

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
“ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ”

Факультет *компьютерных наук*

Кафедра *программирования и информационных систем*

Awareness

Веб-приложение для для повышения самоосознанности

Курсовая работа

09.03.02 Информационные системы и технологии

Обучающиеся _____ Рыбас Д.В, Соболев И.И, Светачев П.С., Попенко
А.П., 3 курс

Руководитель _____ *В.С. Тарасов, преподаватель*

Воронеж 2022

Содержание

Содержание	2
Введение	3
1. Постановка задачи	4
2. Анализ предметной области	6
2.1. Глоссарий	6
2.2. Анализ аналогов	8
2.3. Анализ потребности	9

Введение

Становление человека как самоосознанной личности, пожалуй, один из основных трендов последних лет. Раньше люди не стремились к этому так, как сейчас по причине не закрытых основных материальных потребностей. Им, в первую очередь, нужно было думать о том, как прокормить себя и свою семью. Сейчас же у многих людей эти потребности закрыты, вследствие чего появились новые - духовные. Современный человек ищет ответы на вопросы: “Как мне быть самим собой и понимать, что я делаю в тот или иной момент?” “Какие чувства я испытываю ежедневно?”. Можно обратиться к психологу за помощью в поиске ответов на данные вопросы, но не все люди имеют средства на это, а те, кто обратился к нему, не могут держаться с ним на связи на протяжении всего дня

Из сказанного выше следует, что людям необходимо место, в котором они будут структурировать свои мысли, отслеживать привычки, повышать самоосознанность. Можно заниматься этим, используя листок бумаги и ручку, что является неудобным в силу отсутствия автоматизации, но в эпоху цифрового мира, можно исправить данный недостаток.

Сказанное выше позволяет сделать вывод, что разработка веб-приложения на эту тему имеет высокую актуальность по причине большой востребованности людей в этом.

В данной курсовой работе будет описан процесс разработки такого веб-приложения.

1. Постановка задачи

Цель данного курсового проекта – разработка веб-приложения, являющегося персональным помощником пользователя для повышения его осознанности путём ведения дневника по шаблону, дневника ситуаций, дневника эмоций, а также отслеживания привычек.

Задачи, которые необходимо решить в ходе развития проекта:

- Сбор информации об исследуемой предметной области;
- Проведение анализа полученных данных;
- Создание диаграмм, отражающих основные аспекты приложения;
- Разработка веб-приложения;
- Проведение тестирования веб-приложения;
- Размещение приложения на сервере.

Разработка веб-приложения разделена на разработку серверной и клиентской частей.

Разработка серверной части:

- Проработка и реализация базы данных;
- Реализация работы с используемыми API;
- Формирование запросов к базе данных.

Разработка клиентской части:

- Проработка дизайна веб-приложения;
- Формирование веб-страниц приложения;
- Реализация взаимодействия с серверной частью.

К приложению предъявляются следующие требования:

- предоставить пользователю возможность ведения дневника по шаблону. Пользователю будет предлагаться определённый шаблон, состоящих из тем для размышления, в который он будет вписывать свои развернутые ответы;
- предоставить пользователю возможности ведения дневника эмоций. Пользователю будет предлагаться список эмоций, в котором он будет выбирать те, что испытал за день и список действий, в котором он будет выбирать те, что сделал в этот день, что позволит связывать эмоции человека с его действиями, а также предлагать вариант проведения дня;
- предоставить пользователю возможность ведения дневника ситуаций. Постановка приложением пользователя мысленно в определённые жизненные ситуации с предложением вариантов ответа. При выборе определенного варианта ответа, пользователь будет получать потенциальный исход, а также совет, следует ли ему поступить иначе и поменять свой выбор или остановиться на данном;
- предоставить клиенту возможность узнать, сколько дней подряд он имеет привычку;
- предоставить клиенту возможность создать обращение в техническую поддержку и узнать ответ на это обращение внутри приложения;
- предоставить клиенту возможность ведения трекера привычек в виде календаря на неделю, месяц, год;

— предоставить неавторизованному пользователю возможность регистрации, авторизации, смены пароля.

2. Анализ предметной области

2.1. Глоссарий

В курсовой работе будут использоваться следующие термины:

Администратор - Специалист, уполномоченный управлять и администрировать определенный участок IT-среды.

Веб-приложение (сайт) - Клиент-серверное приложение, в котором клиент взаимодействует с веб-сервером при помощи браузера.

Дневник по шаблону - Предложенные приложениям шаблоны для заполнения дневника.

Дневник ситуаций - Мысленная постановка приложением пользователя в определённые жизненные ситуации с предложением вариантов ответа.

Дневник эмоций - Заполнение пользователем эмоций из списка, испытанных за день и действий, которые он делал за этот день, что позволит связывать эмоции с действиями.

Пользователь - Лицо, которое использует приложение для выполнения конкретной функции.

Репозиторий - Место, где хранятся и поддерживаются какие-либо данные.

СУБД - Система управления базами данных. Комплекс программно-языковых средств, позволяющих создать базы данных и управлять данными.

Яндекс-метрика - Бесплатный интернет-сервис компании Яндекс, предназначенный для оценки посещаемости веб-сайтов и анализа поведения пользователей.

Android - Операционная система для мобильных устройств.

API - Программный интерфейс приложения. Описание способов, которыми одна компьютерная программа может взаимодействовать с другой программой.

Bootstrap - Свободный набор инструментов для создания сайтов и веб-приложений.

CSS - Формальный язык, служащий для описания оформления внешнего вида документа, созданного с использованием языка разметки (HTML, XHTML, XML).

FAQ - Собрание часто задаваемых вопросов по какой-либо теме и ответов на них.

Flask - Фреймворк для создания веб-приложений на языке программирования Python, использующий набор инструментов Werkzeug, а также шаблонизатор Jinja2.

GitHub - Сервис для совместной разработки и хостинга проектов.

HTML - Стандартизированный язык разметки веб-страниц во Всемирной паутине.

JavaScript - Мультипарадигменный язык программирования, используется как встраиваемый язык для программного доступа к объектам приложений.

Linux - Семейство Unix-подобных операционных систем на базе ядра Linux.

PostgreSQL - Свободная объектно-реляционная система управления базами данных.

Python - Высокоуровневый язык программирования общего назначения.

REST - Архитектурный стиль взаимодействия компонентов распределённого приложения в сети.

SQLAlchemy - Программная библиотека на языке Python для работы с реляционными СУБД с применением технологии ORM.

Windows 10 - Операционная система для персональных компьютеров и рабочих станций, разработанная корпорацией Microsoft в рамках семейства Windows NT.

2.2. Анализ аналогов

Для формулирования более конкретных требований к веб-приложению необходимо было провести анализ аналогов. К ним можно отнести такие приложения как, например, AVE, VOS, Daylio.

AVE - это приложение, позволяющее пользователю повышать уровень самоосознанности.

Сильные стороны сервиса:

- наличие заданий, которые пользователь ставит себе на день;
- наличие ленты новостей, где пользователи делятся своими мыслями;
- наличие возможности тренировки различных навыков, указанных в приложении;
- наличие системы достижений.

Слабые:

- наличие платной подписки;

- система уровней, по причине которой невозможно сразу после установки приложения раскрыть для себя всю его функциональность.

VOS – это приложение, позволяющее пользователю повышать уровень самоосознанности.

Сильные стороны сервиса:

- возможность ведения дневника;
- возможность отслеживания собственных эмоций пользователем.

Слабые:

- наличие платной подписки;
- меньшая функциональность в сравнении с AVE.

Daylio – это приложение, позволяющее пользователю отслеживать собственные эмоции.

Сильные стороны сервиса:

- возможность отслеживания эмоций;
- возможность увидеть график настроения.

Слабые:

- наличие платной подписки;
- меньшая функциональность в сравнении с вышеперечисленными аналогами.

2.3. Анализ потребности

После изучения аналогов разрабатываемого проекта можно сделать вывод, что ни один из них не является идеальным решением поставленной задачи. Но существование большого количества сервисов для повышения

самоосознанности говорит о том, что пользователь нуждается в этом. Именно поэтому разработка проекта Awareness является актуальной. Команде разработчиков необходимо учесть достоинства аналогов, а также их недостатки, которые нужно попытаться исправить.