

обзор аналогов и выявлены положительные и отрицательные стороны аналогов и прототипов, сформированы требования к проекту и определен функционал приложения по ролям: пользователя и администратора, выполнено проектирование данных.

Для разработки выбраны следующие технологии: платформа Node.js [1], основанная на движке V8, превращающая JavaScript из узкоспециализированного языка в язык общего назначения; библиотека React [2] для разработки пользовательского интерфейса. В качестве фреймворка выбран Express [3], который является фреймворком web-приложений для Node.js, позволяющий создавать API приложения. Выполненное тестирование позволяет сделать вывод, что приложение работает корректно, цели достигнуты, а все задачи выполнены в полном объеме.

ЛИТЕРАТУРА

1. Документация | Node.js [Электронный ресурс] – Режим доступа: <https://nodejs.org/ru/docs> - Дата доступа 17.04.2023.
2. Документация | React [Электронный ресурс] – Режим доступа: <https://ru.reactjs.org/docs/getting-started.html> - Дата доступа 17.04.2023.
3. Документация | Express [Электронный ресурс] – Режим доступа: <https://expressjs.com/ru/guide/routing.html> - Дата доступа 17.04.2023.

УДК 004.89 + 316.472.4 + 159.942.5

Студ. А.А. Ковалев

Науч. рук. ст. преп. А.С. Наркевич
(Кафедра программной инженерии, БГТУ)

СОЦИАЛЬНАЯ СЕТЬ «ТОВЕ»

Целью проекта является разработка социальной сети «ТоВе», одной из особенностей которой является использование модели машинного обучения в качестве инструмента оценки эмоционального состояния пользователя на основании сообщений, отправляемых пользователем в чатах.

Нейронная сеть, созданная методом обучения с учителем на основе более пятидесяти тысяч записей тренировочного набора данных, была использована как инструмент оценки текстовой составляющей пользовательских сообщений. Пользователю предоставляются результаты её работы, а именно – статистические данные, сформированные на основе анализа сообщений пользователя по четырём эмоциям: радость, грусть, скука и гнев, – и эмоции «нейтральности», значение

оценки которой использовалось в случае отсутствия явного выражения эмоций через текст сообщений пользователя. Пользователю также предоставляется возможность ведения «дневника эмоций» позволяющего создавать личные записи и оценивать эмоциональную составляющую этой записи по тем же пяти характеристикам. Это позволяет контролировать среднее значение оценки эмоционального состояния пользователя, которое доступно другим пользователям в качестве статистической сводки в профиле. Дополнительно для дальнейшего улучшения работы нейронной сети интерфейс пользователя позволяет самостоятельно оценивать текстовое содержимое сообщений в собственных чатах.

Практическая значимость работы заключается в возможности улучшения пользовательского опыта использования социальной сети, развития навыка саморефлексии и оценки собственного психологического состояния пользователя, а также персонификации графического интерфейса на основании данных, полученных в результате анализа.

Дальнейшее развитие и улучшение подобной технологии может упростить процесс идентификации некоторых аффективных расстройств лёгкой и средней степени тяжести человека, а также помочь узнать больше о личности и эмоциональном состоянии пользователей наравне с теми данными о себе, которые пользователи предоставляют в социальных сетях собственноручно.

УДК 004.588

Студ. А.В. Михалькевич
Науч. рук. ст. преп. А.С. Наркевич
(Кафедра программной инженерии, БГТУ)

КАТАЛОГ МАГАЗИНОВ КОМПЬЮТЕРНОЙ ТЕХНИКИ

В настоящее время активно развиваются покупки через интернет-каталоги, которые позволяют пользователям онлайн, в своём браузере или через мобильное приложение, сформировать заказ на покупку, выбрать способ оплаты и доставки заказа, оплатить заказ.

Целью данного диплома является создание маркетплейса для продажи компьютерной техники. Приложение реализовано при помощи языка программирования C# с использованием кроссплатформенного фреймворка для создания веб-приложений – ASP.NET Core. Для реализации клиентской части была использована библиотека Javascript – React. Помимо этого, была реализована микросервисная архитектура. Также применены два паттерна проектирования микросервисной архитектуры: Shared-database-per-service и Synchronous