Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

“ЗАТВЕРДЖЕНО”

Завідувач кафедри

\_\_\_\_\_\_\_\_\_\_\_ Едуард ЖАРІКОВ

“\_\_\_” \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2025 р.

**Вебзастосунок для агрегації та інтелектуального аналізу новинного контенту**

**Програма та методика тестування**

КПІ.ІП-1110.045440.04.51

“ПОГОДЖЕНО”

Керівник проєкту:

\_\_\_\_\_\_\_\_\_\_\_\_ Тетяна ШУЛЬКЕВИЧ

|  |  |
| --- | --- |
| Нормоконтроль: | Виконавець: |
| \_\_\_\_\_\_\_\_\_\_\_ Катерина ЛІЩУК | \_\_\_\_\_\_\_\_\_\_ Олександра ДРУЗЕНКО |

Київ – 2025

**ЗМІСТ**

[1 ОБ’ЄКТ ВИПРОБУВАНЬ 3](#_Toc199686874)

[2 МЕТА ТЕСТУВАННЯ 4](#_Toc199686875)

[3 МЕТОДИ ТЕСТУВАННЯ 5](#_Toc199686876)

[4 ЗАСОБИ ТА ПОРЯДОК ТЕСТУВАННЯ 6](#_Toc199686877)

# ОБ’ЄКТ ВИПРОБУВАНЬ

На основі аналізу дипломного проєкту, об'єктом випробування є вебзастосунок TaskTime для обліку робочого часу персоналу та керування завданнями, розроблений з використанням фреймворку Angular 19 для клієнтської частини та ASP.NET Core 9 для серверної частини. Застосунок призначений для роботи в сучасних веббраузерах з підтримкою TypeScript та функціонує під управлінням операційних систем сімейства Windows та Unix з інтеграцією до хмарної платформи Microsoft Azure та зовнішньої системи Azure DevOps.

# МЕТА ТЕСТУВАННЯ

На основі аналізу дипломного проєкту, метою тестування є:

* перевірка коректності збереження та відображення даних користувачів, записів часу, завдань, заявок на відпустки та організаційної структури в базі даних SQL Server;
* перевірка правильності роботи програмного забезпечення відповідно до функціональних вимог системи обліку робочого часу та керування завданнями;
* перевірка сумісності вебзастосунку з останніми версіями сучасних браузерів (Chrome, Firefox, Safari, Edge);
* перевірка сумісності застосунку з різними операційними системами (Windows, Linux, macOS);
* виявлення помилок, недоліків та проблем функціонування з метою їх усунення;
* перевірка зручності та інтуїтивності графічного інтерфейсу користувача;
* тестування швидкодії системи та відповідності нефункціональним вимогам;
* перевірка коректності інтеграції з Azure DevOps для синхронізації завдань;
* тестування системи безпеки та контролю доступу на основі ролей користувачів.

3 МЕТОДИ ТЕСТУВАННЯ

На основі аналізу дипломного проєкту, для тестування вебзастосунку TaskTime обрано наступні методи тестування відповідно до піраміди тестування:

* Статичне тестування – аналіз коду C# (ASP.NET Core) та TypeScript (Angular) компонентів на відповідність стандартам програмування, перевірка архітектури та документації за допомогою SonarQube.
* Динамічне тестування – виконання програми з різними наборами даних для перевірки коректності роботи модулів обліку часу, управління завданнями та взаємодії з базою даних SQL Server.
* Функціональне тестування – перевірка відповідності реальної поведінки системи функціональним вимогам (облік робочого часу, управління заявками, синхронізація з Azure DevOps).
* Системне тестування – тестування повної інтеграції компонентів: Angular клієнта, ASP.NET Core API, бази даних SQL Server та зовнішнього сервісу Azure DevOps.
* Мануальне тестування – ручна перевірка користувацького інтерфейсу та сценаріїв використання для різних ролей користувачів (працівник, HR-менеджер, адміністратор).
* Тестування "чорної скриньки" – перевірка вихідних результатів веб-інтерфейсу при різних вхідних параметрах фільтрації та операціях CRUD.
* Тестування "сірої скриньки" – тестування алгоритмів автентифікації JWT, авторизації на основі ролей та механізмів синхронізації даних.

# 4 ЗАСОБИ ТА ПОРЯДОК ТЕСТУВАННЯ

На основі аналізу дипломного проєкту, під час проведення тестування будуть використовуватись наступні допоміжні засоби:

* веббраузери: Google Chrome, Mozilla Firefox, Safari, Microsoft Edge різних версій;
* інструменти розробника браузерів для аналізу продуктивності та помилок;
* SQL Server Management Studio для перевірки цілісності даних;
* інструменти моніторингу платформи Microsoft Azure для аналізу навантаження;
* емулятори мобільних пристроїв у браузерах;
* Postman для тестування REST API та HTTP-запитів;
* SonarQube для статичного аналізу якості коду.

Порядок проведення тестування буде наступним:

* статичне тестування коду ASP.NET Core та Angular компонентів за допомогою SonarQube;
* динамічне тестування на відповідність функціональним вимогам (облік робочого часу, управління завданнями, обробка заявок);
* системне тестування інтеграції з базою даних SQL Server та Entity Framework Core;
* тестування взаємодії з зовнішнім API Azure DevOps;
* тестування сумісності з різними браузерами та версіями;
* тестування на мобільних пристроях з різною роздільною здатністю екрана;
* тестування продуктивності та швидкодії системи;
* тестування зручності використання інтерфейсу для різних ролей користувачів;
* тестування безпеки JWT-автентифікації та авторизації на основі ролей;
* тестування захисту від основних типів атак (SQL-ін'єкції, XSS, CSRF);
* регресійне тестування після внесення змін.