## **1. Введение**

Проект представляет собой веб-приложение под названием **Real Dopamine Game**, реализующий базовый функционал развлекательного портала с игровыми элементами. Цель проекта — разработать интерфейс и базовую игровую механику с имитацией азартных игр, таких как **слоты** и **мины**, в развлекательных целях. Проект не предусматривает игру на реальные деньги и служит исключительно для демонстрации клиент-серверного взаимодействия, анимаций и интерактивных элементов с использованием Python и HTML/CSS.

## **2. Техническое задание**

### **2.1. Цели и задачи**

Создать минималистичный портал с простыми играми, балансом пользователя, возможностью пополнения виртуального счёта и визуально привлекательным интерфейсом. Основные задачи:

* Реализация игры "Слоты"
* Реализация игры "Мины"
* Отображение текущего баланса
* Интерактивный интерфейс без регистрации
* Динамическая визуализация выигрыша/проигрыша

### **2.2. Функциональные требования**

#### **2.2.1. Основные функции**

* Навигация между разделами (слоты, мины)
* Виртуальный баланс
* Визуализация выигрыша (анимация Джекпота)
* Простая игровая механика с минимальной логикой

#### **2.2.2. Графика и интерфейс**

* Использование Bootstrap 5 для стилизации
* Анимированный фон (дымка, градиент)
* Мобильная адаптация
* Подсветка активной вкладки

### **2.3. Нефункциональные требования**

* Быстрый отклик интерфейса (до 100 мс)
* Работа на локальном сервере
* Отсутствие базы данных (всё в памяти)

## **3. Архитектура**

* Backend: Python (Flask/FastAPI)
* Frontend: HTML5, CSS3 (Bootstrap)
* Все игровые данные хранятся в памяти во время сессии

## **4. API**

### **4.1. Эндпоинты**

* GET / — страница слотов
* GET /mines — страница мин
* POST /spin — прокрутка слота
* POST /play\_mines — запуск игры в мины
* POST /deposit — пополнение баланса

### **4.2. Примеры параметров**

**/spin**

**/deposit**

## **5. Развёртывание**

### **5.1. Требования**

* Python 3.9+
* Flask или FastAPI
* Bootstrap CDN
* Порт: 5000 или 8000

### **5.2. Инструкция**

1. Клонировать репозиторий
2. Установить зависимости (если есть)
3. Запустить сервер:
4. Перейти в браузере на <http://localhost:5000>

## **6. Тестирование**

### **Ручное тестирование:**

* Проверка корректного отображения игр
* Проверка баланса после ставок
* Проверка подсветки активной вкладки
* Проверка визуального отображения Джекпота

## **7. Ограничения**

* Баланс не сохраняется между сессиями
* Нет пользовательских аккаунтов
* Нет антиспама и ограничений по ставкам
* Нет защиты от многопоточности

## **8. Дальнейшее развитие**

* Ведение пользовательской статистики
* Добавление других игр (рулетка, карточные)
* Поддержка регистрации и логина
* Хранение данных в SQLite или PostgreSQL
* Расширенная анимация выигрышей

## **9. Заключение**

Развлекательный портал **Real Dopamine Game** является отличным примером учебного проекта с минимальным стеком технологий. Он даёт представление о создании игрового интерфейса, взаимодействии с пользователем и архитектуре серверного приложения без использования сложных фреймворков и баз данных.