

старые записи, сохранить, автоматическое сохранение. При нажатии на кнопку «Показать записи» на экран выводится информация из файла о всех сохранённых записях, данная функция также выполняется автоматически при открытии окна удаления записи. Функция удаления записи вызывается нажатием на кнопку с иконкой крестика после чего будет удалена выбранная запись. Удаление старых записей осуществляется при нажатии на соответствующую кнопку, при этом удаляются данные относительно текущей системной даты. Сохранение изменений вызывается при нажатии на кнопку «Сохранить» либо клавишу “Enter”, после чего все изменения будут сохранены в файле. Для автоматического сохранения изменений необходимо установить галочку в поле «Автоматическое сохранение», после чего функция, выполняемая кнопкой «Сохранить» будет выполняться автоматически при любом удалении данных. Функция автоматического сохранения изначально отключена для предотвращения случайных изменений. Для закрытия формы необходимо нажать на кнопку «Закрыть».

Окно редактирования вызывается двойным нажатием на запись в компоненте “ListBox”, после чего будет вызвана соответствующая форма, содержащая в себя поля даты, времени и события, которые заполняются автоматически в зависимости от выбранной записи. Для изменения записи необходимо изменить поле или поля формы и нажать на кнопку «Сохранить». Если перед редактированием записей была нажата кнопка «Сортировать» изменения будут сохранены в отсортированном виде. Закрытие диалогового окна выполняется автоматически при нажатии на кнопку «Сохранить».

Заключение. В ходе выполнения работы было создано приложение, выполняющее роль электронного ежедневника, для создания которого понадобились знания работы с файлами, строковым типом данных, средствами создания GUI приложений, а также навыки работы с динамической памятью для оптимизации работы интерфейса программы. Данное приложение предназначено для применения обычным пользователем с целью планирования и дальнейшего редактирования собственного расписания. Разработанное приложение является интуитивно понятным, приятным в использовании, а также не требующим большого количества ресурсов системы, что позволяет использовать его на не самых мощных компьютерах. Поиск расписания не занимает много времени благодаря функциям сортировки, а также поиска информации по дате, времени и событию. В дальнейшем допустима доработка приложения с упрощением старых функций или добавлением новых.

Список цитируемых источников

1. Общие сведения о Visual Studio [Электронный ресурс]. — Режим доступа : <https://docs.microsoft.com/ru-ru/visualstudio/get-started/visual-studio-ide?view=vs-2022> . — Дата доступа : 03.05.2022.
2. Введение — MFC [Электронный ресурс]. — Режим доступа : <http://www.codenet.ru/progr/visualc/mfc/mfc1.php> . — Дата доступа : 30.04.2022.

УДК 004.55

Е. И. Гуринович, Г. М. Раковцы

Учреждение образования «Барановичский государственный университет», Барановичи, Беларусь

РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЯ ДЛЯ КОНТРОЛЯ РАБОЧЕГО ВРЕМЕНИ СОТРУДНИКОВ С ИСПОЛЬЗОВАНИЕМ DJANGO FRAMEWORK

Введение. После появления факторов, мешающих сотрудникам выполнять свою работу на рабочих местах в офисах, появилась необходимость альтернативы, которая поможет сохранить эффективность работы без нанесения убытков. Компании представляют своим сотрудникам возможность работы из дома и офиса. Это дает возможность сотруднику выбрать свой ритм в работе. Дистанционная работа позволяет экономить время и деньги. Но помимо всего этого сотрудник должен обладать самоорганизацией и самоконтролем. В связи с этим для менеджеров стало более сложно контролировать занятость сотрудников.

В настоящее время для разработки веб-приложений использование облачных технологий является одной из неотъемлемых частей.

Облачные технологии — IT-технологии, которые позволяют хранить и обрабатывать информацию на удалённых серверах. Облачные технологии позволяют перенести вычислительные процессы и хранение данных на удалённые серверы, а пользователю для доступа к ним и работы с ними достаточно иметь выход в интернет со своего личного устройства [1].

Также при создании веб-приложений необходимо знать и учитывать архитектурный стиль взаимодействия REST. Данный набор правил организации написания кода серверного приложения поможет программисту сделать код не только понятным, но и более продуктивным. Чем больше приложение соответствует правилам REST, тем более оно является RESTful.

Хорошей практикой для разработки веб-приложений является реализации технологии CI/CD (непрерывная интеграция/непрерывное развертывание).

CI/CD — набор методов и практик, отвечающий требованиям современной ПО-разработки. Принципы непрерывной интеграции и доставки помогают внедрять решения быстро, оперативно согласовывать их и доводить до релиза, не рискуя при этом качеством продукта.

Цели CI/CD:

– обеспечение последовательного и автоматизированного способа сборки, упаковки и тестирования продуктов или приложений;

– автоматизация развертывания в разных окружениях;

– сведение к минимуму ошибок и проблем.

Основная часть. Цель данного проекта — разработать веб-приложение, которое позволит менеджеру контролировать рабочее время сотрудников, а также поможет бухгалтерии рассчитывать заработную плат. Данное приложение упростит сотруднику организацию рабочего процесса.

Для разработки приложения применялась среда программирования PyCharm Community Edition 2021.1.1 на языке программирования Python при помощи инструментов Django Framework. Выбор данного фреймворка обусловлен простым развертыванием проекта на хостинге.

Django — это программный каркас с богатыми возможностями, подходящий для разработки сложных сайтов и веб-приложений, написанный на языке программирования Python. Архитектура Django похожа на “Model-View-Controller” (далее — MVC). Контроллер классической модели MVC примерно соответствует уровню, который в Django называется «Представление» (View), а презентационная логика «Представления» реализуется в Django уровнем «Шаблонов» (Templates). Из-за этого уровневую архитектуру Django часто называют «Модель-Шаблон-Представление» [2].

Веб-приложение было реализовано по архитектуре «Клиент-Сервер». После запроса к веб-приложению работа производится с REST Framework, который играет роль программного интерфейса (далее — API). Далее API запрашивает данные в базе данных. Для данного приложения используется база данных PostgreSQL.

PostgreSQL — это популярная свободная объектно-реляционная система управления базами данных. PostgreSQL базируется на языке SQL и поддерживает многочисленные возможности [3].

Данная база данных удобна для работы с облачной платформой Heroku, для непрерывного развертывания приложения. Также для хранения медиа-файлов было использовано облачное хранилище DropBox.

Для работы с автоматизацией процесса непрерывной интеграции и непрерывного развертывания был использован сервис Github Actions.

Для визуализации работы приложения представлена диаграмма вариантов использования на рисунке 1.



Рисунок 1 — Диаграмма вариантов использования

Перед тем как работать с веб-приложением, необходимо включить сервер или, если приложение будет выгружено на хостинг, проверить работоспособность хоста сервера и перейти по адресу.

После загрузки веб-приложения появится форма для авторизации с помощью email/пароль, других веб-сервисов (рисунок 2), иначе необходима форма регистрации, которая использует дополнительное подтверждение email-адреса.

После успешной авторизации пользователь перенаправляется на главную страницу.

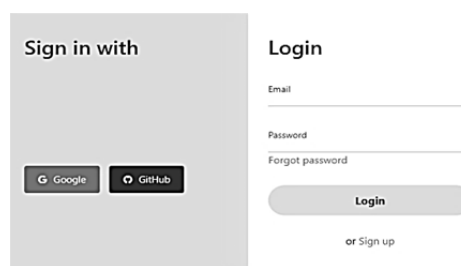


Рисунок 2 — Форма авторизации пользователя

На данной странице есть 2 блока, который разбиты для разделения данных учета работы за неделю и оповещения по запросам на отпуск (рисунок 3).

Следующая вкладка представляет собой подробную информацию о заданиях, которые выполняет пользователь. В данной вкладке есть возможность поиска по промежутку времени, далее эти данные можно отправить в файл с форматом .xlsx для просмотра в приложениях Microsoft Excel, LibreOffice Calc. На данной вкладке создаются новые виды деятельности в проекте, добавляются часы к уже существующим заданиям и удаляются в случае необходимости (рисунок 4).

Следующая вкладка представляет собой список запросов пользователя на отпуск, а также запросы пользователей, для которых необходимо подтверждение запросов отпусков. В запросы пользователей попадают те, для которых текущий пользователь является менеджером (рисунок 5).

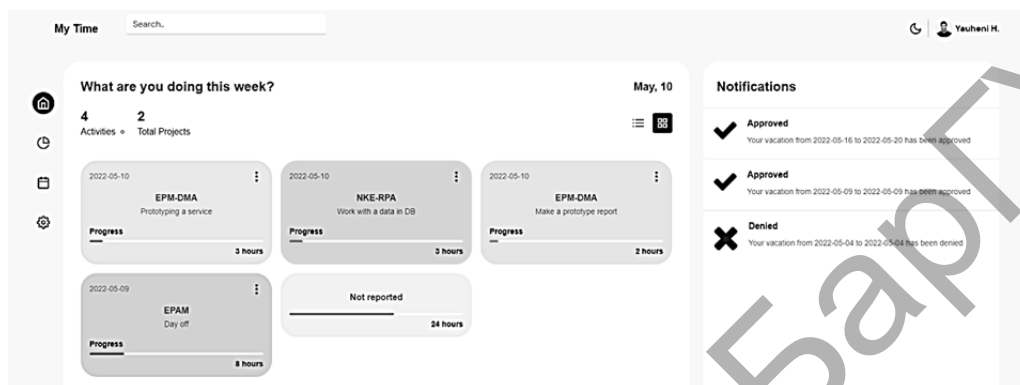


Рисунок 3 — Главная страница приложения

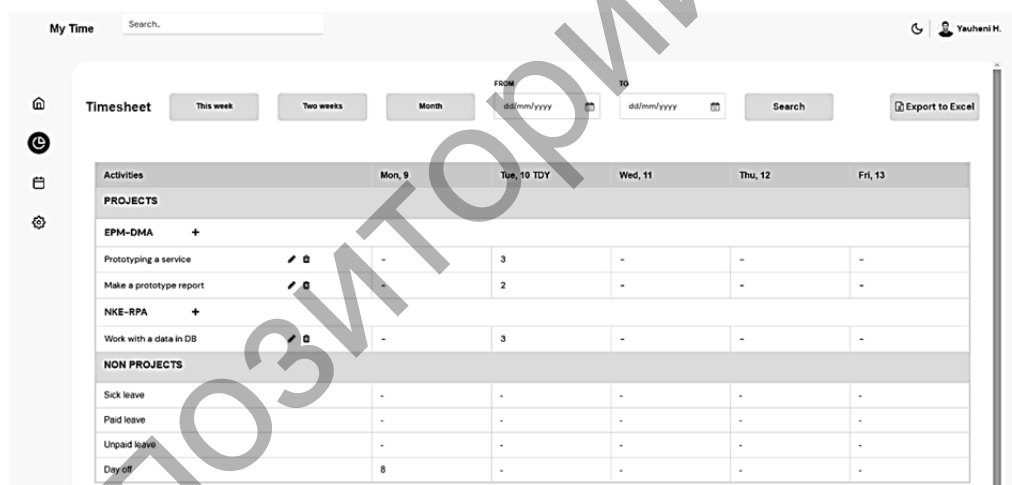


Рисунок 4 — Страница задач пользователя приложения

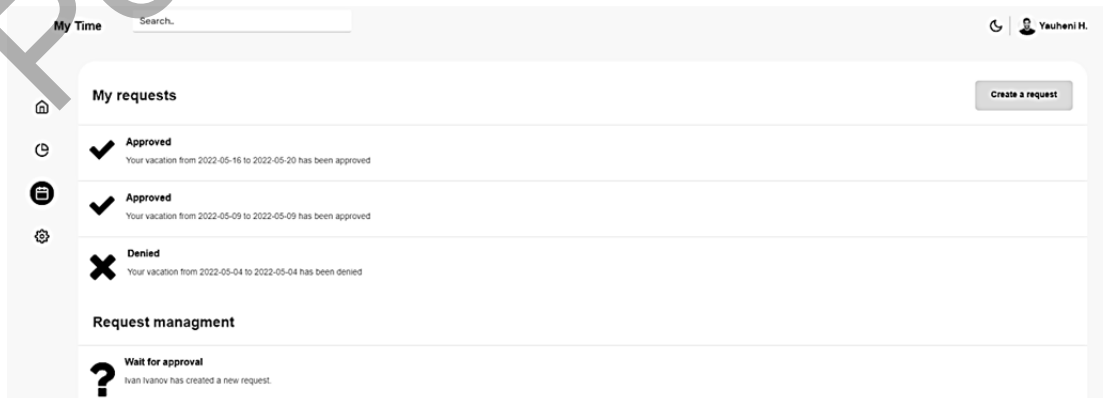


Рисунок 5 — Страница запросов пользователя на отпуск

Последняя вкладка необходима для изменения основных данных пользователя, то есть настройки сотрудника. Предусмотрены такие настройки, как изменение фотографии, имени, фамилии, email-адреса, пароля, подключение/отключение веб-сервисов аккаунта (рисунок 6).

Также пользователь может просматривать данные пользователей, для которых он является менеджером. Для этого в верхней части сайта сделать активной строку поиска. После этого появится список всех людей, за которых человек ответственен. После этого можно искать необходимого сотрудника (рисунок 7).

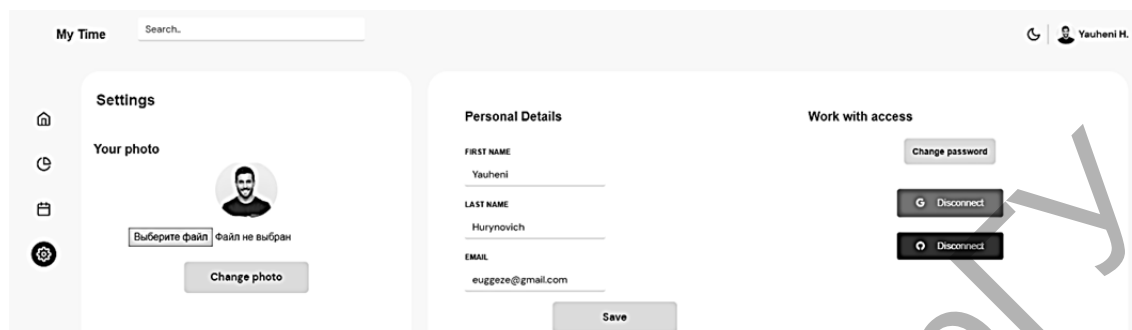


Рисунок 6 — Страница настроек пользователя

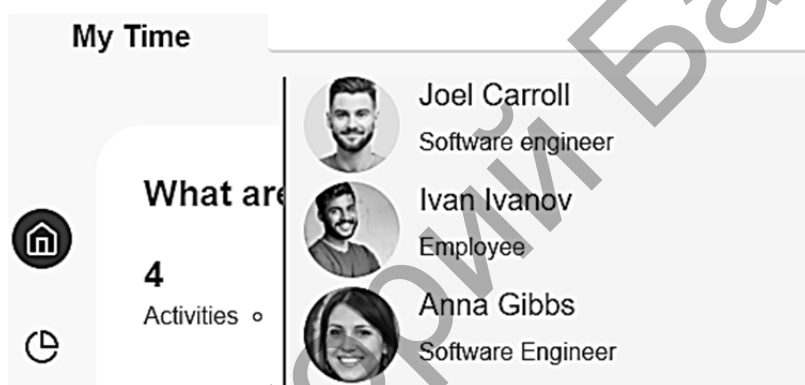


Рисунок 7 — Строка поиска сотрудников

Заключение. В результате исследовательской работы было создано веб-приложение для контроля рабочего времени сотрудников. Приложение было разработано на языке программирования Python с использованием веб-фреймворка Django.

В приложении реализовано:

1. Возможность добавления, редактирования, удаления данных по заданиям, выполняемым сотрудником, данных по запросам отпусков.
2. Хеширование паролей пользователей, подтверждение email-адресов, подключение сервисов GitHub, Google.
3. Попытка перехода пользователем на страницы, которые не являются его. В этом случае пользователь перенаправляется на главную страницу.

Данное приложение во время создания было в процессе непрерывной интеграции на версии Python 3.7, 3.8, 3.9, 3.10. Также было в процессе непрерывного развертывания на веб-сервис Heroku с подключением облачной базы данных PostgreSQL. Было подключено облачное хранилище Dropbox для хранения медиа-файлов для импортирования в проект. Приложение прошло успешное тестирование и готово к использованию.

Список цитируемых источников

1. Что такое облачные технологии? [Электронный ресурс]. — Режим доступа : <https://secretmag.ru/enciklopediya/chto-takoe-oblachnyye-tekhnologii-obyasnyаем-prostymi-slovami.htm> . — Дата доступа : 05.05.2022.
2. Django — фреймворк для веб-разработки в Python [Электронный ресурс]. — Режим доступа : <https://web-creator.ru/articles/django> . — Дата доступа : 05.05.2022.
3. PostgreSQL — объектно-реляционная система управления базами данных [Электронный ресурс]. — Режим доступа : <https://web-creator.ru/articles/postgresql> . — Дата доступа : 05.05.2022.