# Курс Full Stack разработчика

## Full Stack Developer Course (Python, Vue.js, FastAPI)

## Этап 1: Основы Python и асинхронного программирования

### 1.1 Основы Python

- [ ] Типы данных и структуры данных:
  - Списки, кортежи, множества, словари
  - Введение в строки, регулярные выражения, обработку ошибок
  - ООП (Классы, наследование, инкапсуляция, полиморфизм)
- [ ] Работа с файлами и библиотеками:
  - Чтение и запись в файлы (JSON, CSV)
  - Использование стандартных библиотек: `os`, `sys`, `datetime`, `logging`

**Практика:**

- [ ] Написание программы для обработки данных (например, анализ данных из CSV)
- [ ] Создание калькулятора с использованием ООП

### 1.2 Асинхронное программирование с Python

- [ ] Основы `asyncio`:
  - Понимание асинхронности в Python: `async def`, `await`
  - Основы работы с циклами событий
- [ ] Асинхронные библиотеки:
  - Работа с `aiohttp` для выполнения асинхронных HTTP-запросов
  - Асинхронная работа с файловой системой с использованием `aiofiles`

**Практика:**

- [ ] Написание асинхронного веб-скрапера
- [ ] Реализация асинхронной отправки сообщений в Telegram

---

## Этап 2: Создание API с FastAPI

### 2.1 Основы FastAPI

- [ ] Маршруты и обработка запросов:
  - Создание REST API с FastAPI
  - Работа с HTTP методами (`GET`, `POST`, `PUT`, `DELETE`)
- [ ] Модели данных с Pydantic:
  - Валидация данных с помощью Pydantic
  - Создание моделей для ввода и вывода данных

**Практика:**

- [ ] Создание простого API для управления пользователями (регистрация, вывод профиля)

### 2.2 Работа с базой данных

- [ ] Интеграция с SQLAlchemy и PostgreSQL:
  - Создание моделей и миграции базы данных
  - Асинхронная работа с базой данных через `Databases` или `Tortoise-ORM`
- [ ] Аутентификация и авторизация:
  - Реализация JWT для аутентификации пользователей
  - Реализация разных уровней доступа (администраторы, пользователи)

**Практика:**

- [ ] Разработка CRUD-операций для пользователя
- [ ] Реализация системы аутентификации и авторизации с JWT

---

## Этап 3: Разработка фронтенда с Vue.js

### 3.1 Основы Vue.js

- [ ] Компоненты и шаблоны:
  - Создание компонентов, передача данных между компонентами
  - Жизненный цикл компонентов
- [ ] Работа с событиями и состоянием:
  - Использование локального состояния в компонентах
  - Введение в Vuex для управления глобальным состоянием

**Практика:**

- [ ] Разработка компонента регистрации и отображения профиля пользователя

### 3.2 Vue Router и интеграция с API

- [ ] Маршрутизация с Vue Router:
  - Реализация переходов между страницами
  - Настройка маршрутов для разных уровней доступа (например, админ и пользователь)
- [ ] Взаимодействие с API:
  - Использование `axios` для отправки запросов на сервер
  - Отображение данных из API на страницах Vue.js

**Практика:**

- [ ] Создание страницы профиля с возможностью редактирования и сохранения данных
- [ ] Создание страницы для отображения списка пользователей

---

## Этап 4: Интеграция FastAPI и Vue.js

### 4.1 Настройка взаимодействия между фронтендом и бэкендом

- [ ] Настройка CORS и безопасности:
  - Разрешение кросс-доменных запросов с помощью FastAPI
  - Обеспечение безопасности с использованием HTTPS
- [ ] Аутентификация и авторизация:
  - Интеграция JWT для авторизации на фронтенде (Vue.js)
  - Защищенные маршруты с проверкой токенов

**Практика:**

- [ ] Интеграция функционала регистрации и авторизации в приложении
- [ ] Разработка админ-панели с управлением пользователями и ролями

### 4.2 Развертывание проекта

- [ ] Контейнеризация с Docker:
  - Создание Docker-образов для бэкенда и фронтенда
  - Настройка мультиконтейнерных приложений с Docker Compose
- [ ] Размещение на сервере:
  - Использование Nginx для проксирования запросов
  - Размещение проекта на облачных платформах (например, AWS, DigitalOcean)

**Практика:**

- [ ] Размещение проекта на сервере с использованием Docker и Nginx
- [ ] Настройка автоматического развертывания с CI/CD

---

## Этап 5: Работа с платежными системами

### 5.1 Изучение казахстанских платежных систем

- [ ] Интеграция с Kaspi.kz:
  - Изучение API Kaspi.kz для приема платежей
  - Настройка вебхуков и обработка транзакций
- [ ] Обработка транзакций:
  - Реализация системы для приема платежей за регистрацию, подписки или товары

**Практика:**

- [ ] Создание системы для управления подписками с интеграцией Kaspi.kz для оплаты

---

## Этап 6: Внутренняя валюта и криптовалюта

### 6.1 Разработка системы внутренней валюты

- [ ] Механизм начисления внутренней валюты:
  - Введение в систему начисления валюты (например, за посещения, достижения)
- [ ] Использование внутренней валюты:
  - Создание витрины товаров/услуг для использования валюты
  - Реализация механизма обмена валюты на товары, услуги, скидки

**Практика:**

- [ ] Реализация системы "ярмарки" для обмена валюты на товары/услуги

### 6.2 Конвертация внутренней валюты в криптовалюту

- [ ] Создание мем-коина на основе TON:
  - Разработка и интеграция токена в систему
  - Механизм обмена внутренней валюты на токены
- [ ] Риски и безопасность:
  - Обеспечение безопасности при работе с криптовалютами

- Защита от фродов и манипуляций

**Практика:**

- [ ] Интеграция с TON для создания токенов и обмена валюты
- [ ] Создание системы вывода средств в криптовалюту

---

## Этап 7: Масштабирование и оптимизация

### 7.1 Оптимизация производительности

- [ ] Кэширование:
  - Использование Redis для кэширования популярных запросов
  - Оптимизация работы с базой данных
- [ ] Масштабирование:
  - Горизонтальное масштабирование с использованием Docker и Kubernetes
  - Балансировка нагрузки и отказоустойчивость

**Практика:**

- [ ] Оптимизация производительности приложения с использованием Redis
- [ ] Масштабирование приложения с Kubernetes

### 7.2 Тестирование и мониторинг

- [ ] Юнит-тестирование и интеграционные тесты:
  - Написание тестов для API и фронтенда
  - Настройка тестирования в CI/CD
- [ ] Мониторинг и логирование:
  - Настройка мониторинга с использованием Prometheus и Grafana
  - Ведение логов с использованием ELK Stack

**Практика:**

- [ ] Написание тестов для API и компонентов Vue.js
- [ ] Настройка мониторинга и логирования приложения

---

## Итоговый проект

- [ ] **Социальная сеть с внутренней валютой**:
  Разработка полностью функциональной социальной сети с регистрацией пользователей, профилями, системой начисления внутренней валюты, интеграцией с Kaspi.kz для платежей и возможностью обмена валюты на товары или криптовалюту.