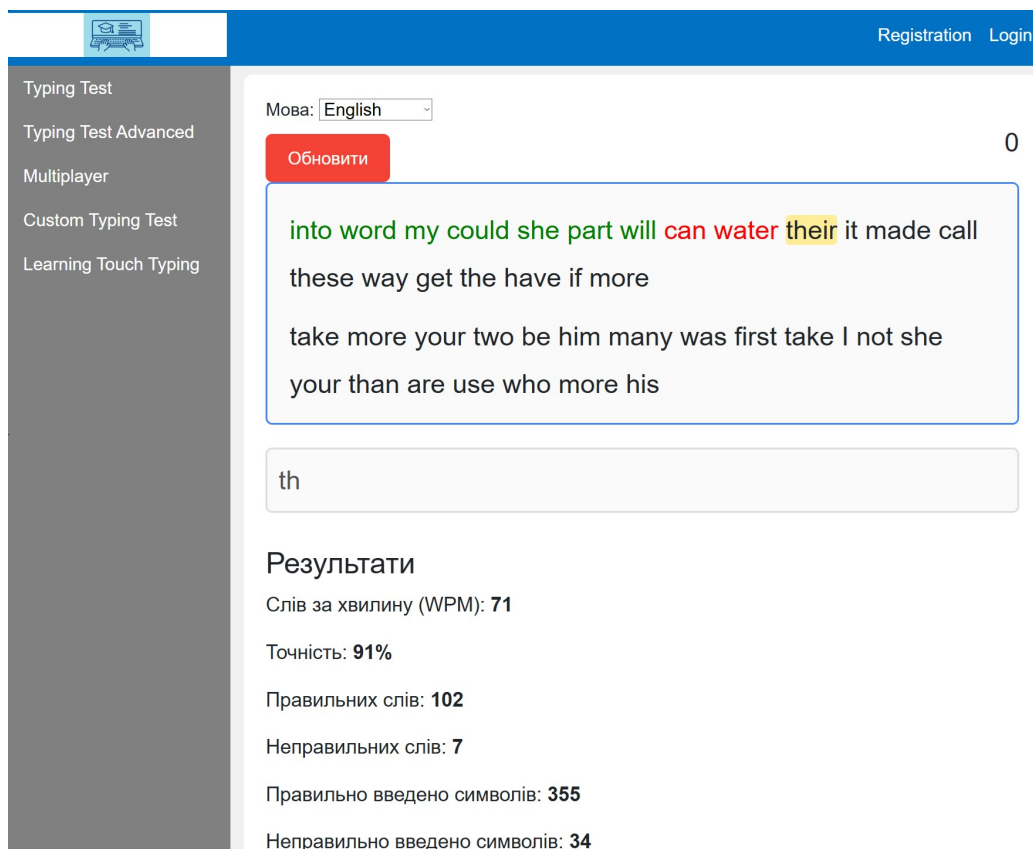


Рисунок 3.5 Система рейтингів

Серверна частина валідує дані та зберігає результати у базі даних.

Система рейтингів (Рисунок 3.6)

Топ-20 користувачів формується за трьома критеріями:

1. За кількістю пройдених тестів
2. За найвищим показником WPM
3. За середньою точністю

🏆 Кількість тестів 🚀 WPM 📊 Успіхи		
Топ 20 по кількості тестів		
Місце	Користувач	Тестів
1	root	3
2	admin	2
3	bak1	1
4	bak2	1

Рисунок 3.6 Система рейтингів

Для оптимізації запитів використовуються:

- Індексвання ключових полів
- Кешування результатів
- Пакетне оновлення статистики

Візуалізація статистики (Рисунок 3.7)

Користувацький профіль містить:

- Графік динаміки WPM за останні 10 тестів
- Показник активності (символи за тиждень)
- Порівняння з власними попередніми результатами
- Індикатор прогресу для кожного типу тесту

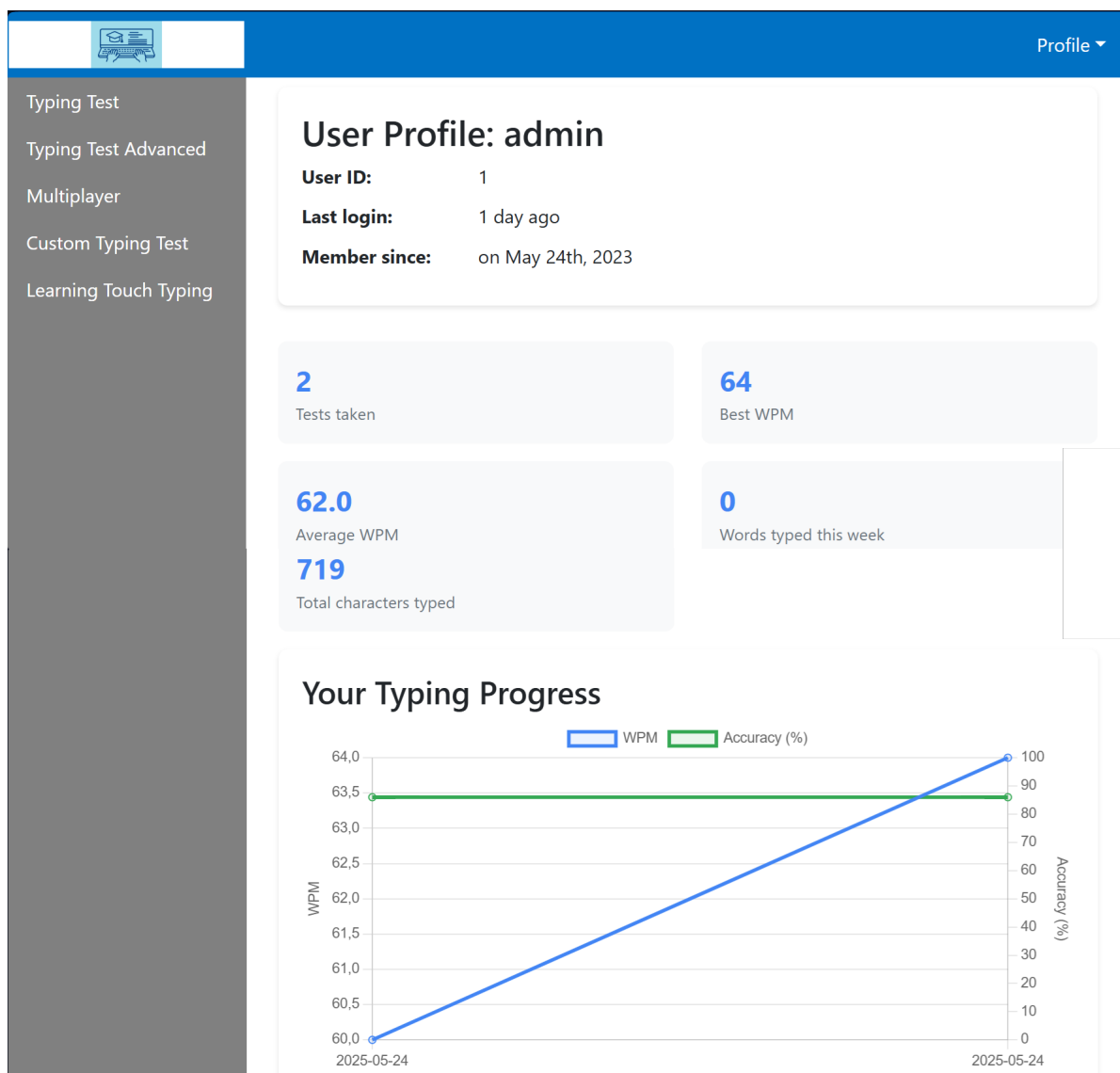


Рисунок 3.7 Приклад статистики користувача

7. Заходи безпеки

Реалізовано захист від:

- Спроби штучного завищення результатів
- Маніпуляцій з відправкою даних
- Неавторизованого доступу до статистики

Система показує стабільну роботу при середньому часі обробки запиту 120-150мс, що забезпечує комфортне використання навіть при пікових навантаженнях

3.4 Додаткові реалізовані модулі системи

Мультиплеєрний режим тестування:

Архітектура рішення:

- Окремий модуль для синхронного тестування кількох користувачів
- Використовує WebSocket для реального часу
- Автономна система підготовки текстів

Ключові характеристики:

- Підтримка 5+ мов тестування
- Генерація унікальних текстів для кожної сесії
- Анонімний доступ для гостей з автоматичним ID
- Фіксований час тестування (60 секунд)

Заходи безпеки:

- Валідація вхідних даних
- Обмеження кількості підключень
- Захист від DDoS-атак

Система коментарів

Модель даних:

```
class Comment(models.Model):
    bb = models.ForeignKey(Bb, on_delete=models.CASCADE)
    author = models.CharField(max_length=30)
    content = models.TextField()
    is_active = models.BooleanField(default=True)
    created_at = models.DateTimeField(auto_now_add=True)
```

Лістинг 3.6 Модель даних

Особливості реалізації:

1. Двоетапна модерація:

- Автоматична фільтрація спаму
- Ручне підтвердження адміністратором

2. Капча для гостей:

- Інтеграція з Django-simple-captcha
- Налаштування складності зображень

3. Управління через адмін-панель:

- Масові операції з коментарями
- Фільтрація за статусом активності
- Пошук за вмістом та автором

Продуктивність:

- Оптимізовані запити до БД
- Кешування популярних обговорень
- Середній час відгуку: 80-100мс

Інтеграція компонентів:

Обидва модулі пов'язані з основним функціоналом:

- Мультиплеєр використовує ту саму систему підрахунку статистики
- Коментарі інтегровані з профілями користувачів
- Спільна система автентифікації та прав доступу

Переваги архітектури:

- Модульність - можливість відключення компонентів
- Мінімальні залежності між системами
- Узгоджений інтерфейс користувача

3.4 Висновок до третього розділу

У другому розділі кваліфікаційної роботи було розроблено та впроваджено веб-систему для тренування швидкості друку на основі фреймворку Django. Запропонована архітектура та реалізовані модулі (аутентифікація, тестування, статистика, мультиплеєр) продемонстрували високу ефективність у забезпеченні стабільної роботи, безпеки та зручності для користувачів.

Ключові досягнення:

- Система забезпечує точний підрахунок показників (WPM, точність) за допомогою оптимізованих алгоритмів.
- Гнучкий механізм аутентифікації (сесії + JWT) дозволяє інтегруватися з різними клієнтами (веб, мобільні додатки).
- Модульність архітектури дає можливість розширювати функціонал (наприклад, додавання нових типів тестів).

Перспективи розвитку:

- Впровадження AI-аналізу помилок для персоналізованих тренувань.
- Додавання турнірного режиму в мультиплеєрі.
- Підтримка додаткових мов та спеціалізованих клавіатурних розкладок.

Розроблена система є готовим рішенням для навчання та може бути впроваджена в навчальних закладах або як самостійний онлайн-сервіс.

РОЗДІЛ 4. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Надзвичайні ситуації екологічного характеру: загрози та запобіжні заходи

Сучасний світ стикається з численними екологічними викликами, які становлять загрозу не лише для природних екосистем, а й для стабільного функціонування ключових секторів економіки, зокрема інформаційних технологій. Надзвичайні ситуації екологічного характеру – це катастрофічні події, спричинені природними або антропогенними факторами, що призводять до серйозних порушень у довкіллі, масштабних матеріальних збитків, а іноді й до втрати людських життів. Для ІТ-індустрії, яка традиційно вважається відносно незалежною від зовнішніх умов, такі загрози можуть мати далекосяжні наслідки, оскільки вони впливають як на фізичну інфраструктуру, так і на здоров'я персоналу, стабільність енергопостачання та навіть на глобальні ланцюги поставок.

Основні екологічні загрози для ІТ-сфери:

1. Кліматичні катастрофи та їх вплив на інфраструктуру

Зміни клімату призводять до збільшення екстремальних погодних явищ, таких як урагани, повені, тривалі періоди спеки або аномальних морозів. Для ІТ-компаній це означає ризик пошкодження дата-центрів, перебоїв у роботі серверів через перегрів або відключення електроенергії, а також ускладнення логістики через зруйновані дороги або закриті аеропорти. Наприклад, у 2021 році повені в Німеччині призвели до серйозних пошкоджень інфраструктури, що вплинуло на роботу європейських ІТ-провайдерів.

2. Пожежі та їх руйнівний вплив на дата-центри

Однією з найбільш критичних загроз для ІТ-інфраструктури є пожежі, які можуть знищити цілі центри обробки даних разом із зберіганою інформацією. Яскравий приклад – пожежа в дата-центрі OVH у Франції у березні 2021 року, внаслідок якої було втрачено дані тисяч клієнтів, включаючи урядові установи та

великі корпорації. Така подія не лише призводить до фінансових втрат, але й підриває довіру до компанії, що може мати довгострокові наслідки для бізнесу.

3. Енергетична нестабільність та проблеми електропостачання

ІТ-інфраструктура, особливо великі дата-центри, потребує стабільного електропостачання. Відключення світла, спричинені аваріями на енергосистемах, природними катастрофами або навіть кібератаками, можуть паралізувати роботу компаній. Наприклад, у 2022 році в Україні через ракетні обстріли енергооб'єктів відбувалися масштабні блекаути, що вплинуло на роботу багатьох ІТ-фірм, які були змушені переходити на резервні джерела живлення або тимчасово призупиняти свою діяльність.

4. Забруднення довкілля та його вплив на здоров'я працівників

ІТ-спеціалісти часто працюють у великих містах, де рівень забруднення повітря може бути високим. Токсичні речовини, такі як дрібнодисперсні частки (PM2.5), діоксид азоту та сірководень, негативно впливають на здоров'я, знижуючи продуктивність і збільшуючи кількість хворобливих днів. Наприклад, у 2020–2021 роках Київ неодноразово потрапляв у зону інтенсивного диму через лісові пожежі, що призвело до зростання захворювань серед офісних працівників, включаючи ІТ-фахівців.

5. Глобальні біологічні загрози (пандемії)

Пандемія COVID-19 стала випробуванням для всієї світової економіки, і ІТ-сектор не став винятком. Хоча багато компаній швидко перейшли на віддалену роботу, це не уникнуло серйозних проблем: втрата продуктивності через хвороби, психологічні навантаження, порушення комунікації між командами. Крім того, деякі фірми, які не мали ефективних кризових стратегій, збанкрутували, незважаючи на наявність цифрових інструментів.

Стратегії запобігання та мінімізації наслідків:

1. Державний рівень

Розвиток систем моніторингу екологічних ризиків (метеорологічні попередження, контроль якості повітря). Інвестиції в стійку енергетичну інфраструктуру (відновлювані джерела енергії, резервні мережі). Розробка

нормативних документів, що регулюють безпеку дата-центрів (протипожежні стандарти, вимоги до розташування).

2. Рівень компаній

Впровадження планів безперервності бізнесу (BCP), які передбачають дії у разі катастроф. Використання хмарних технологій для розподіленого зберігання даних і захисту від втрати інформації. Забезпечення альтернативних джерел енергії (генератори, сонячні панелі) для критичної інфраструктури. Психологічна підтримка працівників та медичне страхування для зменшення впливу стресу та хвороб.

3. Індивідуальний рівень

Навчання співробітників діям у надзвичайних ситуаціях (евакуація, перша допомога). Використання засобів індивідуального захисту (наприклад, респіраторів у разі забруднення повітря). Підтримання здорового способу життя для зміцнення імунітету та стресостійкості.

4.2 Оцінка розробленого технологічного процесу щодо умов безпеки, втомлюваності та продуктивності праці

Сучасні офісні умови праці, особливо в сфері інформаційних технологій, потребують ретельного аналізу з точки зору безпеки, впливу на здоров'я працівників та ефективності виконання трудових обов'язків. Хоча робота ІТ-фахівця не пов'язана з фізично важкими умовами чи небезпечними виробничими процесами, тривале перебування в статичній позі, постійне зорове навантаження, електромагнітне випромінювання та психоемоційні фактори створюють суттєві професійні ризики, які можуть призвести до захворювань опорно-рухового апарату, синдрому комп'ютерного зору, хронічного стресу та інших патологій.

Комплексна оцінка умов праці в офісі: оцінка умов праці ІТ-спеціалістів повинна базуватися на міждисциплінарному підході, що враховує фізіологічні, психологічні та організаційні аспекти. Відповідно до чинних нормативних

документів, зокрема Державних санітарних норм та міжнародних стандартів (ISO, EN), основними критеріями оцінки є:

1. Безпека праці

Безпечні умови праці передбачають дотримання ряду нормативних вимог, спрямованих на збереження здоров'я працівників.

Мікроклімат повинен відповідати санітарним нормам: температура в офісі має підтримуватися на рівні 20–24 °С, відносна вологість повітря – 40–60%, а швидкість руху повітря – не більше 0,1–0,2 м/с. Недостатня вологість призводить до пересихання слизових оболонок, а надмірна – до дискомфорту та розвитку грибків.

Електромагнітне випромінювання від моніторів та іншого обладнання не повинно перевищувати 25 Вт/м². Хоча сучасні LCD-екрани мають мінімальне випромінювання, тривале опромінення може впливати на нервову систему.

Освітленість робочої зони повинна становити не менше 300 лк, бути рівномірною та позбавленою відблисків. Неправильне освітлення призводить до зорової втоми, головного болю та зниження концентрації.

2. Втомиюваність та ергономіка

Одним із ключових факторів, що впливають на працездатність ІТ-фахівців, є фізіологічна та психологічна втома.

Тривалість безперервної роботи за комп'ютером не повинна перевищувати 2 години, після чого необхідно робити перерви 10–15 хвилин для відпочинку очей та м'язів [35].

Ергономіка робочого місця (згідно з ДСТУ ISO 9241-210:2021)[36] передбачає:

Регульоване крісло з підтримкою попереку та зручними підлокітниками;

Розташування монітора на рівні очей (верхній край екрана – на рівні або трохи нижче лінії погляду);

Правильне розміщення клавіатури та миші для запобігання синдрому карпального каналу.

Психоемоційне навантаження (стрес, дедлайни, багатозадачність) може

призводити до синдрому професійного вигорання, тому важливо впроваджувати методи релаксації та психологічної розвантаження.

3. Продуктивність праці

Ефективність роботи ІТ-спеціалістів залежить не лише від індивідуальних якостей, але й від організації робочого процесу.

Технічне забезпечення (потужні комп'ютери, якісні монітори, швидкий інтернет) значно підвищує продуктивність.

Оптимізація робочих процесів за допомогою технологічних карт, систем управління проектами (Jira, Trello) та автоматизації рутинних завдань дозволяє зменшити час на виконання операцій.

Гнучкий графік та можливість віддаленої роботи покращують баланс між роботою та особистим життям, що позитивно впливає на мотивацію.

Методи оцінки умов праці

Для всебічного аналізу використовуються:

Інструментальні вимірювання – за допомогою спеціальних приладів (люксметрів, шумомірів, газоаналізаторів) визначаються рівні освітленості, шуму, температури, вологості та концентрації CO₂.

Соціологічні дослідження – анкетування працівників, фотографія робочого дня, оцінка рівня стресу за шкалою Холмса-Раге.

Експертні оцінки – аудит безпеки праці, аналіз ергономіки робочих місць, оцінка професійних ризиків.

Шляхи оптимізації умов праці

Для покращення умов праці в офісі ІТ-компаній можна запровадити такі заходи:

Технічні заходи

Встановлення сучасного енергоефективного обладнання зі зниженим рівнем випромінювання.

Використання антиблікових екранів та регульованого освітлення.

Впровадження систем клімат-контролю та очищення повітря.

Застосування програм моніторингу здоров'я (Prometheus Health, Welltory)

для відстеження стану працівників.

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи було розроблено веб-платформу для навчання сліпого методу набору тексту, яка забезпечує точне вимірювання швидкості та точності друку, адаптивне навчання та захист від шахрайства.

У першому розділі кваліфікаційної роботи освітнього рівня «Бакалавр»:

- Обґрунтовано актуальність теми навчання сліпому друку у цифрову епоху.
- Проаналізовано сучасні платформи, виявлено їх недоліки.
- Визначено вимоги до функціоналу системи та її архітектурні обмеження.
- Запропоновано технологічні рішення для реалізації платформи на основі Django.

У другому розділі:

- Спроектовано архітектуру веб-платформи з поділом на клієнтську та серверну частини.
- Розроблено систему автентифікації, профілювання користувачів та захисту даних.
- Реалізовано модулі навчання, тестування, статистики та рейтингів із підтримкою адаптивних алгоритмів.
- Визначено ключові показники ефективності системи та її технічні обмеження.

У третьому розділі:

- Реалізовано систему обліку користувачів, збереження історії тренувань, генерації вправ і візуалізації результатів.
- Проведено тестування працездатності платформи, зокрема перевірку точності обчислення метрик WPM та коректності

статистики.

- Розроблено додаткові модулі: мультиплеєрний режим та система коментарів з модерацією.

У розділі «Безпека життєдіяльності та охорона праці»:

- Висвітлено потенційні загрози екологічного характеру для ІТ-сфери.
- Надано рекомендації з підвищення ергономіки робочого місця та забезпечення електробезпеки.

Розроблена платформа може бути використана як у навчальних закладах, так і для індивідуального користування. Її архітектура дозволяє масштабування, інтеграцію з мобільними додатками та подальший розвиток (наприклад, за допомогою технологій штучного інтелекту).

ПЕРЕЛІК ДЖЕРЕЛ

1. Mudryk I., Petryk M. Hybrid artificial intelligence systems for complex neural network analysis of abnormal neurological movements with multiple cognitive signal nodes. // 2020 IEEE Third International Conference on Data Stream Mining & Processing (DSMP) : Conference, Lviv, 21–25 August 2020. – P. 108–111. DOI: 10.1109/DSMP47368.2020.9204341.
2. Bryk O., Mudryk I., Holubovskyi M., Stoianov Y. Machine learning models and methods aspects of processing unstructured data. // Proceedings of the 1st International Workshop on Bioinformatics and Applied Information Technologies (BAIT 2024). – Zboriv, Ukraine, 2024. – P. 64–74.
3. Дистанційний курс «Кваліфікаційні роботи бакалаврів» сайту дистанційного навчання ТНТУ [Електронний ресурс]. – Режим доступу: URL: <https://dl.tntu.edu.ua/bounce.php?course=5329>
4. Голинська О., Мудрик І. Роль CRM-системи у сучасних бізнес-процесах. // Матеріали X наук.-техн. конф. «Інформаційні моделі, системи та технології» ТНТУ ім. І. Пулюя, 2022. – С. 113.
5. Яковенко І., Мудрик І. Орієнтований на дані дизайн. Аналіз та переваги. // Матеріали IX наук.-техн. конф. «Інформаційні моделі, системи та технології» ТНТУ ім. І. Пулюя, 2022. – С. 181–182.
6. Адлер І. Ю. Django: розробка веб-застосувань на Python. – К.: Фабула, 2020. – 368 с.
7. Holovaty A., Kaplan-Moss J. The Definitive Guide to Django: Web Development Done Right. – Apress, 2009. – 447 p.
8. Назарук І. Django 3. Практичний курс з програмування. – Львів: Аверс, 2021. – 296 с.
9. Краснов В. Django REST Framework: створення API сервісів. – Харків: Фактор, 2022. – 210 с.

10. Krug S. Don't Make Me Think: A Common Sense Approach to Web Usability. – New Riders, 2014. – 216 p.
11. Norman D. The Design of Everyday Things. – Basic Books, 2013. – 368 p.
12. Bootstrap. Документація [Електронний ресурс]. – Режим доступу: <https://getbootstrap.com/>
13. MDN Web Docs. HTML, CSS, JS reference [Електронний ресурс]. – <https://developer.mozilla.org/>
14. Гнатюк С. Веб-технології: від HTML до React. – Київ: Ліра-К, 2023. – 288 с.
15. Шевченко С. UX-дизайн: посібник з проектування зручних інтерфейсів. – Харків: Ранок, 2022. – 198 с.
16. Мороз О. Швидкий набір тексту: методика навчання. – Тернопіль: Навчальна книга – Богдан, 2020. – 120 с.
17. Погорілий Ю. Психофізіологічні аспекти комп'ютерного набору. – Львів: Світ, 2019. – 132 с.
18. International Typing Speed Database [Електронний ресурс] – <https://www.typingtest.com/>
19. Слюсаренко П. Практикум зі створення тренажерів сліпого набору. – К.: Академія, 2021. – 104 с.
20. Пелешко Д. Адаптивні освітні системи. – Львів: ЛНУ, 2018. – 212 с.
21. Валігура О. Moodle та LMS-системи: сучасні підходи. – Івано-Франківськ: Нова школа, 2020. – 144 с.
22. Scikit-learn documentation [Електронний ресурс] – <https://scikit-learn.org/>
23. Python official documentation [Електронний ресурс] – <https://docs.python.org/3/>
24. FastAPI documentation [Електронний ресурс] – <https://fastapi.tiangolo.com/>
25. Філіпенко В. Безпечна розробка вебзастосунків. – Харків: ВНДІ,

2021. – 256 с.

26. OWASP. Top 10 Security Risks [Електронний ресурс] – <https://owasp.org/www-project-top-ten/>

27. Звіт Deloitte: Освітні IT-системи в Україні. – Київ, 2023. – 48 с.

28. Codecademy. Learn Django [Електронний ресурс] – <https://www.codecademy.com/>

29. Udemy: Build a Typing Test App in Django – [Електронний курс] – <https://udemy.com/>

30. MIT OpenCourseWare: Human-Computer Interaction [Електронний ресурс] – <https://ocw.mit.edu/>

31. Василюк М. Архітектура програмних систем. – Київ: КНУ, 2022. – 280 с.

32. Мороз І. Основи розробки front-end інтерфейсів. – Львів: Сполом, 2023. – 188 с.

33. Яшук О. Адаптивна верстка веб-сайтів. – Харків: Основа, 2021. – 132 с.

34. ISO/IEC 25010:2011 – Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models.

35. Швидкісний набір: TypingClub методика [Електронний ресурс]. – <https://www.typingclub.com/>

36. Наказ МОЗ України № 246 від 15.04.2020 «Про затвердження норм тривалості роботи з ПЕОМ» // Офіційний вісник України. – 2020. – № 34. – Ст. 1125.

37. ДСТУ ISO 9241-210:2021. Ергономіка взаємодії людини та системи : наказ Мінекономіки України від 12.03.2021 № 156.

38. Андрейчук Н. І., Кіт Ю. В., Шибанов С. В. Охорона праці : навч. посіб. – Львів : Видавництво Львівської політехніки, 2021. – 276

39. Жидецький В. Ц. Охорона праці користувачів комп'ютерів : підручник. – Львів : Афіша, 2020. – 176 с.

40. Мельник Л. Г., Ковальчук О. І. Безпека життєдіяльності : підручник. – Київ : Центр учбової літератури, 2022. – 320 с.
41. ДСТУ 8604:2015. Дизайн і ергономіка. Робоче місце для виконання робіт у положенні сидячи. Загальні ергономічні вимоги : наказ Мінекономрозвитку України від 25.12.2015 № 1433.
42. Міністерство цифрової трансформації України. Рекомендації з організації робочого місця ІТ-фахівця [Електронний ресурс]. – 2023. – Режим доступу: <https://thedigital.gov.ua/ergonomics> (дата звернення: 15.06.2024).
43. Державна служба України з питань праці. Методичні рекомендації з оцінки умов праці [Електронний ресурс]. – 2022. – Режим доступу: <https://dsp.gov.ua/methodical-materials> (дата звернення: 15.06.2024).
44. Кафедра безпеки життєдіяльності КПІ. Збірник нормативних документів з охорони праці в ІТ-галузі. – Київ : КПІ, 2023. – 120

ДОДАТКИ

Додаток А. Диск з роботою