1. Общие сведения о программе

- Полное наименование программы: Рекомендательная система с защитой от сетевых атак в финансовом секторе
- Обозначение программы: **recommendation_system**
- Назначение:
    - Реализация рекомендательной системы в финансовом секторе
    - Защита от DDoS, SQL-инъекций и XSS через контейнеризацию
- Возможные применения:
    - Банковские рекомендации (кредиты, инвестиции, страхование)
    - Логирование и анализ трафика

2. Функциональное назначение программы

- Цель: реализовать рекомендательную систему с защитой от сетевых угроз
- Функции программы:
    - Nginx: прокси, логирование в JSON, шифрование TLS 1.2+
    - Flask: бэкенд рекомендаций, безопасный поиск через ORM (SQLAlchemy)
    - MySQL: хранение данных рекомендаций
    - Fail2ban : блокировка IP при частых запросах

2.1 Описание функционирования

- Пользователь отправляет запрос через браузер или **curl**
- Nginx проксирует запрос на Flask
- Flask обращается к MySQL для получения рекомендаций
- Nginx записывает логи в JSON
- Fail2ban считывает логи и блокирует подозрительные IP

3. Описание логической структуры программы

3.1 Архитектура системы

Пользователь

[proxy:443 (Nginx)] → [fail2ban: (IP-блокировка)]

[recommendation_system:5000 (Flask)] ← [db:3306 (MySQL)]

3.2 Логика работы

1. Пользовательский запрос:
    - Пример: **GET /search?q=Card HTTP/1.1**
    - IP: **172.21.0.1**, User-Agent: **curl/8.5.0**
2. Nginx (прокси):
    - Перенаправляет запрос на Flask
    - Записывает логи в JSON-формате:json { "@timestamp": "2025-06-05T10:00:00+00:00", "remote_addr": "172.21.0.1", "request": "GET /search?q=Card HTTP/1.1", "status": 200 }
3. Flask (бэкенд):
    - Использует SQLAlchemy для безопасного поиска: python results = Recommendation.query.filter(Recommendation.product.ilike(f"%{query}%")).all()
    - Генерирует HTML-ответ с карточками рекомендаций
4. MySQL (хранилище):
    - Хранит таблицу **recommendations** с полями: sql

      id INT PRIMARY KEY AUTO_INCREMENT,

      product VARCHAR(100),

      reason TEXT
5. Fail2ban:
    - Извлекает IP из логов Nginx
    - Блокирует IP после 50+ запросов за 60 секунд

3.3 Алгоритмы и обработка данных

- Поиск рекомендаций:
    - Пользователь вводит **?q=...**
    - Flask преобразует запрос в **ILIKE** через SQLAlchemy
    - Возвращаются рекомендации, содержащие вхождение запроса
- Логирование:
    - Nginx записывает:
        - Время запроса (**@timestamp**)
        - IP-адрес (**remote_addr**)
        - Тип и путь запроса (**request**)
        - Статус ответа (**status**)
- Безопасность:
    - ORM (SQLAlchemy) предотвращает SQL-инъекции

- JSON-логи Nginx — для анализа и аудита
- Fail2ban (не настроен) — для блокировки подозрительного трафика

4. Используемые технические средства

4.1 Программные средства

- Docker Engine - Контейнеризация сервисов
- Docker Compose - Оркестрация контейнеров
- Nginx - Прокси, логирование в JSON
- Flask - Бэкенд рекомендаций, обработка запросов
- MySQL - Хранение данных рекомендаций
- Fail2ban - Блокировка IP
- Bootstrap - Фронтенд: стилизация HTML-карточек

4.2 Аппаратные средства

- Хост:
  - RAM: 4 GB+
  - CPU: 2 ядра+
  - Диск: 20 GB свободного места
- Серверы:
  - Локальная машина (для тестирования)
  - Сервер с поддержкой Docker

5. Вызов и загрузка программы

5.1 Подготовка

Перейдите в папку проекта

cd ~/recommendation_system

Убедитесь, что Docker и Compose установлены

docker --version && docker-compose –version

5.2 Сборка и запуск

Соберите и запустите систему

docker-compose down

docker-compose build

docker-compose up -d

5.3 Проверка работы

Проверьте, все ли контейнеры запущены

docker ps

Проверьте логи Nginx

docker logs recommendation_system_proxy_1

Проверьте данные в MySQL

docker exec -it db mysql -u root -padmin -e "SELECT * FROM recommendation_db.recommendations"

6. Входные данные

6.1 Пользовательский запрос

- Пример: **GET /search?q=Card HTTP/1.1**
- Поля:
    - **q** — строка поиска
    - **Host**, **User-Agent**, **X-Forwarded-For** — для анализа

6.2 IP-адрес клиента

6.3 MySQL-данные

- Таблица: **recommendations**
- Поля: **id**, **product**, **reason**

7. Выходные данные

- HTML-интерфейс
- JSON-логи Nginx
- MySQL-данные
- Логи Fail2ban