

# 1. Вступ

## 1.1 Призначення

Цей документ визначає вимоги до вебзастосунку PartWork, який допомагає людям знаходити короткострокові підробітки або пропонувати власні послуги.

Система виступає посередником між замовниками (які створюють запит) і виконавцями (які готові виконати завдання за винагороду).

## 1.2 Цілі документа

описати логіку та функціональність продукту;

визначити архітектурні рішення й обґрунтувати використані технології;

задати вимоги до продуктивності, безпеки, UX і розгортання.

## 1.3 Аудиторія

розробники — для реалізації бекенду, фронтенду й бази даних;

тестувальники — для перевірки функціональних сценаріїв;

викладач — для оцінки проєкту.

## 1.4 Обсяг продукту

Застосунок дозволяє користувачам:

створювати короткострокові завдання (наприклад, доставка, ремонт, вигул тварин);

знаходити завдання поблизу;

домовлятися про виконання й оплату;

отримувати сповіщення;

залишати оцінки після завершення роботи.

## 2. Загальний опис

### 2.1 Перспектива продукту

Архітектура: клієнт–серверна, з Single Page Application (SPA) на фронтенді та Service-Oriented Architecture (SOA) на бекенді.

Frontend: React + TypeScript + React Router + Redux Toolkit.

Обрано SPA, бо користувач часто взаємодіє з картою, фільтрами та іншим.

Backend: Python FastAPI + PostgreSQL + SQLAlchemy + JWT-автентифікація.

FastAPI забезпечує високу швидкість обробки запитів і простоту.

Інфраструктура: Docker + Cloud Deployment (Render / Railway).

### 2.2 Основні функції

Реєстрація / авторизація користувачів.

Створення завдань: назва, опис, оплата, адреса, час.

Пошук підробітків: фільтри за містом, ціною, категорією.

Відгуки виконавців: надсилання пропозиції замовнику.

Підтвердження / відхилення завдання.

Оплата після виконання (інтеграція з Stripe/WayForPay).

Система рейтингів і відгуків.

Повідомлення та push-сповіщення.

Адмін-панель модерації.

### 2.3 Класи користувачів

Замовник: створює завдання, вибирає виконавця, залишає відгуки.

Виконавець: переглядає доступні запити, відгукується, отримує оплату.

Адміністратор: контролює користувачів і контент, вирішує спірні ситуації.

### 2.4 Обмеження реалізації

Система не обробляє готівкові операції, лише електронні платежі.

Максимум 10 активних завдань від одного користувача.

## 2.5 Припущення та залежності

Сервери платіжних систем доступні 24/7.

Користувач має стабільне інтернет-з'єднання.

Геолокаційні сервіси працюють через API Google Maps або OpenStreetMap.

## 3. Системні функції

### 3.1 Реєстрація та автентифікація

Пріоритет: високий

Опис: створення облікового запису, підтвердження email, вхід, вихід.

Вимоги:

REQ-AUTH-1: Користувач вводить email + пароль, отримує токен JWT.

REQ-AUTH-2: Підтвердження через посилання з обмеженим терміном.

REQ-AUTH-3: Можливість зміни пароля.

### 3.2 Створення і пошук підробітків

Пріоритет: високий

Опис: замовник створює завдання; виконавці шукають і фільтрують за параметрами.

Вимоги:

REQ-TASK-1: створення завдання з назвою, описом, оплатою, адресою, категорією;

REQ-TASK-2: пошук із фільтрами;

REQ-TASK-3: редагування/видалення активних завдань.

### 3.3 Відгуки виконавців

Пріоритет: високий

REQ-OFR-1: виконавець може відгукнутися на завдання;

REQ-OFR-2: замовник бачить список пропозицій і може вибрати одну.

### 3.4 Підтвердження та оплата

Пріоритет: високий

REQ-PAY-1: після підтвердження виконавця завдання позначається як “У процесі”;

REQ-PAY-2: після завершення робіт виконавець запитує оплату;

REQ-PAY-3: оплата здійснюється через інтеграцію з платіжним шлюзом (Stripe / WayForPay).

### 3.5 Оцінки та рейтинг

Пріоритет: середній

REQ-RAT-1: після виконання сторони залишають оцінки (1–5 зірок);

REQ-RAT-2: рейтинг виконавця оновлюється автоматично;

REQ-RAT-3: система перевіряє дублі та зловживання.

### 3.6 Повідомлення

Пріоритет: середній

REQ-NTF-1: система надсилає email / push-сповіщення про зміни статусів;

REQ-NTF-2: реалізація через Firebase Cloud Messaging.

### 3.7 Адміністрування

Пріоритет: середній

REQ-ADM-1: адмін може блокувати користувачів, переглядати скарги;

REQ-ADM-2: створення звітів про активність.

## 4. Вимоги до зовнішніх інтерфейсів

### 4.1 Інтерфейс користувача

Сучасний мінімалістичний дизайн.

Темна/світла тема, адаптація під мобільні пристрої.

Карта підробіток (через Leaflet).

Форма “Створити завдання”, сторінка профілю, чат між користувачами.

## 4.2 Програмні інтерфейси

інтеграція з Firebase Messaging, Stripe/WayForPay, PostgreSQL.

## 4.3 Комунікаційні інтерфейси

HTTPS / TLS

WebSocket для миттєвих чатів;

SMTP для email-підтверджень.