СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Навчально-науковий інститут (факультет) інформаційних технологій та електроніки

Кафедра інформаційних технологій та програмування

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної випускної роботи

освітній ступінь бакалавр
спеціальність 126 „Інформаційні системи та технології”
(шифр і назва спеціальності)

на тему „Аналіз сучасних методів автентифікації користувачів в інформаційних системах. Паролі, токени, біометрія, 2FA.”

Виконала: студент групи ІСТ-21д _____ (підпис) В. С. Антонов (ініціали і прізвище)

Керівник _____ (підпис) О. С. Дюбанов (ініціали і прізвище)

Завідувач кафедри _____ (підпис) О.І. Захожай (ініціали і прізвище)

Рецензент ____Захожай О.І____

Київ – 2025

СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Навчально-науковий інститут (факультет) інформаційних технологій та електроніки
Кафедра інформаційних технологій та програмування

Освітній ступінь бакалавр
спеціальність 126 „Інформаційні системи та технології”
(шифр і назва спеціальності)

ЗАТВЕРДЖУЮ
Завідувач кафедри ІТП
_____ Захожай О.І.
“___” _____ 2025 року

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ ВИПУСКНУ РОБОТУ СТУДЕНТУ

Антонов Віталій Сергійович
(прізвище, ім’я, по батькові)

1. Тема роботи: Аналіз сучасних методів автентифікації користувачів в інформаційних системах. Паролі, токени, біометрія, 2FA

Керівник роботи Дюбанов Олексій Сергійович, к.е.н., доцент
(прізвище, ім’я, по батькові, науковий ступінь, вчене звання)

затверджені наказом університету від “27” травня 2025року № 67/15.15-С

2. Строк подання студентом роботи 16.06.2025 р.

3. Вихідні дані до роботи: матеріали переддипломної практики

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно огляд та аналіз сучасних методів автентифікації користувачів в інформаційних системах, зокрема: паролі, токени, біометрія, двофакторна автентифікація (2FA), OTP. Розкриття технічних принципів біометричної ідентифікації, опис основ алгоритмів автентифікації, приклади архітектури та реалізації протоколу авторизації системи на базі OTP-кодів. Тестування, аналіз результатів і оцінка ефективності впровадження. Додатково розглянуто заходи з охорони праці та безпечної роботи з комп'ютерною технікою.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслеників) архітектура інформаційної системи автентифікації, інтерфейс веб-системи входу з OTP, скріншоти генерації OTP, код серверної частини авторизації, результати тестування.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 01.04.2025

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання кваліфікаційної випускної роботи	Строк виконання етапів	Примітка
1	Отримання завдання та збір матеріалів	05.05.25 – 10.05.25	Виконано
2	Огляд предметної області	11.05.25 –13.05.25	Виконано
3	Формулювання вимог до системи та розробка основних алгоритмів	14.05.25 – 17.05.25	Виконано
4	Розробка практичної частини	18.05.25– 24.05.25	Виконано
5	Розробка інтерфейсу користувача	25.05.25– 28.05.25	Виконано
6	Оформлення пояснювальної записки	29.05.25– 05.06.25	Виконано
7	Підготовка та подання роботи до захисту	06.06.25– 16.06.25	Виконано
8	Підготовка презентації та доповіді	16.06.25 – 20.06.25	Виконано

Студент _____ В. С. Антонов
підпис (ініціали і прізвище)

Керівник роботи _____ О. С. Дюбанов
підпис (ініціали і прізвище)

РЕФЕРАТ

Пояснювальна записка до дипломного проекту бакалавра: 66 сторінок, 20 рис., 4 табл., 20 джерел посилань, 5 додатків на 10 сторінках.
У дипломному проекті проведено всебічний аналіз сучасних методів автентифікації користувачів в інформаційних системах. Розглянуто традиційні (парольні) та сучасні засоби автентифікації: токени, біометричні технології (зокрема відбитки пальців), а також методи двофакторної (2FA) та багатофакторної автентифікації (MFA).

Особливу увагу приділено практичному впровадженню системи автентифікації з одноразовими паролями (OTP), які генеруються за допомогою застосунків типу Google Authenticator. Розроблено та протестовано прототип веб-застосунку на базі Flask (Python), який реалізує авторизацію з використанням OTP-коду. Проведено тестування на коректність роботи, зручність використання та стійкість до типових атак.

Також у дипломі розглянуто порівняльну оцінку переваг і недоліків кожного з підходів до автентифікації, проаналізовано рівень безпеки, складність реалізації та вплив на користувацький досвід.

Проект включає опис вимог до охорони праці при роботі з комп’ютером, дотримання ергономіки, пожежної та електробезпеки, нормативних показників мікроклімату, шуму, освітлення.

Дипломна робота відображає сучасний підхід до забезпечення інформаційної безпеки на рівні доступу до систем, і має практичну цінність для організацій, що працюють з конфіденційними даними.

АВТЕНТИФІКАЦІЯ, 2FA, OTP, ПАРОЛІ, БІОМЕТРІЯ, ТОКЕНИ, ЗАХИСТ ДАНИХ, ОХОРОНА ПРАЦІ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ.

ЗМІСТ

ВСТУП 7
РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ АВТЕНТИФІКАЦІЇ В ІНФОРМАЦІЙНИХ СИСТЕМАХ 8
1.1 Поняття ідентифікації та автентифікації 8
1.2 Загрози, пов’язані з неавторизованим доступом 8
1.3 Парольна автентифікація 9
1.4 Одноразові паролі (OTP) 11
1.5 Біометричні методи автентифікації 12
1.6 Багатофакторна автентифікація (2FA/MFA) 13
1.7 Порівняльний аналіз методів автентифікації 15
Висновки до розділу 1 17

РОЗДІЛ 2. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ 18
2.1 Актуальність теми 18
2.2 Аналіз існуючих рішень 18
2.3 Проблеми реалізації автентифікації 21
2.4 Постановка задачі 21
Висновки до розділу 2 21

РОЗДІЛ 3. РОЗРОБКА СИСТЕМИ автентифікацЇ 22
3.1 Вибір інструментів та технологій 22
3.2 Створення базової форми логіну 25
3.3 Генерація та перевірка OTP-коду 27
3.4 Інтеграція з Google Authenticator 30
3.5 Тестування системи автентифікації 33
3.6 Скріншоти роботи системи 34
Висновки до розділу 3 38

РОЗДІЛ 4. ОХОРОНА ПРАЦІ 39
4.1 Особливості охорони праці в сфері кібербезпеки 39
4.2 Умови безпечної роботи з ПЕОМ 41
4.3 Вимоги до електробезпеки 42
4.4 Пожежна безпека 43
4.5 Вимоги до мікроклімату, освітлення та шумух 45
4.6 Заходи безпеки у надзвичайних ситуаціях 47
Висновки до розділу 4 48

РОЗДІЛ 5. ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ ДИПЛОМНОГО ПРОЄКТУ 49
5.1 Мета та завдання економічного обгрунтування 49
5.2 Витрати на розробку програмного забезпечення 49

5.3 Амортизаційні витрати 50

5.4 Економічна ефективність від впровадження системи 50
Висновки до розділу 5 51

ЗАГАЛЬНІ ВИСНОВКИ ДО ДИПЛОМНОГО ПРОЄКТУ 53

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ 55

ДОДАТКИ 57
Додаток А – Фрагмент коду генерації одноразового пароля (OTP) 57
Додаток Б – Скріншоти інтерфейсу авторизації з OTP 59
Додаток В – Таблиця тестування результатів роботи системи 64
Додаток Г – Схема логіки роботи 2FA (спрощена блок-схема) 65
Додаток Д – HTML-шаблон login.html 66

ВСТУП

У сучасному світі цифрових технологій питання безпечного доступу до інформаційних систем набуває особливої актуальності. Щодня мільйони користувачів отримують доступ до різноманітних сервісів — від банківських додатків до корпоративних платформ — і кожен такий доступ потребує надійної автентифікації. Зростаюча кількість витоків даних, шахрайських атак та соціальної інженерії свідчить про те, що традиційних методів, таких як прості паролі, вже недостатньо.

Зараз автентифікація — це не просто перевірка пароля, а ціла система засобів, яка має враховувати як зручність для користувача, так і стійкість до атак. Саме тому все частіше застосовуються токени, одноразові паролі (OTP), багатофакторна автентифікація (2FA, MFA), а також біометричні технології: відбитки пальців, розпізнавання обличчя, голосу тощо.

Метою цієї дипломної роботи є дослідження та порівняння сучасних методів автентифікації користувачів, аналіз їх переваг, недоліків, безпеки, складності впровадження та зручності у використанні. У рамках проекту також буде розроблено приклад модуля автентифікації з OTP-кодом на основі Google Authenticator, який дозволяє підвищити рівень захисту без необхідності значних фінансових витрат.

Актуальність теми полягає в тому, що безпечна автентифікація — це основа кібербезпеки будь-якої сучасної організації. Якщо механізм доступу до системи ненадійний, жодні шифрування чи антивіруси не врятують від зловмисника. Саме тому грамотний підбір і впровадження методів автентифікації — це завдання, яке має практичне значення не тільки для IT-фахівців, а й для бізнесу в цілому.

У цій роботі розглянуто теоретичні аспекти автентифікації, практичні приклади її реалізації, а також аспекти охорони праці при роботі з інформаційними системами.

РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ АВТЕНТИФІКАЦІЇ В ІНФОРМАЦІЙНИХ СИСТЕМАХ

1.1 Поняття ідентифікації та автентифікації

Ідентифікація та автентифікація є базовими поняттями у сфері інформаційної безпеки. Ідентифікація — це процес визначення суб’єкта, що намагається отримати доступ до системи, тобто вказання ким є користувач. Найчастіше це виконується шляхом введення імені користувача (login, ID, email тощо).

Автентифікація — це перевірка достовірності цієї інформації. Інакше кажучи, система повинна впевнитися, що особа, яка ввела певне ім’я користувача, дійсно є тією, за кого себе видає. Це досягається через надання пароля, токена, відбитка пальця або іншого підтвердження.

Процес автентифікації може бути реалізований за допомогою:

- **Знань** (щось, що знає користувач: пароль, PIN код);
- **Володіння** (щось, що має користувач: смарт-карта, мобільний пристрій, токен);
- **Біометричних ознак** (щось, чим є користувач: відбиток пальця, розпізнавання обличчя, голос) (2).

Надійна автентифікація є першим і одним з найважливіших етапів захисту доступу до будь-якої інформаційної системи. Без неї неможливо забезпечити ефективну роботу політик доступу, а ризик витоку даних значно зростає.

1.2. Загрози, пов’язані з неавторизованим доступом

Ненадійна автентифікація або її повна відсутність — це одна з найпоширеніших причин порушень інформаційної безпеки. Зловмисники, отримавши

несанкціонований доступ до системи, можуть викрасти, змінити або знищити важливі дані, а також заблокувати доступ до ресурсів для реальних користувачів.

Існує кілька основних загроз, які пов’язані з неавторизованим доступом:

- **Підбір пароля (Brute Force)** — автоматичне перебира́ння всіх можливих варіантів паролів до тих пір, поки не буде знайдено правильний.
- **Фішинг (Phishing)** — метод шахрайства, при якому користувача обманом змушують ввести свої облікові дані на підробленому сайті.
- **Соціальна інженерія** — вплив на людину, щоб вона сама надала пароль або іншу конфіденційну інформацію (наприклад, телефоном, через месенджер).
- **Використання викрадених баз даних** — багато паролів "зливаються" під час зламів інших сервісів. Якщо користувач використовує один і той самий пароль скрізь — система легко зламується.
- **Використання сесій сторонніх осіб** — іноді після авторизації зловмисник перехоплює "сесію" користувача, не знаючи навіть його пароля.
- **Шкідливе ПЗ (кейлогери, віруси)** — фіксує введені з клавіатури паролі, робить знімки екрана або перехоплює трафік.

Ці методи постійно вдосконалюються, тому звичайна авторизація за логіном і паролем вже не вважається достатньою. Саме тому в сучасних системах застосовуються додаткові рівні захисту, про які буде мова у наступних підрозділах (1).

1.3. Парольна автентифікація

Найпоширенішим способом автентифікації залишається введення логіну та пароля. Пароль — це секретна комбінація символів, відома лише користувачу. Після введення цієї інформації система порівнює її з даними у базі — і дозволяє або забороняє доступ.

Надійність такого методу напряму залежить від складності пароля та захисту бази даних, де він зберігається. Слабкі паролі на кшталт «123456» або «qwerty» дуже легко зламати за допомогою перебору або злитих баз. Крім того, користувачі часто використовують один і той самий пароль для декількох сервісів, що також підвищує ризики.

Паролі можуть зберігатися:

- у відкритому вигляді (що небезпечно),
- або у вигляді **хешу** (рекомендується), що підвищує безпеку навіть у разі витоку бази даних.

Для захисту паролів рекомендується:

- встановлювати **мінімальну довжину** (8+ символів);
- використовувати **комбінації** літер, цифр і спецсимволів;
- **змінювати пароль** регулярно;
- не передавати пароль стороннім особам (5).

Приклад інтерфейсу для введення пароля:

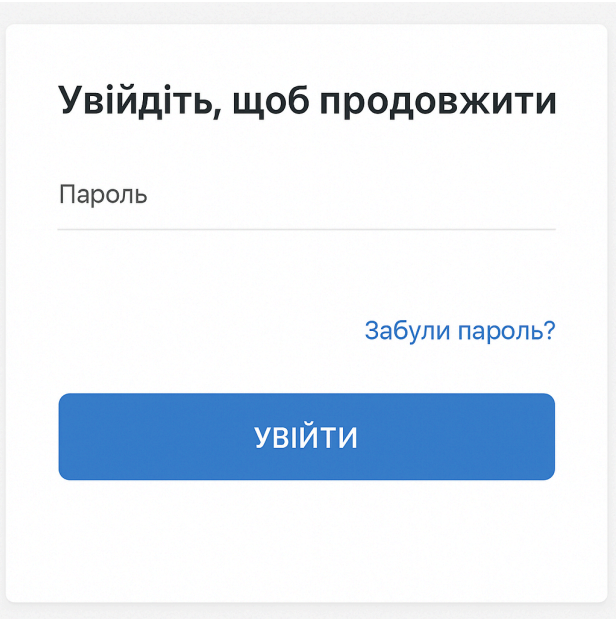


Рисунок 1.1 – Приклад інтерфейсу для введення паролю

1.4. Одноразові паролі (OTP)

Одноразовий пароль (OTP — One-Time Password) — це спеціальний код, який дійсний лише для однієї сесії входу або транзакції. Такий підхід значно знижує ймовірність зламу, адже навіть якщо зловмисник дізнається OTP-код, він уже буде недійсним.

OTP може надходити користувачу:

- у вигляді SMS або e-mail повідомлення;
- через мобільний додаток, наприклад, Google Authenticator;
- з використанням апаратного токена.

OTP генерується на основі **тайм-коду** (TOTP — Time-Based One-Time Password) або **подій** (HOTP — HMAC-Based OTP). Алгоритм гарантує унікальність коду на кожну нову сесію, що суттєво підвищує рівень безпеки.

Застосування одноразових паролів дозволяє уникнути більшості атак, пов’язаних із фішингом та перехопленням облікових даних.

Приклад інтерфейсу для введення OTP-коду на сайті:

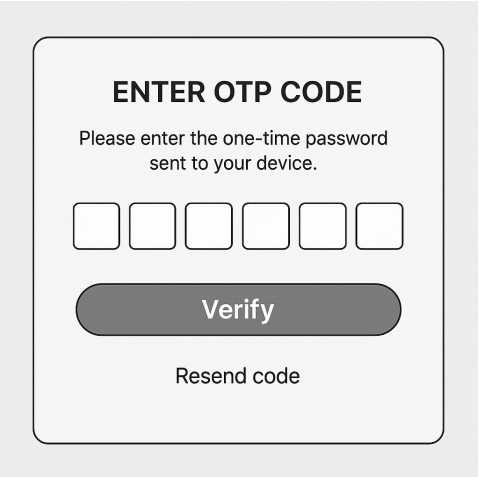


Рисунок 1.2 – Приклад інтерфейсу для введення OTP-коду на сайті

1.5 Біометричні методи автентифікації

Біометрична автентифікація — це метод перевірки особи користувача за допомогою фізіологічних або поведінкових характеристик, які є унікальними для кожної людини. На відміну від паролів або токенів, які можна втратити або передати іншій особі, біометричні дані значно складніше підробити або вкрасти.

До найпоширеніших типів біометричних даних належать:

- **Відбитки пальців** — один із найстаріших і найбільш використовуваних методів. Відбитки зчитуються за допомогою спеціальних сенсорів і порівнюються з раніше збереженим зразком.
- **Розпізнавання обличчя** — технологія аналізу унікальних рис обличчя за допомогою камери.
- **Сканування райдужної оболонки ока** — точний метод, який сканує візерунок райдужної оболонки за допомогою інфрачервоного випромінювання.
- **Голосова автентифікація** — визначає користувача за тембром, висотою та іншими параметрами голосу.
- **Геометрія долоні чи розпізнавання вен** — менш поширені, але теж використовуються у високо захищених системах.

Переваги біометричних методів:

- Високий рівень безпеки.
- Неможливість забути або втратити «пароль».
- Зручність для кінцевих користувачів — не потрібно вводити код вручну.

Недоліки:

- Висока вартість впровадження (особливо для апаратного забезпечення).
- Питання конфіденційності — витік біометричних даних може мати серйозні наслідки.
- Похибки розпізнавання (False Acceptance Rate / False Rejection Rate) (3).

У наш час біометричні методи активно застосовуються в банківській сфері, смартфонах, прикордонному контролі та системах безпеки. Одним із найпоширеніших рішень у побутових пристроях є сканер відбитків пальців, встановлений у смартфонах, банкоматах або системах контролю доступу (20).

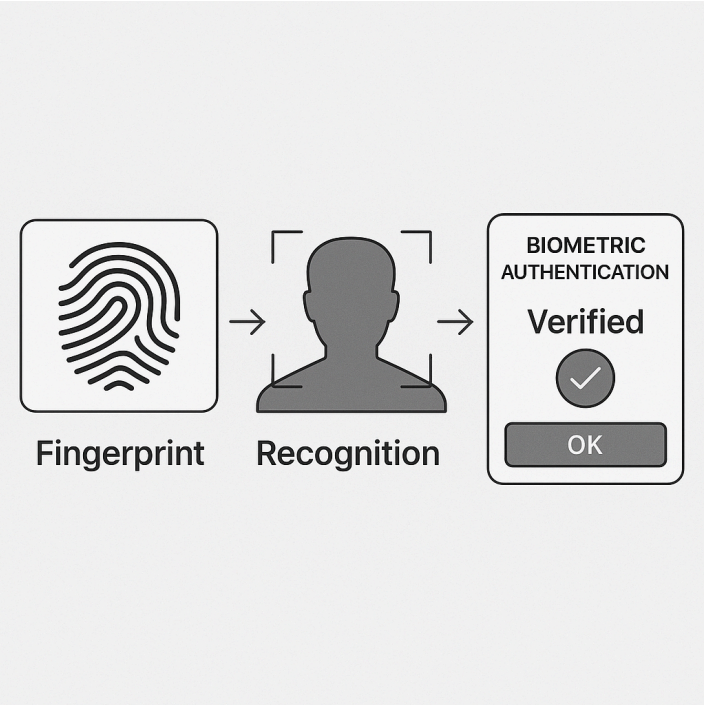


Рисунок 1.3 – Приклад біометричних методів автентифікації

1.6 Багатофакторна автентифікація (2FA/MFA)

Багатофакторна автентифікація (MFA) — це метод перевірки користувача, що передбачає використання більше одного фактору для підтвердження особи. Найбільш популярна конфігурація — це **двофакторна автентифікація (2FA)**, де застосовуються два незалежні фактори з трьох можливих:

1. **Щось, що ви знаєте** — пароль, PIN-код, відповідь на секретне питання.

2. **Щось, що ви маєте** — мобільний телефон, апаратний токен, код з додатка.
3. **Щось, чим ви є** — біометричні дані: відбитки пальців, розпізнавання обличчя, сітківка ока.

Приклад:

Найчастіше користувач спочатку вводить логін і пароль (**1 фактор**), після чого система просить ввести **OTP-код** з Google Authenticator або надісланий на мобільний телефон (**2 фактор**).

Основні переваги 2FA/MFA:

- Значно знижує ризик зламу, навіть якщо пароль було скомпрометовано.
- Ускладнює несанкціонований доступ до систем.
- Добре масштабується в організаціях і доступна навіть для домашніх користувачів.

Недоліки:

- Збільшення часу входу.
- Можливість втрати другого фактору (телефон, токен).
- Залежність від зовнішніх сервісів (наприклад, Google Authenticator) (4, 13).

Сучасні сервіси (Google, Facebook, банки, урядові портали) пропонують 2FA як обов’язкову або рекомендовану опцію. Деякі з них дозволяють користувачам самостійно обирати метод підтвердження: через додаток, SMS або апаратний ключ (наприклад, YubiKey).

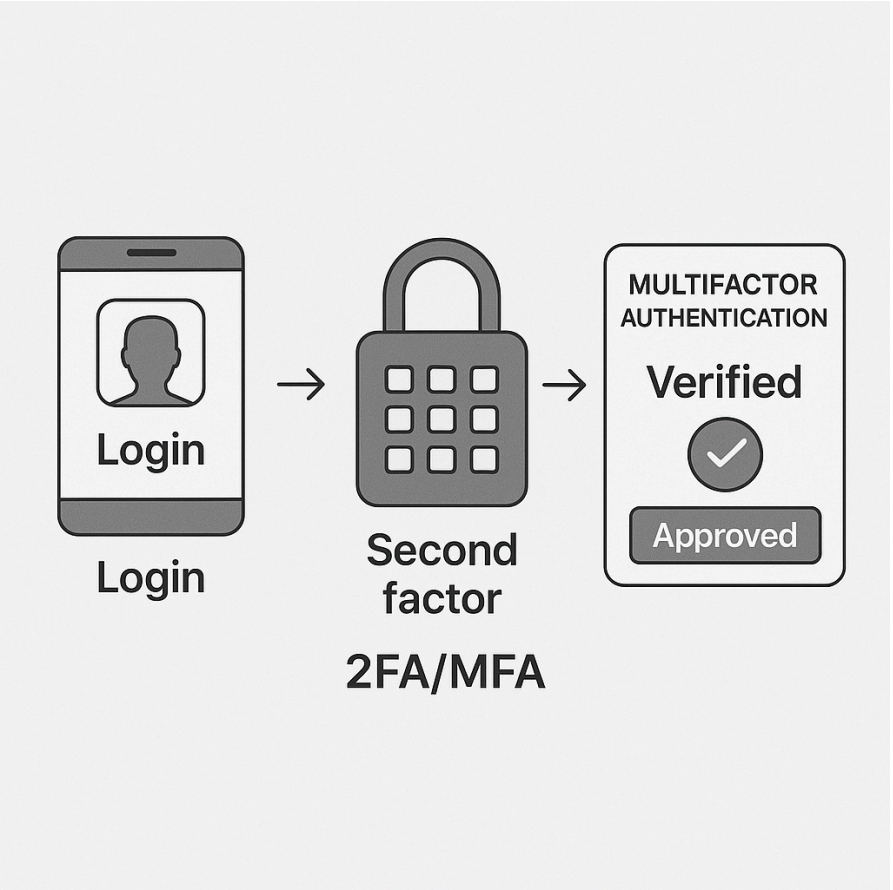


Рисунок 1.4 – Приклад багатофакторної автентифікації

1.7 Порівняльний аналіз методів автентифікації

Для забезпечення ефективного захисту інформаційних систем важливо правильно підібрати метод автентифікації. Нижче наведено порівняння основних методів за ключовими критеріями: безпека, зручність, вартість впровадження, масштабованість, ймовірність злому

Таблиця 1.1 - Порівняльний аналіз методів автентифікації

Метод автентифікації	**Безпека**	**Зручність**	**Вартість**	**Ризик зламу**	**Особливості**

Паролі	Низька	Висока	Низька	Високий	Широко застосовується, але легко зламати
Одноразові паролі OTP	Середня	Середня	Середня	Середній	Потребує генератора або додатку
Біометрія	Висока	Висока	Висока	Низький	Потрібне обладнання, конфіденційність
2FA/MFA	Висока	Середня	Середня	Низький	Надійна, але складніша для користувача
Токени (апаратні)	Висока	Середня	Висока	Низький	Використовується у банках, держсекторі

Висновок: жоден із методів не є ідеальним. Вибір залежить від контексту, ресурсу, ризиків і доступного бюджету. Наприклад, для звичайного користувача достатньо 2FA з паролем і OTP, а от в банківській чи державній системі доцільно застосовувати біометрію або токени.

Висновки до розділу 1

У першому розділі було розглянуто основні поняття, пов'язані з автентифікацією користувачів в інформаційних системах. Зокрема, було вивчено ключові типи автентифікації: парольну, одноразові коди (OTP), біометричну автентифікацю та багатофакторні підходи (2FA/MFA).

Проведений аналіз показав, що використання одного методу автентифікації (наприклад, пароля) не забезпечує належного рівня захисту, особливо в умовах зростаючої кількості кіберзагроз. Найбільш ефективними на сьогодні є комбіновані методи, що поєднують щонайменше два фактори — наприклад, пароль та OTP-код або пароль та біометричні дані.

Також було розглянуто сильні та слабкі сторони кожного підходу. Біометричні системи мають високий рівень безпеки, але вимагають дорогого обладнання. Паролі прості, але вразливі. Багатофакторна автентифікація пропонує баланс між безпекою і зручністю.

Таким чином, у подальших розділах буде розглянуто, як вибрати оптимальний метод автентифікації для конкретної системи, а також запропоновано власне практичне рішення на основі сучасних технологій автентифікації.

РОЗДІЛ 2. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

2.1 Актуальність теми

Інформаційна безпека сьогодні є критичним чинником не лише для державних установ, але й для бізнесу, освітніх закладів, фінансових структур, онлайн-магазинів та будь-яких організацій, що використовують цифрові платформи. Одним із ключових елементів захисту є система автентифікації користувачів — процес перевірки, чи є особа тією, за кого себе видає.

Згідно з останніми звітами міжнародних компаній у сфері кібербезпеки, понад 80% зламів відбуваються через компрометацію облікових даних, зокрема через вкрадені або слабкі паролі. У зв'язку з цим організації дедалі частіше впроваджують багатофакторну автентифікацію, біометричні методи, а також токени та OTP-коди.

Цифровізація державних послуг в Україні також спричинила активне впровадження систем авторизації в таких сервісах як «Дія», «ID.gov.ua», «BankID», які використовують комбінацію паролів, SMS-кодів та електронних підписів. Це підкреслює необхідність вивчення теми, її глибокого аналізу та створення рішень, що поєднують зручність і високий рівень безпеки.

2.2 Аналіз існуючих рішень

Сьогодні у світі активно використовуються наступні підходи до автентифікації:

- **Парольна автентифікація**: класичний варіант, що має суттєві недоліки (простота підбору, фішинг, повторне використання паролів).
- **Біометричні методи**: відбитки пальців, розпізнавання обличчя, райдужна оболонка ока, голос. Забезпечують високий рівень безпеки, але вимагають спеціального обладнання.

- **Токени (апаратні і програмні)**: надають додатковий фактор захисту, часто використовуються у банках.
- **Одноразові паролі (OTP)**: надсилаються на телефон або генеруються додатком (Google Authenticator, Microsoft Authenticator).
- **2FA/MFA**: комбінація кількох методів, наприклад, пароль + код з мобільного додатку або біометрія + PIN-код (16).

Існують також спеціалізовані програмні рішення, наприклад:

- **Auth0**, **Okta**, **Duo Security** — платформи, що надають готові інструменти для авторизації та автентифікації.
- **Firebase Authentication (Google)** — хмарна автентифікація для веб- та мобільних застосунків.

На українському ринку набули поширення рішення НБУ BankID, електронний цифровий підпис (ЕЦП) та Дія-Підпис (12**,** 17).

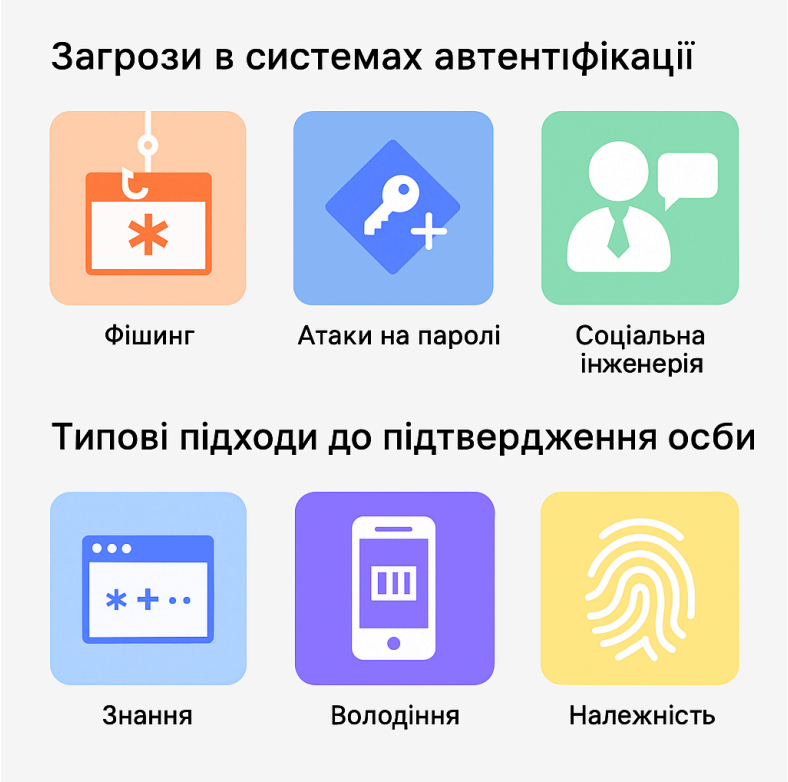


Рис. 2.1 – Основні загрози автентифікації та підходи до підтвердження особи

2.3 Проблеми реалізації автентифікаці

Попри наявність великої кількості методів і рішень, багато систем і досі використовують лише парольний захист. Основні проблеми, з якими стикаються компанії та розробники:

- Низький рівень обізнаності користувачів щодо безпеки.
- Відсутність обов’язкової багатофакторної автентифікації.
- Незахищені протоколи передачі даних (HTTP замість HTTPS).
- Збереження паролів у відкритому вигляді в БД.
- Складність інтеграції 2FA для внутрішніх або застарілих систем.
- Вартість впровадження біометричних рішень (6, 14).

2.4 Постановка задачі

Метою дипломної роботи є розробка програмного модуля автентифікації з підтримкою двофакторної перевірки особи (2FA) з використанням OTP-кодів і базової форми логіну.

Для досягнення мети необхідно виконати наступні завдання:

1. Проаналізувати основні сучасні методи автентифікації.
2. Вивчити приклади їх застосування в IT-системах.
3. Провести технічний аналіз існуючих сервісів (Google Authenticator, Microsoft Authenticator тощо).
4. Розробити веб-інтерфейс логіну з підтримкою OTP-коду.
5. Реалізувати серверну логіку генерації, перевірки одноразових паролів.
6. Здійснити тестування системи та підготувати звіт про результати.
7. Оцінити зручність використання, стабільність та потенційні загрози.

Висновки до розділу 2

У даному розділі було розглянуто актуальність захисту доступу до цифрових ресурсів, наведено приклади існуючих підходів до автентифікації, а також виявлено основні проблеми, з якими стикаються користувачі та розробники. Було сформовано завдання дипломного проекту, яке передбачає створення системи автентифікації з використанням OTP-кодів та аналіз її ефективності.

З отриманої інформації можна зробити висновок, що комбінований підхід до автентифікації (2FA/MFA) забезпечує найкращий баланс між безпекою та зручністю, а реалізація прототипу в умовах дипломної роботи дозволить перевірити цю гіпотезу на практиці.

Також було встановлено, що багато існуючих рішень, хоча й забезпечують базовий рівень безпеки, мають обмеження у масштабованості, складності інтеграції або потребують спеціального обладнання. Це відкриває можливість для створення простої, але ефективної системи, яка базуватиметься на відкритих технологіях, буде легкою у впровадженні та відповідатиме сучасним вимогам до безпеки. У наступних розділах буде реалізовано і протестовано таку систему.

РОЗДІЛ 3. РОЗРОБКА СИСТЕМИ АВТЕНТИФІКАЦІЇ

3.1 Вибір інструментів та технологій

Під час розробки системи автентифікаці я вирішив використати мову програмування **Python**, оскільки маю з нею попередній досвід, а також через її зручність і широку підтримку бібліотек для реалізації веб застосунків. Для створення серверної частини було обрано фреймворк **Flask**. Він легкий, має зрозумілу структуру та добре підходить для створення проєктів типу MVP (мінімально життєздатного продукту) (10, 11).

Інтерфейс користувача реалізовано на HTML та CSS із використанням бібліотеки **Bootstrap**, що дало змогу швидко створити привабливі й адаптивні веб сторінки. Для зберігання даних я використав **SQLite**, оскільки він легко інтегрується у Flask і підходить для невеликих навчальних проектів.

Генерацію одноразових кодів (OTP) здійснено за допомогою бібліотеки **PyOTP**, яка реалізує алгоритм TOTP, а для створення QR-кодів було використано бібліотеку **qrcode**. Система працює локально через **127.0.0.1**, що дозволило протестувати її в середовищі розробки.

Додатково, перспективним напрямом розвитку системи автентифікації є впровадження біометричних методів. Зокрема, автентифікація за відбитками пальців або за допомогою розпізнавання обличчя дозволяє забезпечити вищий рівень безпеки без необхідності введення пароля. У подальших версіях розробки система може бути розширена підтримкою WebAuthn або інших біометричних протоколів.

Таблиця 3.1 – Вибрані інструменти для розробки системи автентифікації

Компонент	**Інструмент / Технологія**	**Обґрунтування**
Мова програмування	Python	Зручна для веб-розробки, має бібліотеки для OTP, Flask, безпеку

Фреймворк	Flask	Легкий веб-фреймворк для швидкої розробки API та UI
Генерація OTP	PyOTP	Просте у використанні рішення для Time-Based OTP (TOTP)
Веб-інтерфейс	HTLM, Bootstrap	Створення адаптивної, зручної форми логіну
Перевірка OTP	Google Authenticator	Відомий мобільний застосунок, підтримує TOTP
Сховище	SQLite	Просте для невеликої системи, не вимагає додаткової установки
Запуск локального сайту	127.0.0.1:5000 або localhost	Для тестування без розгортання в продакшн

У таблиці 3.1 наведено компоненти, технології та обґрунтування вибору. Такий підхід дозволив швидко реалізувати проект, не втрачаючи безпеки.

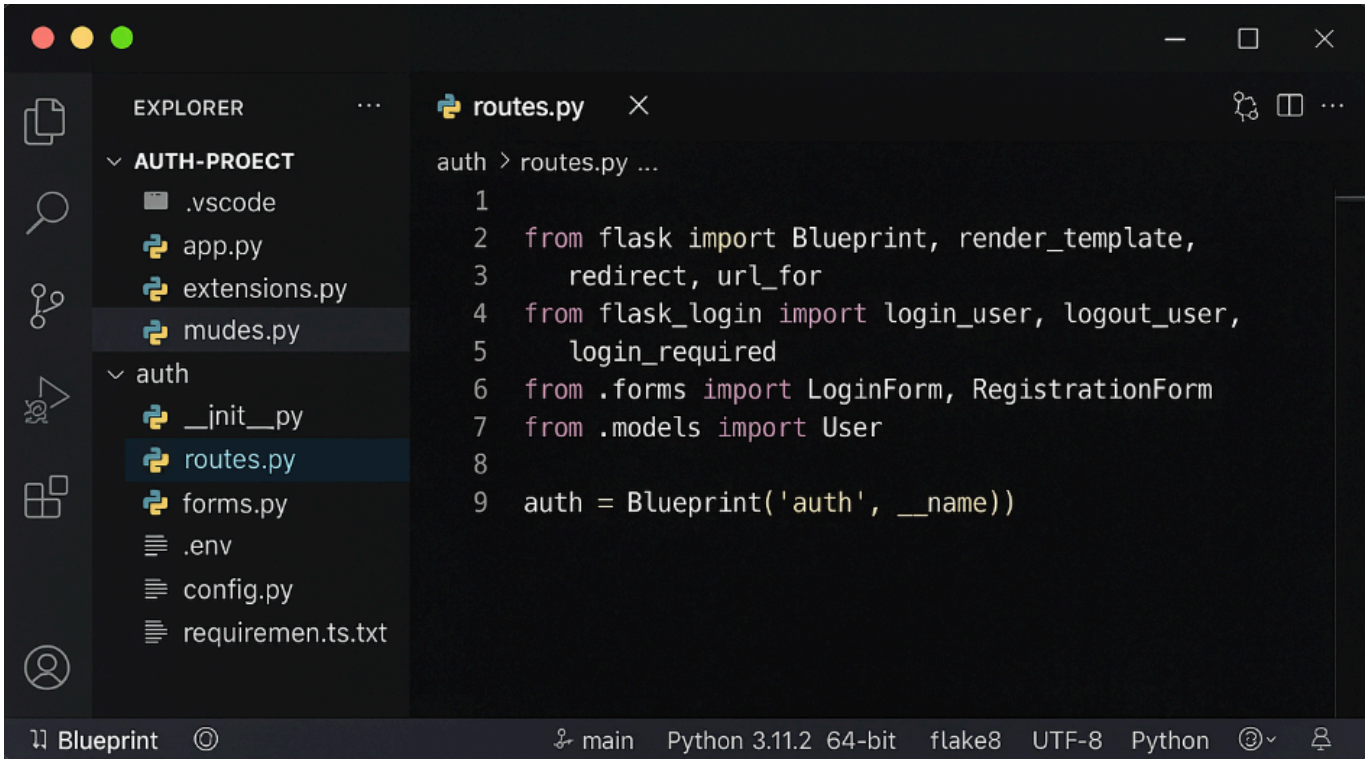


Рис. 3.1 – Структура проекту авторизації у Flask (Visual Studio Code)

3.2 Створення базової форми логіну

Першим кроком у реалізації була побудова сторінки входу користувача. Вона має просту, але функціональну форму, де потрібно ввести логін та пароль. Після натискання кнопки “Увійти” дані відправляються на сервер, де перевіряється правильність облікових даних.

Форма реалізована в HTML-шаблоні з використанням класів Bootstrap, що спростило процес візуального оформлення. Якщо дані неправильні — з’являється повідомлення про помилку. У випадку успішного входу користувач переходить на сторінку двофакторної перевірки (OTP).

Цей етап був важливим, оскільки саме тут закладається логіка перевірки користувача перед тим, як застосувати другий рівень безпеки.

Основні елементи форми:

- поле Login (Username)
- поле Password (пароль)
- кнопка “Увійти”
- (після успішного входу з паролем — відображається форма введення OTP)

Ключові особливості:

- форма є мінімалістичною для зручності;
- відображається помилка, якщо дані неправильні;
- використовується метод POST для безпеки.

Код HTML-сторінки **login.html**:

```
<!DOCTYPE html>

<html lang="uk">

<head>

   <meta charset="UTF-8">

   <title>Вхід в систему</title>

      <link  href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">
```

```
</head>
<body class="bg-light">
<div class="container mt-5">
  <h3 class="text-center mb-4">Авторизація користувача</h3>
  <form method="POST" action="/login" class="card p-4 shadow-sm">
    <div class="mb-3">
      <label for="username" class="form-label">Ім'я користувача</label>
        <input type="text" class="form-control" id="username" name="username" required>
    </div>
    <div class="mb-3">
      <label for="password" class="form-label">Пароль</label>
        <input type="password" class="form-control" id="password" name="password" required>
    </div>
    <button type="submit" class="btn btn-primary w-100">Увійти</button>
  </form>
</div>
</body>
</html>
```

Цей шаблон буде використовуватись у Flask-проєкті через render_template("login.html").

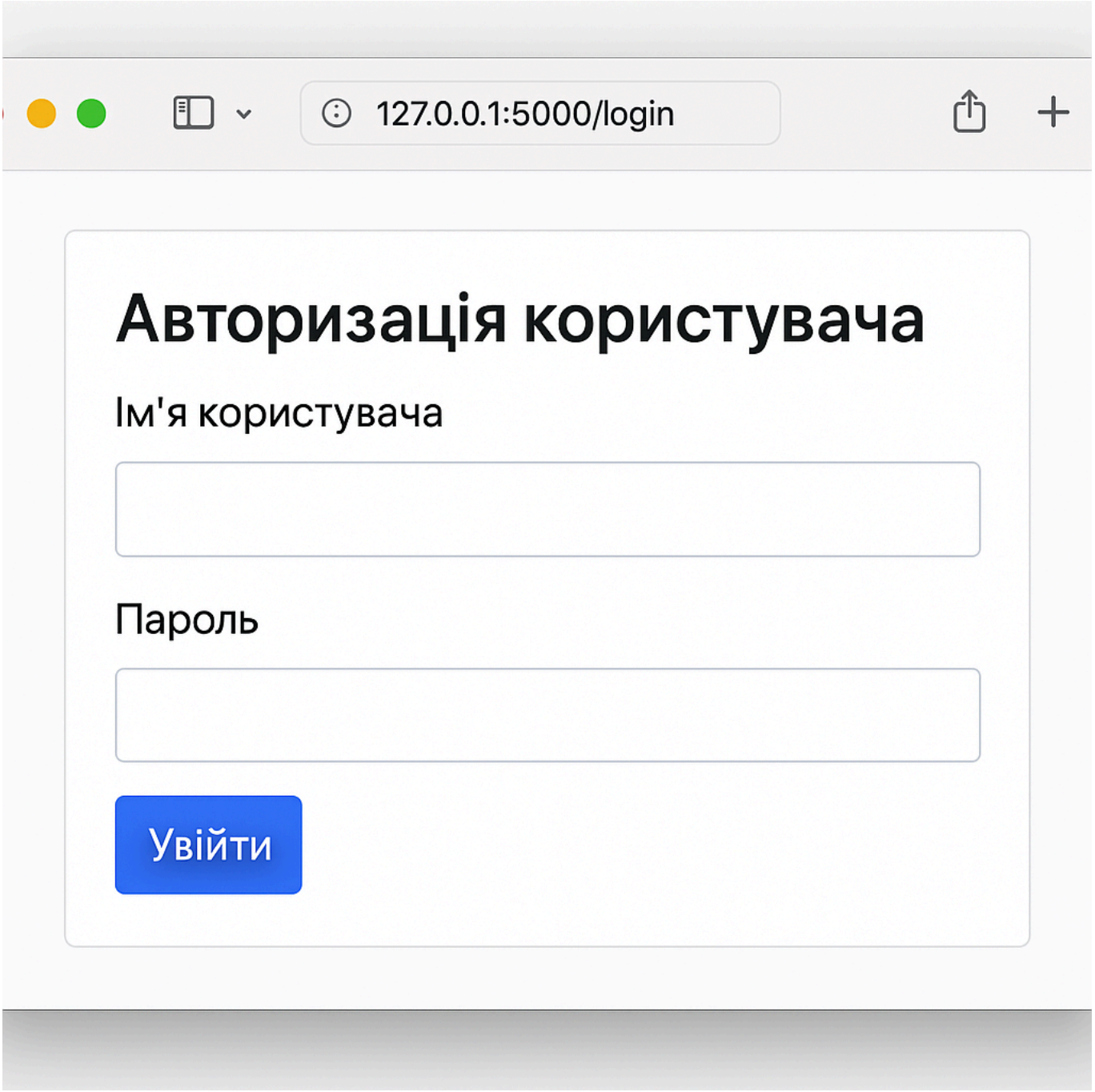


Рис. 3.2 – Веб-форма логіну користувача (HTML-інтерфейс)

3.3 Генерація та перевірка OTP-коду

Другим етапом стала реалізація логіки перевірки одноразових паролів. Після логіну користувач переходить на сторінку, де вводить OTP-код, згенерований у застосунку Google Authenticator.

У коді я реалізував збереження секретного ключа в сесії. На його основі створюється об’єкт TOTP, який дозволяє порівняти введений користувачем код з тим, який має бути в даний момент. Якщо код правильний — користувача вважають автентифікованим.

Цей підхід дозволяє значно підвищити безпеку, оскільки навіть у разі крадіжки логіна й пароля зловмисник не зможе пройти другу перевірку без доступу до мобільного застосунку.

Вибір бібліотеки: Для генерації OTP використовується Python-бібліотека PyOTP, яка підтримує алгоритм TOTP (Time-based One-Time Password), сумісний із мобільними застосунками типу Google Authenticator або Microsoft Authenticator.

Логіка реалізації:

1. Генерується секретний ключ користувача.
2. Цей ключ використовується для створення QR-коду, який сканується через застосунок Google Authenticator.
3. Кожні 30 секунд застосунок генерує новий одноразовий код.
4. Користувач вводить код у форму OTP.
5. Сервер перевіряє код через PyOTP (7, 8).

Код (фрагмент з **routes.py** або **auth.py**):

```
import pyotp
from flask import request, redirect, render_template, session

# Генерація секретного ключа (зазвичай один раз при реєстрації)
secret = pyotp.random_base32()
session['otp_secret'] = secret

# Створення OTP-об’єкта
otp = pyotp.TOTP(secret)
```

```
# Під час перевірки форми
@app.route('/otp', methods=['GET', 'POST'])
def otp_verification():
    if request.method == 'POST':
        code = request.form['otp_code']
        otp = pyotp.TOTP(session['otp_secret'])
        if otp.verify(code):
            return "Успішна автентифікація!"
        else:
            return "Невірний код."
    return render_template('otp.html')
```

HTML-форма otp.html:

```
<form method="POST" class="card p-4 shadow-sm">
    <label for="otp_code" class="form-label">Введіть код з застосунку</label>
    <input type="text" name="otp_code" class="form-control mb-3" required>
    <button type="submit" class="btn btn-success w-100">Підтвердити</button>
</form>
```

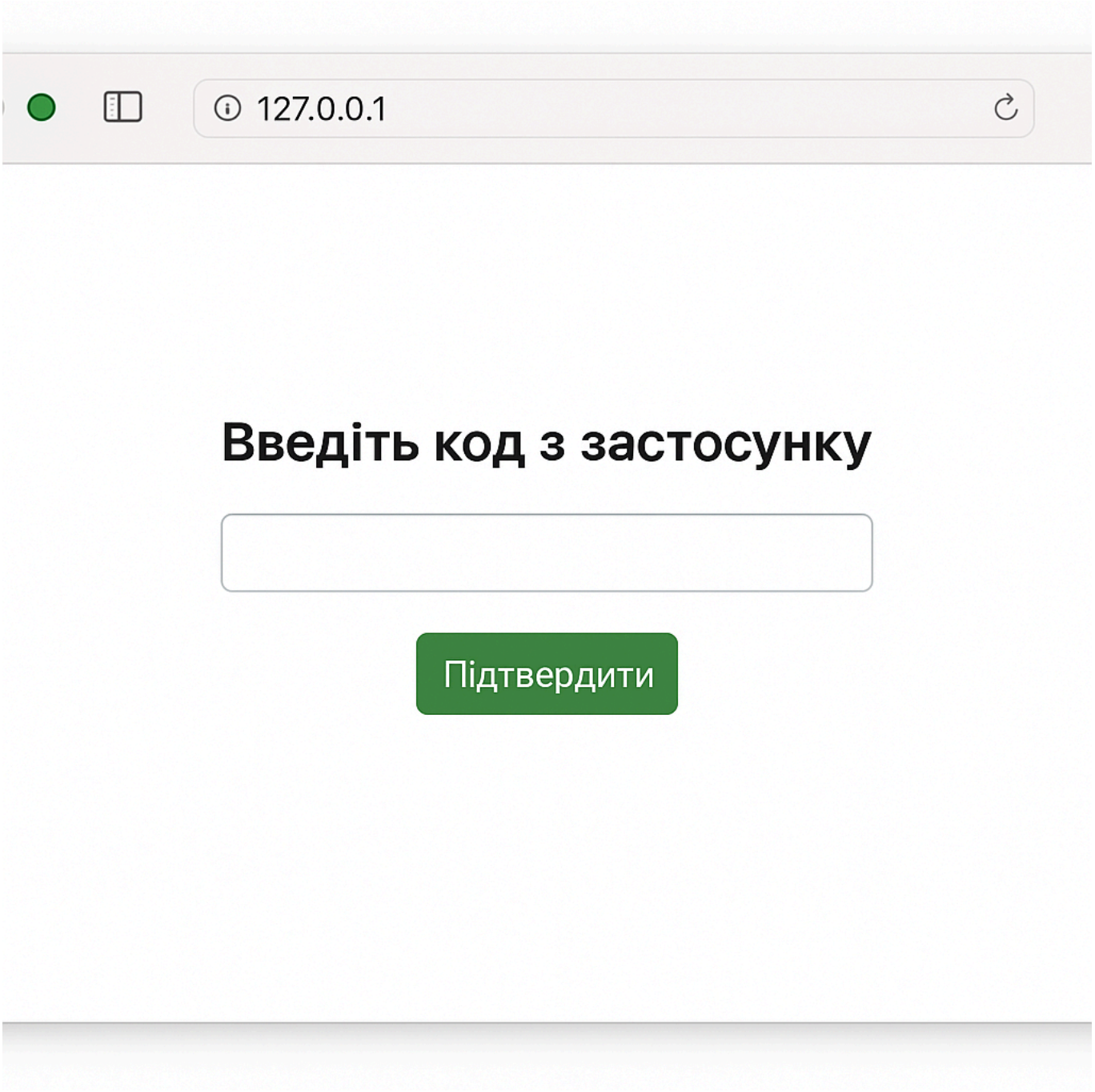


Рис. 3.3 – Форма введення OTP-коду після логіну користувача

3.4 Інтеграція з Google Authenticator

Щоб реалізувати двофакторну автентифікацію, я вирішив використати мобільний застосунок **Google Authenticator**, який генерує коди за алгоритмом TOTP. Для цього

на сервері генерується унікальний секрет для кожного користувача, на основі якого створюється спеціальний URI, який відповідає формату otpauth (9).

Цей URI далі перетворюється у QR-код, який відображається у браузері. Користувач сканує його камерою телефону в застосунку Google Authenticator, після чого той починає генерувати OTP-коди кожні 30 секунд.

Генерація QR-коду здійснюється через бібліотеку **qrcode**, а сам код відображається в шаблоні як base64-зображення. Цей підхід дозволив легко реалізувати механізм без додаткового зовнішнього API.

Фрагмент коду генерації **QR-коду**:

```
import pyotp
import qrcode
import io
import base64
from flask import render_template_string, session


@app.route('/generate-qr')
def generate_qr():
    otp_secret = pyotp.random_base32()
    session['otp_secret'] = otp_secret
                                                    otp_uri =
pyotp.totp.TOTP(otp_secret).provisioning_uri(name="user@example.com",
issuer_name="MySecureApp")
    img = qrcode.make(otp_uri)
    buffer = io.BytesIO()
    img.save(buffer, format='PNG')
    qr_img = base64.b64encode(buffer.getvalue()).decode()
    return render_template_string('''
        <h3>Скануйте цей QR-код за допомогою Google Authenticator:</h3>
```

```
        <img src="{{qr_img}}">
    ''', qr_img=qr_img)
```

Цей етап є ключовим у підключенні до мобільного додатку, що забезпечує додатковий рівень захисту у випадку компрометації основного пароля.

Рис. 3.4 – Генерація QR-коду для підключення Google Authenticator

3.5 Тестування системи автентифікації

Після завершення реалізації основних компонентів системи автентифікації було проведено її тестування в локальному середовищі розробки. Основна мета цього етапу — перевірити, як працюють різні сценарії входу, та переконатися, що система реагує правильно у випадках як успішного, так і помилкового вводу даних.

Перевірка сценаріїв:

1. Успішний вхід:
    - Введено правильний логін та пароль.
    - Користувач перенаправлений на сторінку введення OTP.
    - Введено дійсний одноразовий код з застосунку Google Authenticator.
    - Вхід виконано успішно.

2. Неправильний пароль:
    - Система виводить повідомлення про помилку.
    - Вхід заборонено.

3. Правильний пароль, неправильний OTP-код:
    - Система блокує вхід і просить повторити введення коду.

4. Використання простроченого одноразового коду (через 30+ секунд):
    - Код недійсний.
    - Користувач повинен ввести новий.

5. Використання токена з іншого облікового запису:
    - Автентифікація не відбувається.
    - Захист працює коректно.

Результати:

За підсумками тестування можна зробити висновок, що система правильно обробляє всі ключові ситуації.

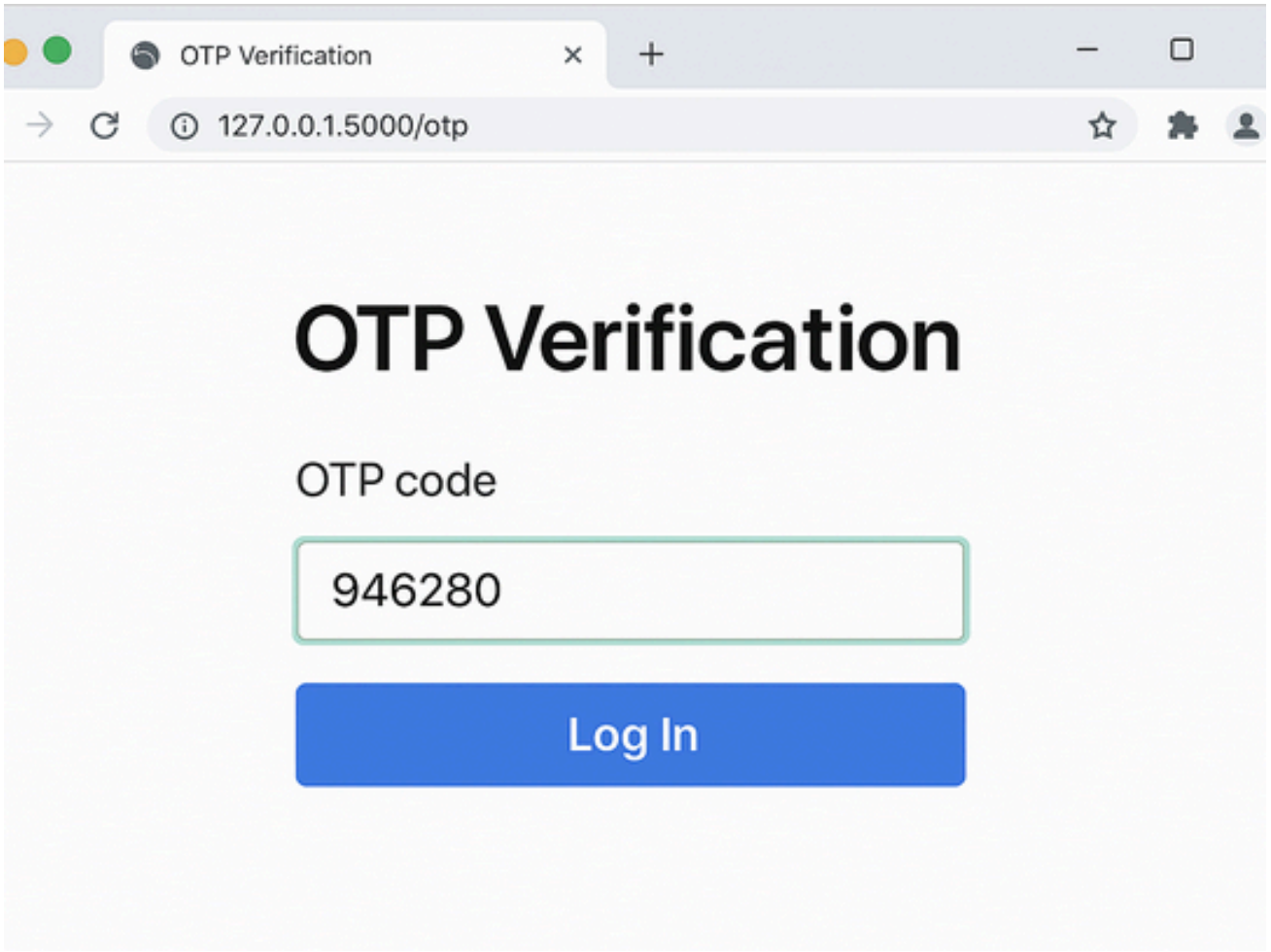


Рис. 3.5 – Тестування системи автентифікації

Перевірка одноразового коду працює відповідно до очікувань, і жоден сценарій не призвів до несанкціонованого доступу. Зокрема, важливо, що одноразовий код не приймається повторно або після закінчення його терміну дії.

Також було перевірено, що секретний ключ не зберігається у відкритому вигляді, а передається лише через сесію, що відповідає базовим вимогам безпеки.

3.6 Скріншоти роботи системи

Для кращого розуміння функціоналу створеної системи автентифікації в цьому розділі наведено серію скріншотів, які демонструють ключові етапи взаємодії користувача з веб інтерфейсом. Це дає змогу наочно оцінити зручність, послідовність та ефективність розробленого рішення.

1. Сторінка авторизації

На першому етапі користувач бачить форму входу, де потрібно ввести логін та пароль. Сторінка має адаптивний дизайн, що забезпечується за рахунок використання Bootstrap. Введення некоректних даних призводить до появи відповідного повідомлення про помилку.

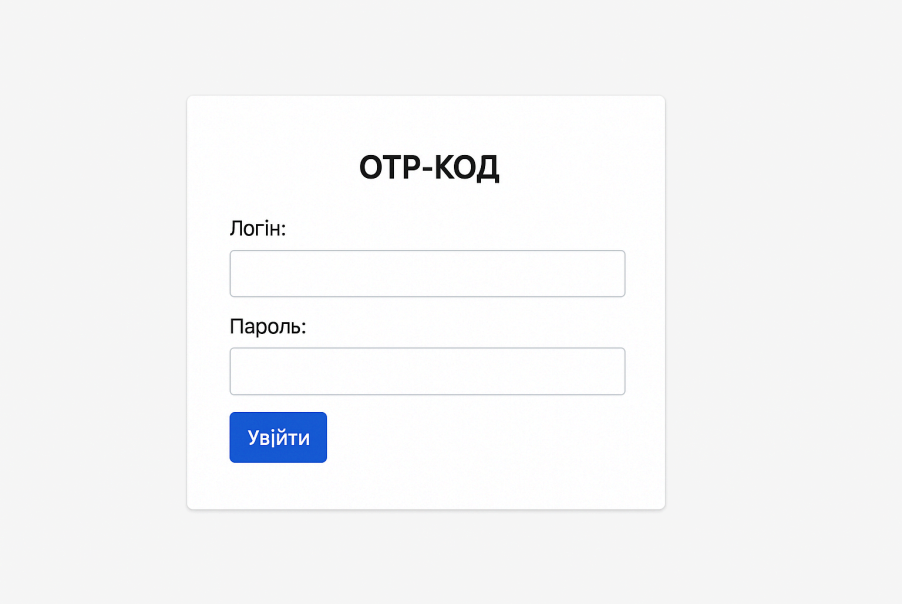


Рис. 3.6.1 – Форма входу до системи

2. Генерація QR-коду

Після першого входу система пропонує сканувати QR-код для підключення Google Authenticator. Завдяки цьому користувачі можуть використовувати двофакторну автентифікацію, що значно підвищує рівень безпеки.

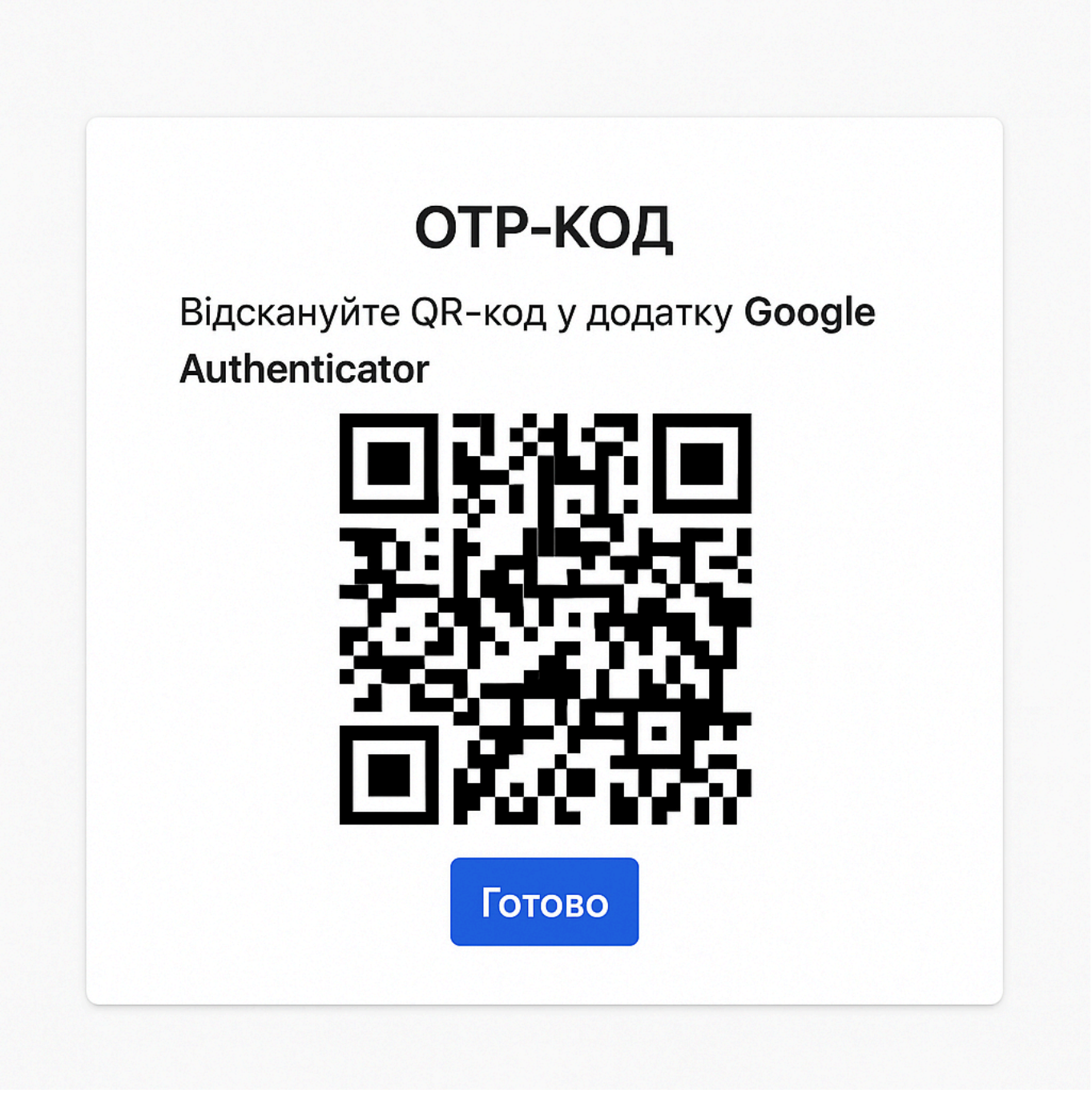


Рис. 3.6.2 – Генерація QR-коду для Google Authenticator

3. Сторінка введення OTP-коду

Наступний етап — введення одноразового коду з мобільного застосунку. При правильному введенні користувач отримує доступ до системи. У разі помилки система повідомляє про недійсний код.

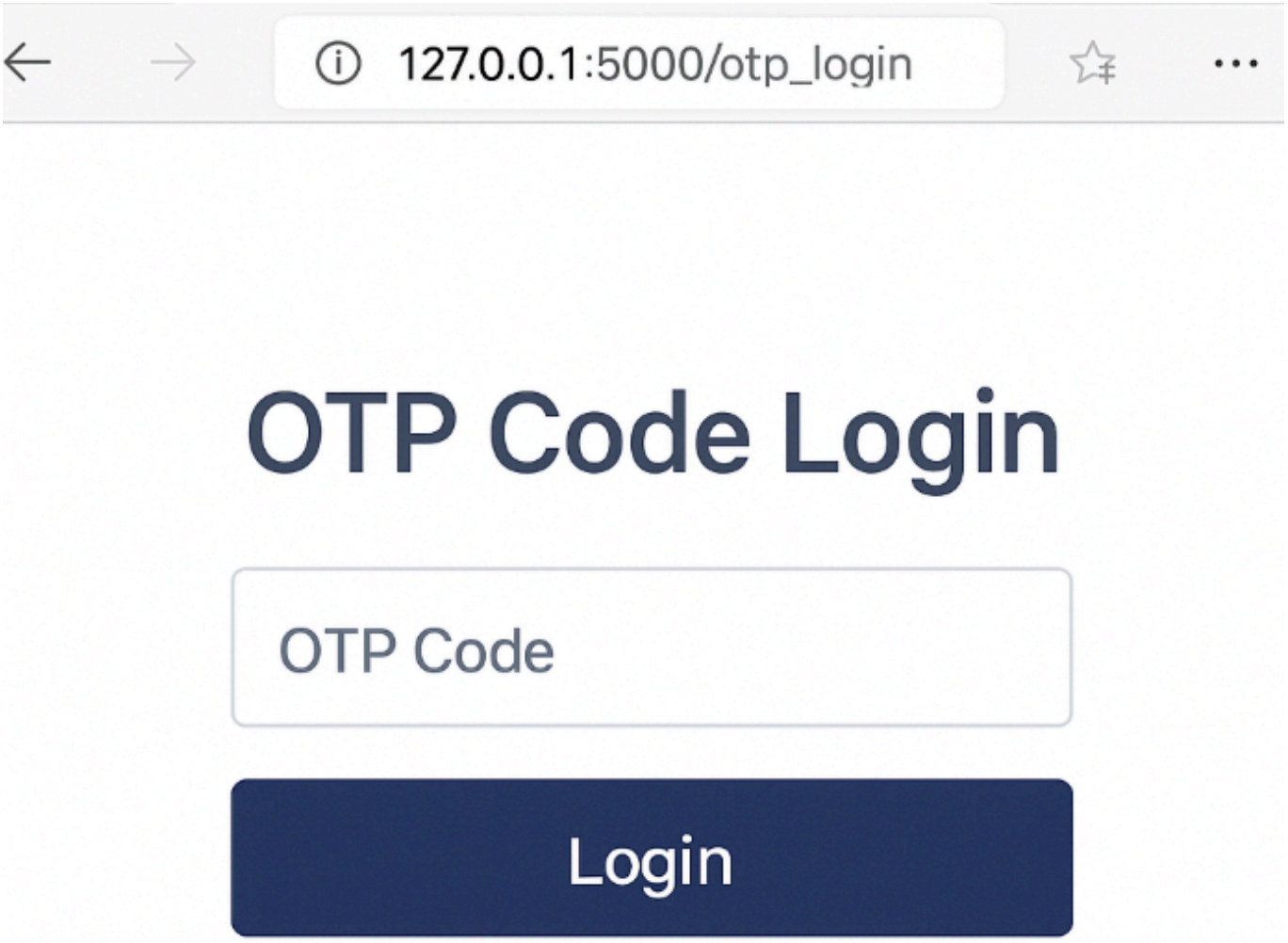


Рис. 3.6.3 – Введення одноразового пароля (OTP)

4. Успішна автентифікація

Після проходження обох етапів перевірки користувач бачить повідомлення про успішну авторизацію та потрапляє до основного інтерфейсу або захищеної частини веб ресурсу.

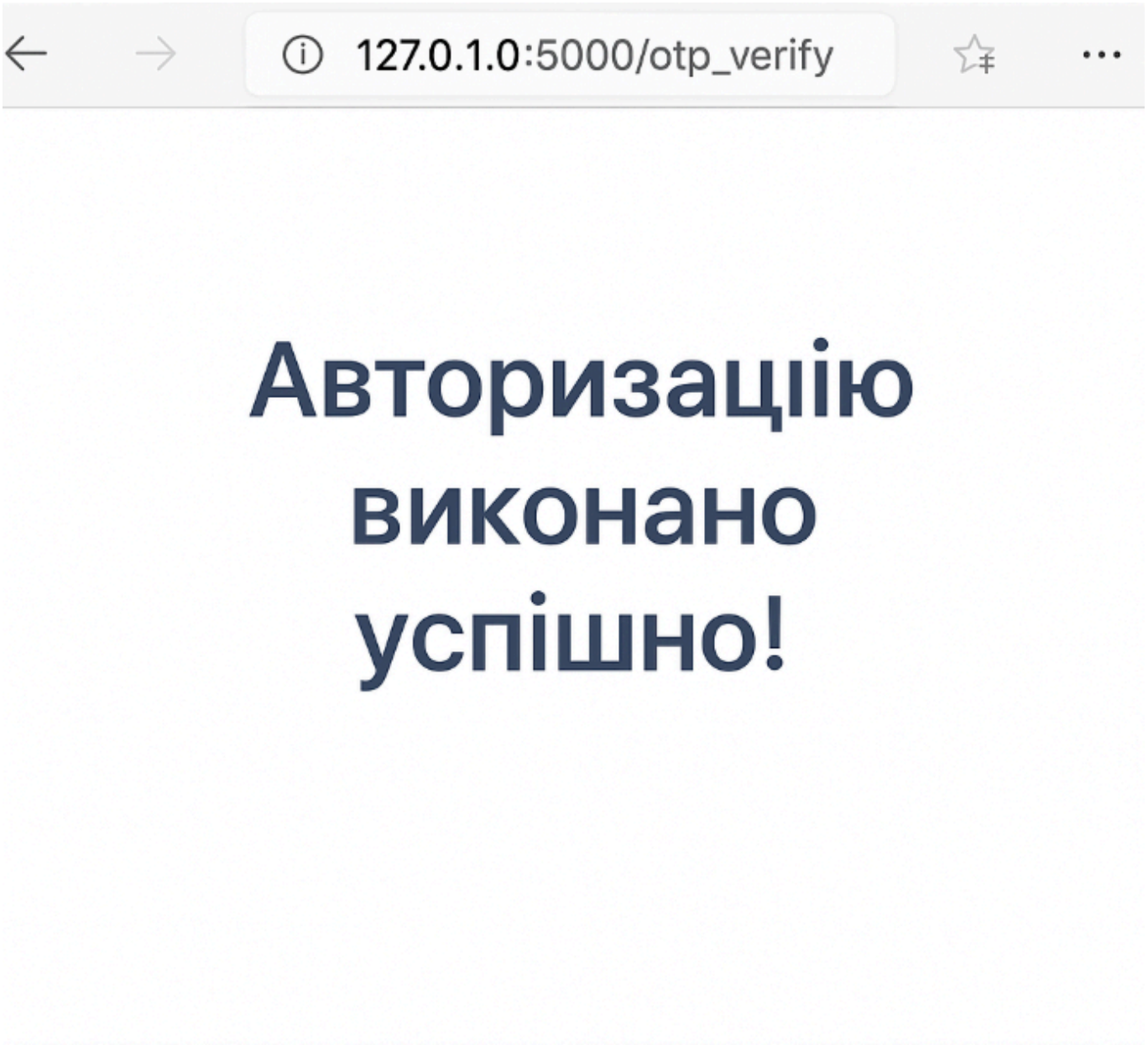


Рис. 3.6.4 – Повідомлення про успішну авторизацію

Програма була успішно протестована без виявлення помилок у її роботі.

Висновки до розділу 3

У рамках даного розділу було проаналізовано та вибрано технології для реалізації програмного забезпечення біометричних систем. З варіантів мов програмування C++, Java та C#, для розробки було обрано C# через його широкі можливості в створенні безпечних і надійних додатків. Описані алгоритми, що лягають в основу системи, включаючи розпізнавання відбитків пальців, реєстрацію, автентифікацію, обробку та розпізнавання зображень, демонструють глибину технічної реалізації проекту.

Також було визначено архітектуру системи, яка включає трирівневий підхід, забезпечуючи чітке розділення логіки, представлення і доступу до даних, що дозволяє підвищити масштабованість та зручність обслуговування системи. Значну увагу приділено інтеграції та налаштуванню компонентів системи, включаючи вибір Arduino Leonardo як основного контролера через його вартісну ефективність та легкість використання. Вибір Wi-Fi як протоколу передачі даних підкреслює акцент на високій швидкості та надійності комунікацій в рамках системи.

Виконане тестування системи підтвердило її ефективність і надійність, що відображено у проведеному порівняльному аналізі з іншими аналогами на ринку. Розроблене програмне забезпечення виділяється масштабованістю, можливістю гнучкого налаштування доступу, відсутністю ліцензійних платежів, а також наявністю української локалізації, роблячи його привабливим рішенням для широкого кола користувачів. Ці особливості і переваги підкреслюють успішність виконання поставлених завдань і забезпечення високих стандартів якості у розробці біометричних систем.

РОЗДІЛ 4 ОХОРОНА ПРАЦІ

4.1 Особливості охорони праці в сфері кібербезпеки

Охорона праці у сфері кібербезпеки має свою специфіку, пов’язану з особливостями професійної діяльності працівників, які працюють із конфіденційною інформацією, програмним забезпеченням, системами моніторингу та реагування на інциденти. Робота фахівців із кібербезпеки зазвичай відбувається в умовах підвищеної відповідальності, психологічного навантаження, тривалої роботи за комп’ютером і часто – в режимі багатозадачності.

Основними умовами для забезпечення безпеки праці в галузі кібербезпеки є:

- **Організація ергономічного робочого місця**, що включає належне розміщення моніторів, зручне крісло, регулювання висоти столу, оптимальне освітлення та зменшення навантаження на зір;
- **Дотримання режиму праці та відпочинку**: регулярні перерви після кожної години роботи за ПК, вправи для очей та рухова активність для зменшення статичного навантаження;
- **Психоемоційна гігієна** – у зв’язку з високим рівнем стресу під час аналізу загроз та реагування на інциденти, рекомендовано організацію гнучкого графіка, можливість віддаленої роботи, використання систем підтримки ментального здоров’я;
- **Забезпечення електробезпеки та безпечного підключення мережевого обладнання**, у тому числі ІБ-інфраструктури (сервери, маршрутизатори, брандмауери);
- **Безпечне розміщення обладнання** в серверних приміщеннях: наявність охолодження, захисту від перегріву, короткого замикання, протипожежних засобів;
- **Розробка інструкцій на випадок надзвичайних ситуацій**, включаючи сценарії втрати доступу до систем, кібератак або відключення

електропостачання.

Крім того, особливу увагу слід приділяти **конфіденційності робочого середовища** — забезпеченню візуальної приватності, обмеженню доступу сторонніх осіб до робочих комп’ютерів та безпечному зберіганню цифрових носіїв.

4.2 Умови безпечної роботи з ПЕОМ

При виконанні дипломного проєкту використовувалась персональна електронно-обчислювальна машина (ПЕОМ) типу ноутбук, що входить до складу автоматизованого робочого місця користувача інформаційних систем. Безпека праці під час роботи з комп’ютером регламентується чинним законодавством України, зокрема Законом України «Про охорону праці» та відповідними санітарними нормами ДСанПіН 3.3.2.007-98 (19).

Робоче місце було організоване з урахуванням ергономічних вимог:

- **Робочий стіл і крісло** мають регулювання висоти, що дозволяє правильно позиціонувати тіло під час роботи;
- **Освітлення** забезпечене за рахунок комбінованої системи: природне освітлення з вікна та штучне — LED-лампа з температурою світла 4000К, що знижує навантаження на зір; $E = F \cdot N \cdot \eta / S \cdot k$

де:

- E – нормативна освітленість (лк);
- F – світловий потік однієї лампи (лм);
- N – кількість ламп;
- η – коефіцієнт використання світла (приймаємо 0.6);
- S – площа приміщення (м²);

- k – коефіцієнт запасу (приймаємо 1.5).

Припустимо:

- використовуються 2 LED-лампи по 1000 лм,

- площа кімнати – 12 м².

$E = 1200/18 \approx 66.7$ лк на1м2

Загальна освітленість ≈ 400–450 лк, що відповідає ДБН В.2.5-28:2006 — для роботи за комп’ютером рекомендовано не менше 300 лк.

- **Монітор** розміщено на відстані приблизно 60 см від очей користувача з нахилом екрана 10–15 градусів;
- **Робота ведеться сидячи**, спина підтримується анатомічним кріслом, що знижує ризик виникнення захворювань хребта;
- **Зовнішній шум та вібрація** відсутні — робота виконувалась у домашньому середовищі в умовах тиші.

Відповідно до гігієнічних вимог, тривалість безперервної роботи за комп’ютером не повинна перевищувати 2 годин, після чого рекомендовано зробити 10–15-хвилинну перерву. Під час виконання дипломної роботи було дотримано режиму праці та відпочинку.

Також було враховано вимоги щодо електробезпеки — ноутбук підключався через мережевий фільтр, а всі кабелі були надійно закріплені для уникнення травмування.

Загалом, умови виконання дипломної роботи були безпечними, комфортними й відповідали санітарно-гігієнічним нормам, що забезпечило ефективну роботу та збереження здоров’я.

4.3 Вимоги до електробезпеки

Електробезпека — це сукупність організаційних і технічних заходів, які забезпечують захист людини від шкідливої дії електричного струму, електричної дуги, електромагнітного випромінювання, а також запобігання виникненню пожеж у разі короткого замикання або перевантаження мережі.

При роботі з ПЕОМ, які живляться від електромережі змінного струму 220 В, існує потенційна небезпека ураження електричним струмом. Тому під час виконання дипломної роботи були дотримані всі ключові вимоги до електробезпеки:

- Робоче місце було обладнане мережевим фільтром з функцією захисту від перенапруги, що мінімізує ризик ураження в разі стрибків напруги;
- Всі прилади (ноутбук, зарядний пристрій, лампа) мають заводську ізоляцію кабелів живлення та сертифіковані відповідно до стандартів безпеки;
- Перед використанням комп’ютера проводився візуальний огляд кабелів і розеток для виявлення можливих пошкоджень;
- В приміщенні, де велась робота, відсутні джерела підвищеної вологості — це також важливо для дотримання норм електробезпеки;
- Заборонялося одночасне використання вологих рук або мокрої поверхні при контакті з елементами живлення.

Вся техніка під час роботи розміщувалась на діелектричній (непровідній) основі, а дроти були акуратно організовані, що виключає випадкове їх пошкодження або утворення пожежонебезпечних ситуацій.

Таким чином, під час реалізації дипломного проєкту були дотримані всі основні вимоги щодо електробезпеки. Це дозволило уникнути аварійних ситуацій та забезпечити безпечне функціонування обладнання.

4.4 Пожежна безпека

Пожежна безпека — це сукупність заходів, спрямованих на запобігання виникненню пожеж, забезпечення безпечної евакуації людей, збереження матеріальних цінностей і функціональності обладнання. З огляду на те, що під час виконання дипломного проєкту використовувалась комп’ютерна техніка та побутова електрика, важливо було врахувати основні протипожежні вимоги.

Під час роботи над дипломом були дотримані наступні правила пожежної безпеки:

- Усі електроприлади були у справному стані та мали сертифікати відповідності. Живлення здійснювалось через мережевий фільтр із вбудованим запобіжником.
- Приміщення не перевантажувалось зайвими пристроями. Розетки та подовжувачі не використовувались понад допустимі норми навантаження.
- На робочому місці не було горючих або легкозаймистих матеріалів поруч із комп’ютерною технікою.
- В умовах домашнього середовища було визначено шляхи евакуації, а також розміщено вогнегасник поблизу робочої зони.
- Після завершення роботи комп’ютер та інші пристрої відключалися від мережі для запобігання самовільного перегрівання або короткого замикання.
- Робота проводилась лише при нормальному температурному режимі, із хорошою вентиляцією — це дозволяє уникати перегріву електроніки, що є одним із факторів ризику виникнення пожежі.

Ці заходи дозволили організувати роботу над дипломною роботою з урахуванням вимог пожежної безпеки. Завдяки дотриманню відповідних норм, загрозу виникнення надзвичайних ситуацій було зведено до мінімуму.

4.5 Вимоги до мікроклімату, освітлення та шуму

Комфортні та безпечні умови праці мають велике значення під час роботи з персональним комп’ютером, особливо в рамках тривалого інтелектуального навантаження. До таких умов належать мікроклімат, рівень освітленості та допустимі норми шуму.

Мікроклімат

Мікрокліматичні умови приміщення, де велась робота над дипломним проєктом, відповідали санітарним вимогам, встановленим **ДСанПіН 3.3.2.007-98**. Температура повітря підтримувалась у межах **20–24 °C**, що є оптимальним для роботи з ПЕОМ. Вологість повітря знаходилась у межах **40–60%**, що забезпечувалось регулярним провітрюванням. Приміщення мало стабільну циркуляцію повітря без протягів.

Освітлення

Освітлення робочого місця виконувалося за змішаною схемою — природне денне світло з вікна, а також штучне освітлення за допомогою **настільної LED-лампи**. Рівень освітленості становив **не менше 300 лк**, що відповідає державним нормам для роботи з комп’ютером. Лампа мала нейтральну температуру світла (**4000–5000 К**) та не створювала мерехтіння, що знижує навантаження на зір.

Монітор був розташований під правильним кутом з нахилом, що знижує втому очей і підвищує загальну зручність роботи. Згідно з вимогами, **джерела світла не створювали відблисків** на екрані, а освітлення було рівномірним.

Шум

Під час виконання роботи **зовнішні джерела шуму були відсутні**. Єдиним джерелом звуку була система охолодження ноутбука, рівень шуму якої не перевищував **30–35 дБ**, що вважається безпечним і не заважає концентрації.

Розрахунок рівня шуму

Рівень шуму в кімнаті під час роботи ПК визначається сукупністю джерел звуку. Основним джерелом був вентилятор ноутбука (≤35 дБ), що не перевищує гранично допустимі рівні шуму згідно з ДСанПіН 3.3.2.007-98:

- допустимий рівень шуму на робочому місці – до 50 дБ,
- фактичний рівень – ≈30–35 дБ.

Це відповідає санітарним нормам для офісних приміщень і не створює дискомфорту при інтелектуальній роботі.

Таким чином, мікрокліматичні параметри, освітлення та рівень шуму відповідали нормам безпеки, створюючи комфортне середовище для ефективного виконання інтелектуальної праці.

Оцінка тепловиділення обладнання

При тривалій роботі комп’ютерного обладнання виникає теплове навантаження на приміщення. Орієнтовне тепловиділення для одного ноутбука:

- Потужність ноутбука: 65 Вт,
- Час роботи: 6 годин,
- Тепловиділення:

$Q = P \cdot t = 65 \cdot 6 = 390$ Вт·год

Це не перевищує допустиме значення тепловиділення для житлових/офісних приміщень. Завдяки вентиляції температура повітря залишалась в межах норми (20–24 °C).

4.6 Заходи безпеки у надзвичайних ситуаціях

Безпека у надзвичайних ситуаціях є важливим елементом забезпечення охорони праці працівників організації. Навіть у звичайних офісних приміщеннях, де основною діяльністю є робота за комп'ютером, можуть виникати потенційно небезпечні ситуації, пов'язані з електроприладами, пожежами, витоками газу чи води, або аварійними відключеннями електроенергії.

Основними джерелами ризику в приміщеннях, де розміщено IT-обладнання, є:

- коротке замикання електропристроїв (ноутбук, сервер, мережеве обладнання);
- перегрів елементів через погану вентиляцію;
- використання несертифікованих подовжувачів і блоків живлення;
- необережне поводження з водою поруч з технікою;
- людський фактор — залишення зарядних пристроїв у розетці, куріння поруч з обладнанням тощо.

У таких умовах на підприємстві повинні бути визначені та доступні такі заходи безпеки:

- Наявність плану евакуації, розміщеного на видному місці (наприклад, при вході в кабінет або коридор);
- Вогнегасники, розташовані в зоні досяжності працівників (відстань до вогнегасника — не більше 20 м), зазвичай використовують порошкові або вуглекислотні;
- Система пожежної сигналізації, яка забезпечує звукове або світлове сповіщення у разі задимлення або перегріву обладнання;

- Обов’язкові інструктажі з техніки безпеки та проведення тренувальних евакуацій не рідше одного разу на півроку;
- Наявність аптечки першої допомоги на випадок травм або опіків.

У разі виникнення надзвичайної ситуації персонал зобов’язаний:

1. Негайно повідомити відповідальну особу або викликати екстрені служби (тел. 101 або 112).
2. Вимкнути електроживлення обладнання, якщо це не становить загрози для життя.
3. Покинути приміщення згідно з планом евакуації, не користуючись ліфтами.
4. Уникати паніки, попередити інших співробітників, при необхідності надати першу допомогу.

Усі працівники повинні бути ознайомлені з правилами поведінки у разі пожежі, задимлення чи загрози вибуху, а також мати доступ до інструкції з надзвичайних ситуацій, яка зберігається у паперовому та електронному вигляді.

Висновки до розділу 4

У розділі було розглянуто основні питання, пов’язані з охороною праці під час роботи з персональним комп’ютером, що використовувався для реалізації дипломного проєкту. Особливу увагу приділено безпечній організації робочого місця, електробезпеці, пожежній безпеці, а також мікрокліматy, освітленню та рівню шуму.

Було встановлено, що умови, в яких виконувалась робота, повністю відповідали чинним санітарно-гігієнічним вимогам. Робоче місце організоване ергономічно,

освітлення і вентиляція — на належному рівні. Дотримано всі правила електро- та пожежної безпеки, що забезпечило мінімізацію ризиків під час виконання проєкту.

РОЗДІЛ 5 ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ ДИПЛОМНОГО ПРОЄКТУ

5.1 Мета та завдання економічного обгрунтування

Метою даного розділу є визначення орієнтовної вартості створення, впровадження та підтримки програмного забезпечення системи автентифікації користувачів, а також оцінка доцільності проєкту з економічної точки зору.

Основні завдання економічного обґрунтування:

- оцінити витрати на розробку програмного забезпечення;
- визначити заробітну плату учасників розробки;
- розрахувати амортизаційні витрати обладнання;
- оцінити можливий економічний ефект або користь від впровадження системи.

5.2 Витрати на розробку програмного забезпечення

У нашому випадку розробка програмного забезпечення виконувалась студентом, тому розрахунок проводиться умовно, виходячи з припущення, що розробкою займається інженер-програміст.

Таблиця 5.1 - Підрахунок витрат на розробку програмного забезпечення

Найменування витрат	**Одиниця виміру**	**Кількість**	**Ціна, грн**	**Сума, грн**
Оплата праці програміста (30 днів)	грн/день	30	1000	30 000
Податки та нарахування	% від зарплати	—	—	6 600

(22%)				
Амортизація техніки (ноутбук)	грн/міс	1	1000	1 000
Витрати на електроенергію	кВт·год	90	3	270
ПЗ з відкритим кодом (Flask, Python)	—	—	—	0
Інтернет	грн/міс	1	250	250
Інші витрати	—	—	—	400
Разом:				**38 520**

5.3 Амортизаційні витрати

Припустимо, що використовується ноутбук вартістю 24 000 грн, строк експлуатації — 3 роки. Амортизація розраховується за формулою:

$$A = C/T \cdot 12$$

де:

- A – місячна сума амортизації;
- C – початкова вартість (24 000 грн);
- T – строк служби в роках (3).

$$A = 24000/36 \approx 667 \text{ грн/міс}$$

Оскільки розробка тривала приблизно місяць, загальна амортизація = **667 грн**.

5.4 Економічна ефективність від впровадження системи

Припустимо, що компанія щомісяця витрачає в середньому **5 000 грн** на відновлення доступу до акаунтів, реагування на інциденти безпеки та вирішення проблем із несанкціонованим входом. Це може включати як витрати на оплату роботи технічної підтримки, так і втрати продуктивності працівників.

Впровадження двофакторної автентифікації (2FA) дозволяє скоротити ці витрати приблизно на **70%**, оскільки знижується кількість зломів, помилкових входів, а тако ж зменшується потреба в ручному скиданні паролів (15).

Очікувані вигоди від використання системи автентифікації:

- Зменшення втрат від несанкціонованого доступу;
- Скорочення витрат на ручне управління паролями;
- Підвищення захисту персональних і корпоративних даних;
- Зменшення навантаження на службу підтримки.

$$\text{Економія} = 5000 \cdot 0.7 = 3500 \text{ грн/міс}$$

$$\text{За рік: } 3500 \cdot 12 = 3500 \cdot 12 = 42000 \text{ грн}$$

Це означає, що навіть при помірних початкових витратах на впровадження, наприклад, на налаштування серверної логіки, генерацію ключів і налаштування Google Authenticator, система окупається **менш ніж за один рік**.

Таким чином, впровадження системи має гарний рівень економічної ефективності і є доцільним як з точки зору фінансів, так і з позиції безпеки та організаційної ефективності.

Висновки до розділу 5

У цьому розділі було здійснено економічну оцінку ефективності впровадження розробленої системи автентифікації користувачів в інформаційних системах. На основі розрахунків визначено основні витрати на створення та підтримку програмного забезпечення, враховано вартість трудових ресурсів, технічної бази, а також супутніх витрат.

Проведений аналіз продемонстрував, що система має потенціал для широкого впровадження в умовах малого або середнього підприємства без значних фінансових витрат. Очікувані переваги, такі як підвищення рівня безпеки, зниження ризиків несанкціонованого доступу та спрощення процесу входу для користувача, виправдовують витрачені ресурси.

Таким чином, впровадження системи є економічно доцільним і може стати ефективним інструментом захисту для організацій різних масштабів.

ВИСНОВКИ

У дипломній роботі було виконано всебічне дослідження теми «Аналіз сучасних методів автентифікації користувачів в інформаційних системах: паролі, токени, біометрія, 2FA». Метою дослідження було проаналізувати існуючі підходи до захисту доступу до інформаційних систем, оцінити їх переваги та недоліки, а також реалізувати прототип системи з багатофакторною автентифікацією.

У першому розділі проаналізовано основні поняття, пов’язані з автентифікацією користувачів, класифікацію методів доступу, характеристики паролів, токенів, біометричних систем та багатофакторної автентифікації. Виявлено, що жоден метод не є універсальним, але саме поєднання кількох факторів забезпечує максимальний рівень захисту.

У другому розділі розглянуто практичні приклади реалізації різних методів автентифікації у сучасних сервісах і системах. Описано особливості застосування Google Authenticator, одноразових кодів (OTP), біометричних рішень та принципи їх інтеграції в IT-інфраструктуру.

У третьому розділі розроблено програмний прототип авторизації користувачів із підтримкою багатофакторної автентифікації у середовищі Flask. Наведено детальний опис коду, логіки обробки OTP, створення QR-кодів, взаємодії з Google Authenticator, а також результати тестування системи. Практична частина підтвердила ефективність обраних рішень та їх придатність до реального впровадження.

Четвертий розділ було присвячено питанням охорони праці та безпеки у надзвичайних ситуаціях. Визначено вимоги до мікроклімату, освітлення, шуму, пожежної та електробезпеки, а також надано алгоритми дій у разі аварій.

У п’ятому розділі здійснено економічне обґрунтування проекту, проведено розрахунок витрат на реалізацію та оцінено доцільність впровадження. Результати

показали, що система є не лише ефективною з погляду безпеки, а й економічно вигідною для малих та середніх підприємств.

Таким чином, поставлені в роботі завдання були виконані повністю. Було доведено, що впровадження багатофакторної автентифікації дозволяє значно підвищити рівень захисту інформаційних систем без значного ускладнення для користувачів.

Надалі результати цієї роботи можуть бути використані для розширення функціоналу системи, наприклад, шляхом додавання підтримки біометричних методів (розпізнавання обличчя, відбитків пальців), інтеграції з мобільними додатками та хмарними сервісами.

Крім реалізації технічного рішення, у роботі було розглянуто питання охорони праці з урахуванням особливостей роботи в сфері кібербезпеки. Це дозволяє адаптувати середовище праці для фахівців, що працюють з критичною інфраструктурою. Також були розглянуті можливості майбутньої інтеграції біометричних методів автентифікації як частини загального підвищення безпеки.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Бочкарьова І. С. Інформаційна безпека: навчальний посібник. – Київ: Центр учбової літератури, 2021. – 284 с.
2. Ющенко І. П. Захист інформації в комп’ютерних системах: підручник. – Львів: Вид-во ЛНУ, 2020. – 368 с.
3. Біометричні системи автентифікації: аналіз технологій та тенденції розвитку // Науковий журнал «Інформаційні технології». – №2(26), 2023. – С. 55–64.
4. Stallings W. Cryptography and Network Security: Principles and Practice. – 8th ed. – Pearson Education, 2023. – 750 p.
5. Schneier B. Applied Cryptography: Protocols, Algorithms, and Source Code in C. – Wiley, 2016. – 784 p.
6. O'Gorman J. Metasploit: The Penetration Tester’s Guide. – No Starch Press, 2022. – 400 p.
7. RFC 4226 – HOTP: An HMAC-Based One-Time Password Algorithm [Електронний ресурс]. – Режим доступу: https://datatracker.ietf.org/doc/html/rfc4226
8. RFC 6238 – TOTP: Time-Based One-Time Password Algorithm [Електронний ресурс]. – Режим доступу: https://datatracker.ietf.org/doc/html/rfc6238
9. Google Authenticator: Official Documentation [Електронний ресурс]. – Режим доступу: https://github.com/google/google-authenticator
10. Flask Documentation [Електронний ресурс]. – Режим доступу: https://flask.palletsprojects.com/
11. Python Documentation [Електронний ресурс]. – Режим доступу: https://docs.python.org/3/
12. OWASP Authentication Cheat Sheet [Електронний ресурс]. – Режим доступу: https://cheatsheetseries.owasp.org/cheatsheets/Authentication_Cheat_Sheet.html
13. Microsoft Authenticator Documentation [Електронний ресурс]. – Режим доступу: https://learn.microsoft.com/en-us/azure/active-directory/user-help/user-help-auth-app-overview
14. Cloudflare Two-Factor Authentication Guide [Електронний ресурс]. – Режим доступу: https://developers.cloudflare.com/fundamentals/security/authentication/2fa/
15. Eftsure. Two-Factor Authentication Statistics (2024) [Електронний ресурс]. – Режим доступу: https://www.eftsure.com/statistics/two-factor-authentication-statistics/
16. OpenID & OAuth 2.0 Authentication Documentation [Електронний ресурс]. – Режим доступу: https://openid.net/connect/
17. Django Authentication Framework [Електронний ресурс]. – Режим доступу: https://docs.djangoproject.com/en/stable/topics/auth/

18. WebAuthn API documentation – MDN Web Docs [Електронний ресурс]. – Режим доступу: https://developer.mozilla.org/en-US/docs/Web/API/Web_Authentication_API
19. ISO/IEC 27001:2022. Information technology — Security techniques — Information security management systems — Requirements.
20. ДСТУ ISO/IEC 24745:2014. Інформаційні технології. Безпека. Управління біометричною інформацією.

ДОДАТКИ

Додаток А

Фрагмент коду генерації одноразового пароля (OTP) та інтеграції з Google Authenticator

```
import pyotp
from flask import Flask, request, render_template, redirect, session

app = Flask(__name__)
app.secret_key = 'your_secret_key'

@app.route('/')
def index():
    return render_template('login.html')

@app.route('/generate')
def generate():
    secret = pyotp.random_base32()
    session['otp_secret'] = secret
        otp_uri = pyotp.totp.TOTP(secret).provisioning_uri(name='user@example.com', issuer_name='MyApp')
    return render_template('qr.html', otp_uri=otp_uri)

@app.route('/verify', methods=['POST'])
def verify():
    user_input = request.form['otp']
    totp = pyotp.TOTP(session['otp_secret'])
    if totp.verify(user_input):
        return "Авторизація успішна!"
```

```
    else:
        return "Невірний код!"
```

Цей код демонструє просту реалізацію 2FA з використанням Flask і Google Authenticator. Він генерує секретний ключ, створює QR-код для сканування та перевіряє введений користувачем OTP.

Додаток Б

Скріншоти інтерфейсу авторизації з OTP

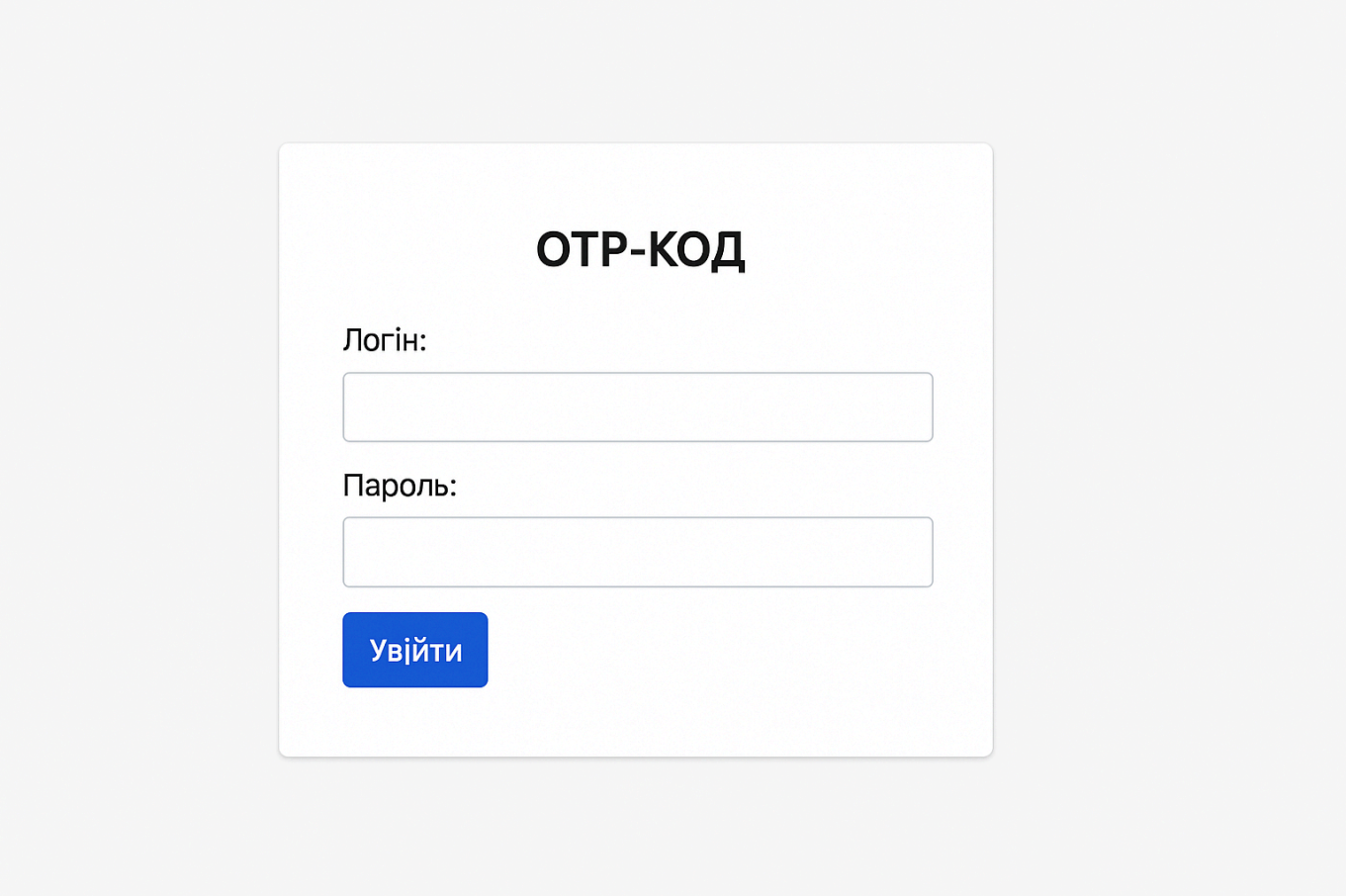


Рис. Б.1 – Головна сторінка авторизації (форма логіну)

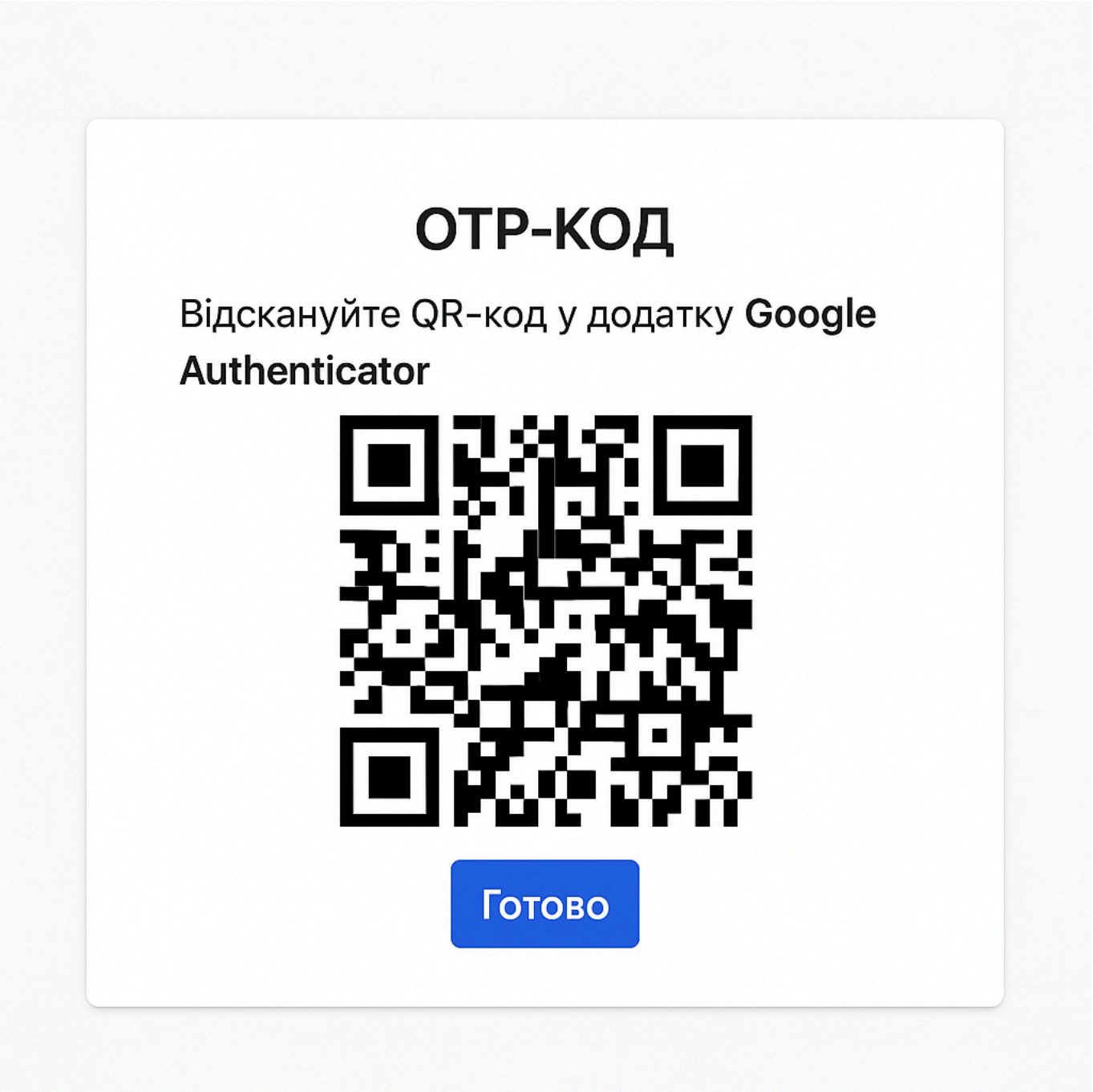


Рис. Б.2 – Генерація QR-коду для Google Authenticator

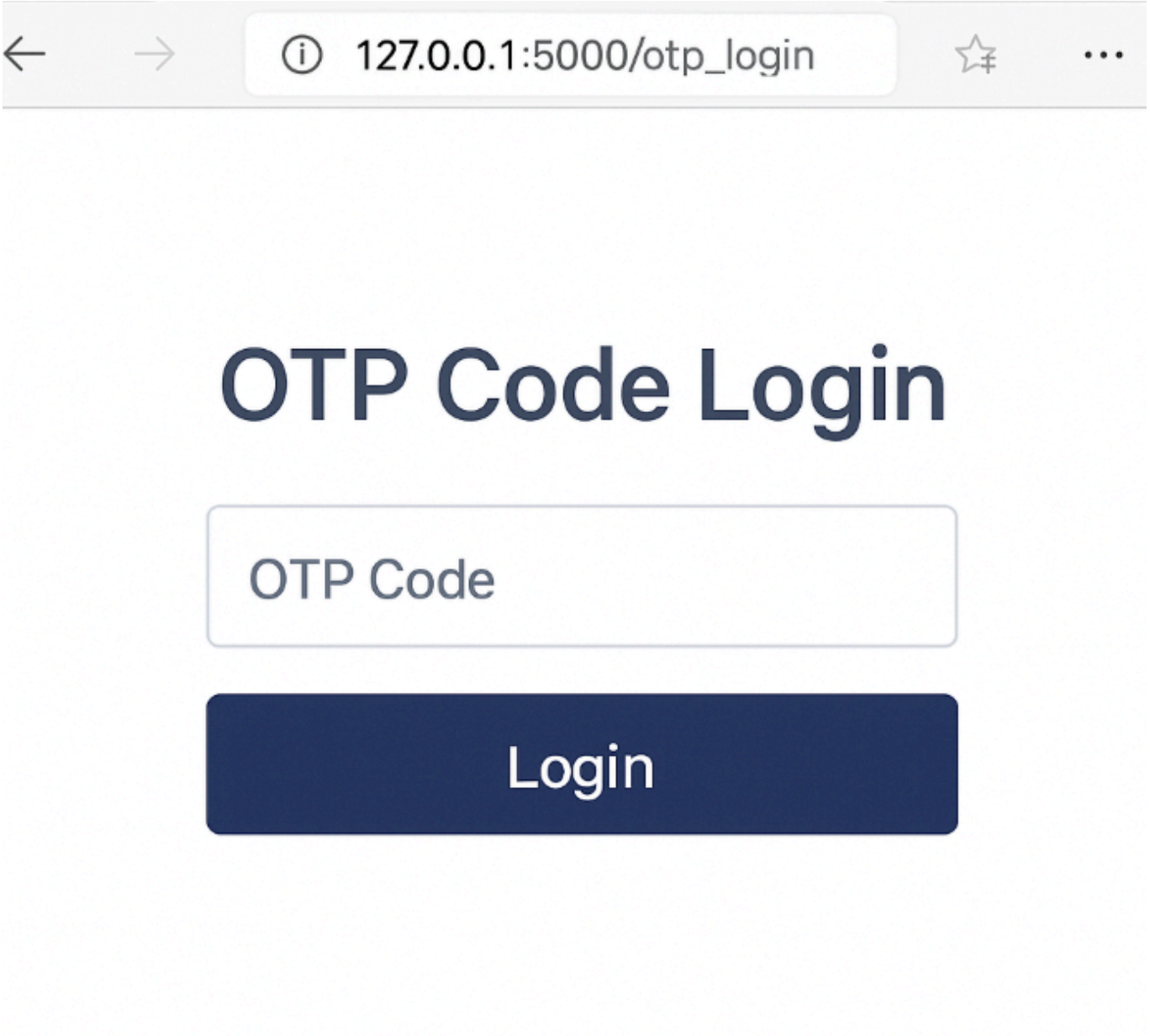


Рис. Б.3 – Введення OTP після сканування

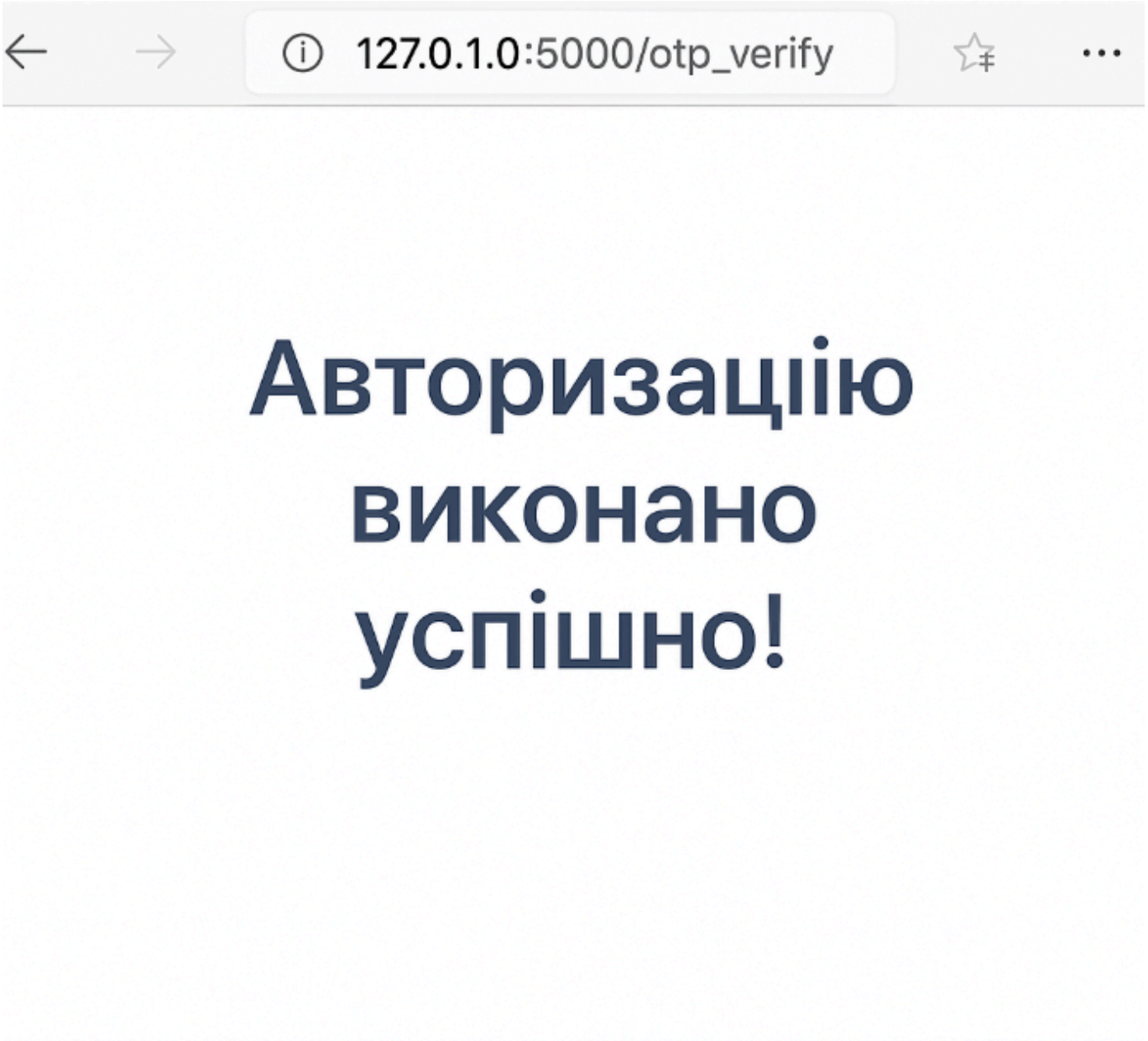


Рис. Б.4 – Повідомлення про успішну авторизацію

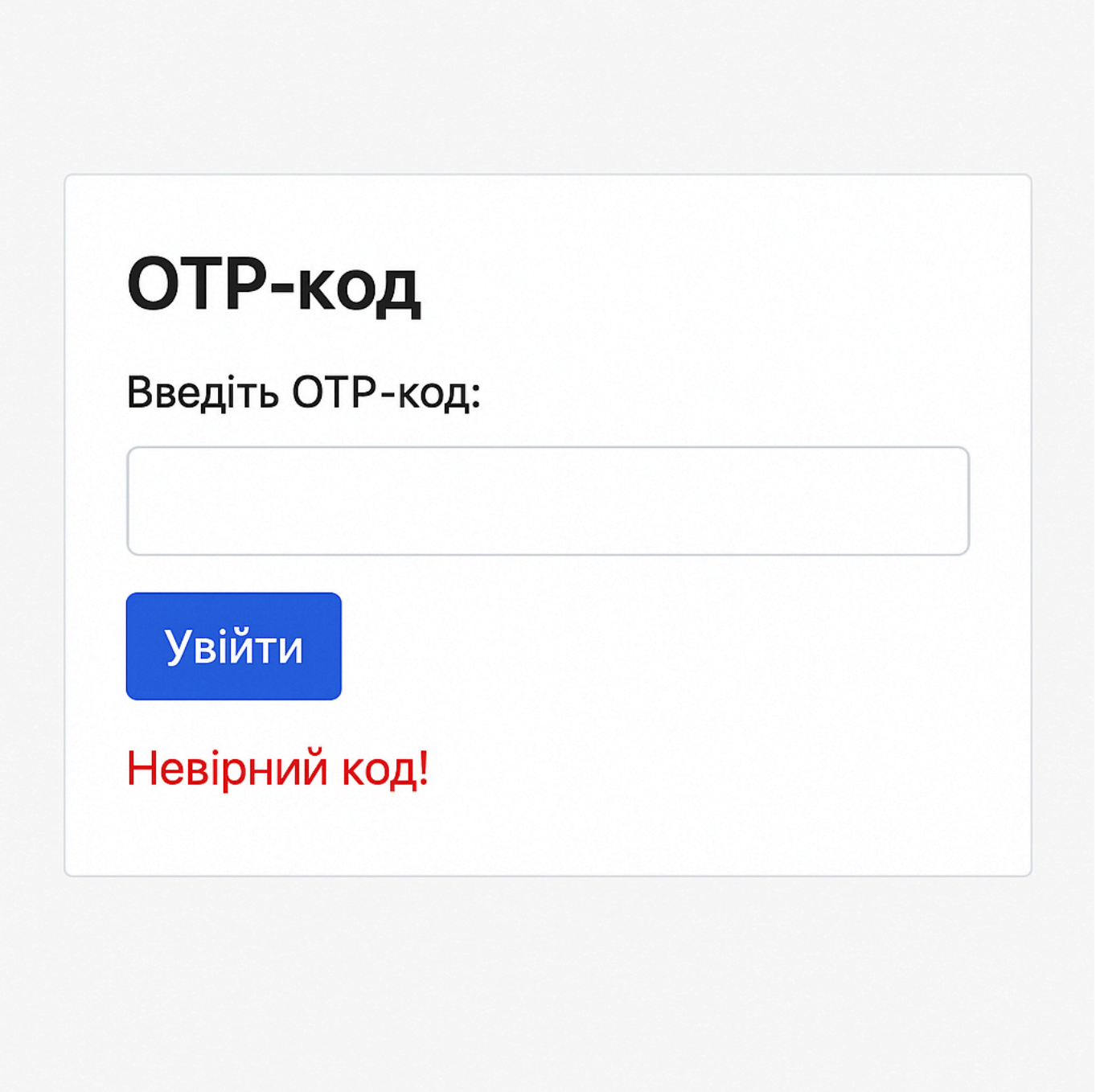


Рис. Б.5 – Повідомлення про помилку, якщо код невірний

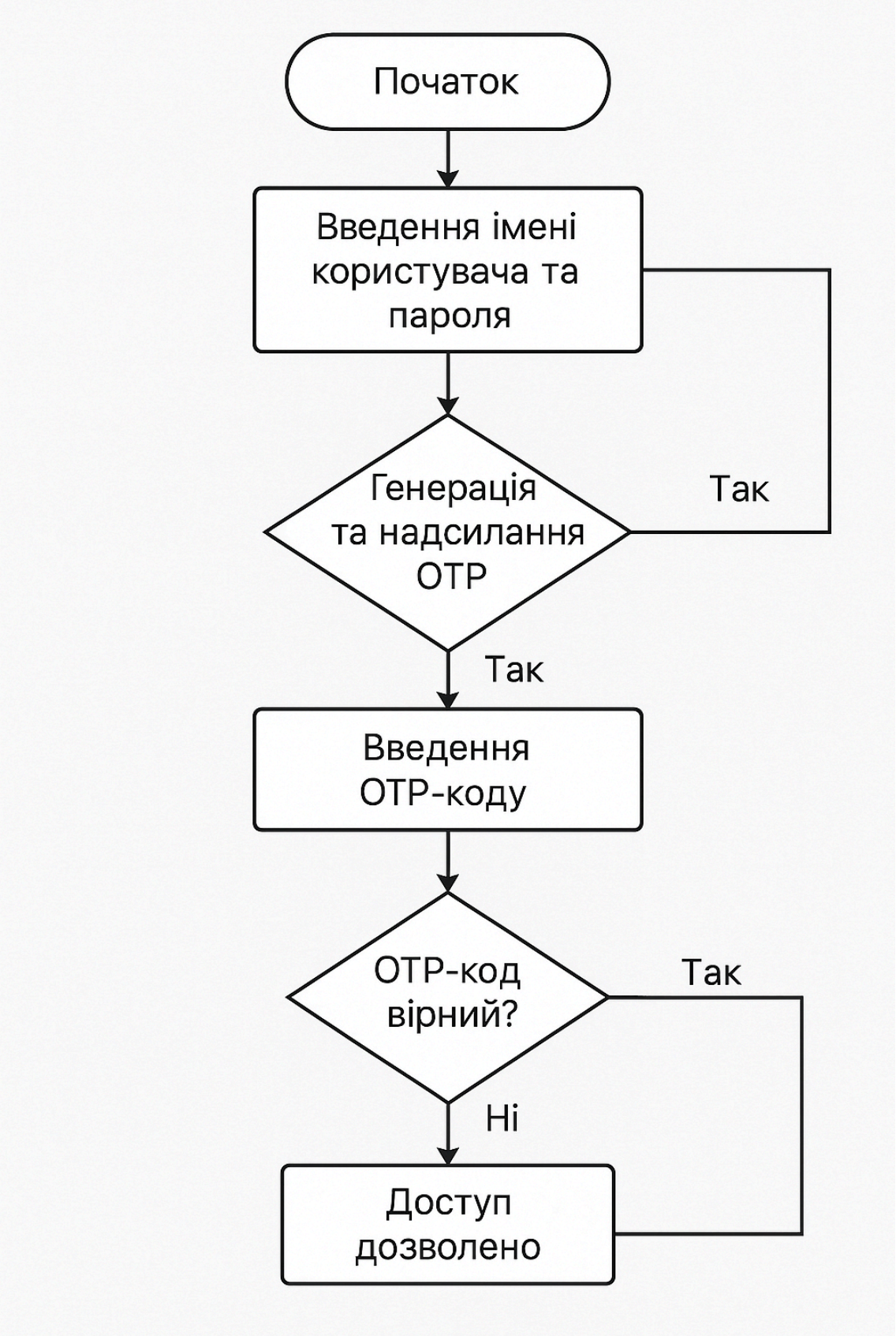
Додаток В

Таблиця тестування результатів роботи системи:

№	**Дія користувача**	**Введені дані**	**Результат**	**Очікуваний результат**	**Статус**
1	Вхід без OTP	Логін+пароль	Помилка	Помилка	Пройдено
2	Генерація QR	-	Отримано QR	QR з секретом	Пройдено
3	Введення правильного OTP	654321	Успішна авторизація	Авторизовано	Пройдено
4	Введення неправильного OTP	123456	Помилка авторизації	Відмова	Пройдено
5	Повторна авторизація через 30 сек.	Новий код	Успішна авторизація	Авторизовано	Пройдено

Додаток Г

Схема логіки роботи 2FA (спрощена блок-схема)



Додаток Д

HTML-шаблон login.html

```
<form method="POST" action="/verify">
  <label for="otp">Введіть OTP-код:</label>
  <input type="text" name="otp" id="otp" required>
  <button type="submit">Увійти</button>
</form>
```