

СОДЕРЖАНИЕ

1 ВВЕДЕНИЕ	3
1.1 Общая информация о проекте	3
1.2 Общая характеристика деятельности заказчика	4
1.3 Описание задания по проектной практике	4
1.4 Описание достигнутых результатов по проектной практике	5
2 ЗАКЛЮЧЕНИЕ	6
СПИСОК ИСТОЧНИКОВ	7

1 ВВЕДЕНИЕ

1.1 Общая информация о проекте

1.1.1 Название проекта

Lifeguard

1.1.2 Описание проекта

Мобильное, десктопное и web-приложение для автоматизации процессов пожарной охраны, включающее:

- учёт дежурных смен;
- учёт мероприятий;
- учёт инвентаризации;
- автоматизацию документооборота.

1.1.3 Цель проекта

Создание единой информационной системы для решения следующих задач:

- облегчение бюрократических процессов в пожарной охране;
- централизованное хранение и обработка данных о личном составе и инвентаре;
- предоставление персоналу удобного инструмента для управления расписанием мероприятий.

1.1.4. Актуальность проекта

В настоящее время функции, необходимые для работы пожарной охраны, разрознены между различными приложениями, многие из которых не адаптированы под специфику данной сферы. Lifeguard объединяет все ключевые аспекты в одном решении, что повышает эффективность работы спасательных служб.

1.1.5. Ожидаемый результат

Готовое к внедрению мультиплатформенное приложение (мобильное, десктопное и web-версия) для ОАСФ СПСО "Феникс".

1.2 Общая характеристика деятельности заказчика

1.2.1 Наименование заказчика

Студенческий пожарно-спасательный отряд "ФЕНИКС".

1.2.2 Организационная структура

Добровольное объединение студентов, специализирующееся на:

- подготовке спасателей-волонтеров;
- организации мероприятий по безопасности;
- проведении обучающих курсов.

1.2.3. Описание деятельности

Деятельность отряда:

- подготавливает курсантов к аттестации на спасателей-добровольцев;
- обеспечивает безопасность на общественных мероприятиях;
- проводит мастер-классы;
- проводит учебно-тренировочные сборы;
- участвует в субботниках.

1.3 Описание задания по проектной практике

Задание включало разработку модуля для классификации SMS-сообщений (спам/не спам) с использованием машинного обучения.

Основные этапы:

1. Анализ и выбор алгоритмов (ChooseClassifier.py):

- тестирование классификаторов (SVC, RandomForest, Naive Bayes...);
- выбор SVC с TfidfVectorizer как оптимального решения.

2. Обучение модели (TrainModel.py):

- векторизация текста (TfidfVectorizer);
- обучение SVC на предоставленном датасете;
- экспорт модели через joblib.

3. Развёртывание системы (main.py):

- создание Flask-приложения для тестирования модели.

1.4 Описание достигнутых результатов по проектной практике

- Выбор оптимального классификатора для обучения;
- достижение точность модели: 98.2%;
- реализован веб-интерфейс для проверки сообщений.

2 ЗАКЛЮЧЕНИЕ

В ходе прохождения практики была выполнена разработка части многофункционального приложения для автоматизации процессов пожарной охраны.

Основные результаты работы:

1. Разработан класс логгера с методами Info, Warning и Error, что позволило улучшить мониторинг работы приложения и оперативно выявлять критические ситуации.
2. Помощь в разработке белой темы приложения.
3. Помощь в разработке страниц документооборота, просмотра смен.

СПИСОК ИСТОЧНИКОВ

- Scikit-learn документация: https://scikit-learn.org/stable/user_guide.html;
- документация Flask: <https://flask.palletsprojects.com/en/stable/>;
- документация библиотеки joblib: <https://joblib.readthedocs.io/en/stable/>;
- информация о LifeGuard:

<https://projects.mospolytech.ru/tproduct/413315123-369351020332-lifeguard>.