

**РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ КАФЕДРАЛЬНОГО
ВЕБ-СЕРВЕРА ДЛЯ ДИСТАНЦІЙНОГО ВИКЛАДАННЯ КУРСІВ З
ЕЛЕКТРОПРИВОДУ**

У даній статті розглядається можливість реалізації невеликого веб-сервера, основним завданням якого являється надання доступу до кафедральних навчальних курсів з електроприводу з подальшою можливістю їх накопичення та редагування. Середовищем розробки для веб-додатку був мікрофреймворк Flask для створення веб-додатків на мові програмування Python. Flask класифікується як мікрофреймворк, тому що не вимагає певних інструментів або бібліотек, свідомо надаючи свободу вибору розробнику який функціонал реалізовувати в своєму додатку, підтримуючи розширення, які можуть додавати функції програми, як якщо б вони були реалізовані в самому Flask. Цей функціонал реалізується за допомогою набору розширень, які за бажанням можна встановити у мікрофреймворк таким чином, як якщо б вони були реалізовані в самому Flask. У веб-додатку були реалізовані наступні функції: система реєстрації користувачів, надавання ролі для викладачів, адміністративна сторінка для викладачів та два види файлового менеджера як для звичайних користувачів, так і для викладачів. Система реєстрації та надавання ролі реалізується за допомогою розширення Flask-Security. Адміністративна сторінка та файловий менеджер було створено за допомогою розширення Flask-Admin. Для стилізації додатку використовувався шаблонізатор Jinja2, який дозволяє підставляти змінні з Python прямо у html-сторінці в браузері та є стандартом при написанні веб-додатків на Flask. Також використовувався CSS-фреймворк під назвою Bootstrap, призначений для гнучкої інтерфейсної веб-розробки. В якості сервера використовується мініатюрний одноплатний комп'ютер Orange Pi Zero, який виділяється на фоні інших мікрокомп'ютерів компактними габаритами, низьким енергоспоживанням та привабливою ціною.

Ключові слова: веб-сервер, веб-додаток, мікрофреймворк, Flask, Python, Orange Pi Zero.

I.D. Horbatko, O.E. Hryhorenko. Development of software of the department web-server for remote teaching of courses on the electric drive. This article discusses the possibility of implementing a small web-server, the main task of which is to provide access to the department training courses on electric drive with the subsequent possibility of their accumulation and editing. The development environment for the web-application was the Flask microframework for creating web-applications in the Python programming language. Flask is classified as a microframework, since it does not require either specific tools or libraries, deliberately giving the developer latitude in the choice of the specific set of functions in his app, supporting an extension that can add program functions as if they were implemented in Flask itself. This set of functions is implemented using a set of extensions that, if desired, can be installed in the microframework in such a way as if they were implemented in Flask itself. The web-application implements the following functions: a user registration system, granting a role for teachers, an administrative page for teachers, and two types of file manager for both regular users and teachers. The system for registering and granting a role is implemented through the Flask-Security extension. Both the admin page and file manager have been created using the Flask-Admin extension. To style the application, the Jinja2 templating engine was used, which makes it possible to substitute variables from Python di-

¹ студент, ДВНЗ «Приазовський державний технічний університет», м. Маріуполь

² студент, ДВНЗ «Приазовський державний технічний університет», м. Маріуполь, aeg.sasha15@gmail.com

rectly in the html page in the browser and is a standard when making up web-applications in Flask. CSS framework called Bootstrap designed for flexible front-end web-development has been used as well. A miniature single-board computer Orange Pi Zero is used as a server, which stands out against other microcomputers with its compact dimensions, ultra-low power consumption and an attractive price.

Keywords: web-server, web-app, microframework, Flask, Python, Orange Pi Zero.

Постановка проблеми. Сучасна ситуація, пов'язана з пандемією, змушує повністю переходити на дистанційні методи викладання навчальних матеріалів, що в свою чергу підвищує попит на сервіси, які вирішують цю проблему. Проте, нинішні сервіси, що надаються університетом, не були розраховані на подібне збільшення користувачів та не відрізняються стабільною і надійною роботою. Рішенням даної проблеми може стати розробка власного веб-додатку, розрахованого на малий круг користувачів, так як його робота поширюється тільки на конкретну кафедру. Це дозволить спростити процес програмної розробки, а також сильно знизить витрати на придбання та обслуговування серверного обладнання, так як для роботи подібного веб-додатку буде досить використання мініатюрного одноплатного комп'ютера.

Аналіз останніх досліджень і публікацій. На сьогоднішній день найпопулярнішими веб-фреймворками для розробки веб-додатків на мові Python є Django та Flask. Причому, Flask обігнав Django і посів перше місце, враховуючи, що індустрія веб-розробки протягом останніх п'яти років мала тенденцію до мікрофреймворків та мікросервісів [1]. Flask – це хороший вибір, коли треба швидко створити сайти малого або середнього розміру, такі як форуми та особисті блоги, в той час, як Django ідеально підходить для великих проектів, таких як сайти електронної комерції та CMS (Content management system – система управління контентом). Flask мінімалістичний і не має обмежень, що означає, що розробники можуть реалізувати саме те, що вони хочуть, за допомогою зовнішніх бібліотек, що робить Flask гнучким та розширюваним [2].

На основі перерахованого вище, оптимальним вибором інструментарію для розробки веб-сервера був обраний саме Flask, так як в необхідному додатку потрібен невеликий функціонал, який виконує лише основні задачі, для меншої ресурсоемності та простоти його реалізації.

Мета роботи – розробка веб-сервера, який надає дистанційний доступ до кафедральних навчальних курсів з електроприводу.

Виклад основного матеріалу. Поняття «веб-сервер» може ставитися як до апаратної начинки, так і до програмного забезпечення. Або навіть до обох частин, які працюють спільно.

З точки зору «заліза», «веб-сервер» – це комп'ютер, який зберігає файли сайту (HTML-документи, CSS-стили, JavaScript-файли, картинки та інше) та доставляє їх на пристрій кінцевого користувача (веб-браузер і т. д.). Він підключений до мережі Інтернет і може бути доступний через доменне ім'я.

З точки зору програмного забезпечення, веб-сервер включає в себе кілька компонентів, які контролюють доступ веб-користувачів до розміщених на сервері файлів, як мінімум – це HTTP-сервер. HTTP-сервер – це частина програмного забезпечення (ПЗ), яка розуміє URL (веб-адреси) та HTTP (протокол, який браузер використовує для перегляду веб-сторінок).

На самому базовому рівні, коли браузеру потрібен файл, розміщений на веб-сервері, браузер запитує його через HTTP-протокол. Коли запит досягає потрібного веб-сервера («заліза»), сервер HTTP (ПЗ) приймає запит, знаходить запитуваний документ (якщо немає, то повідомляє про помилку 404) і відправляє назад, також через HTTP.

Щоб опублікувати веб-сайт, необхідний або статичний, або динамічний веб-сервер.

Статичний веб-сервер складається з комп'ютера («залізо») з сервером HTTP (ПЗ). Він називається «статикою», тому що сервер посилає розміщені файли в браузер «як є».

Динамічний веб-сервер складається з статичного веб-сервера і додаткового програмного забезпечення, найчастіше сервера додатку та бази даних. Він називається «динамічним», тому що сервер додатків змінює вихідні файли перед відправкою в браузер по HTTP [3].

У даному випадку був розроблений веб-сервер, де в якості «заліза» застосовується мікрокомп'ютер Orange Pi, а в якості ПЗ – веб-додаток, написаний за допомогою мікрофреймворка Flask на мові програмування Python.

Flask класифікується як мікрофреймворк, тому що не вимагає певних інструментів або бібліотек, свідомо надаючи лише базові функції. Іншими словами, Flask своєю простотою надає

розробнику вибір який функціонал використовувати для вашої програми, підтримуючи розширення, які можуть додавати функції програми, як якщо б вони були реалізовані в самому Flask. Існують розширення для перевірки форм, обробки завантаження, різних технологій відкритої аутентифікації і декількох загальних інструментів [4].

Для розробки веб-сервера застосовувалися такі розширення, як Flask-Security та Flask-Admin. Flask-Security дозволяє швидко додавати загальні механізми безпеки у Flask, а саме: сесійна аутентифікація; реєстрація користувача; доступ на основі ролей; хешування пароля та його перевірка на складність; базова HTTP-аутентифікація; підтвердження по електронній пошті; скидання та відновлення пароля.

Аутентифікація на основі сеансу повністю виконується розширенням Flask-Login та Flask-WTF. Flask-Security обробляє конфігурацію Flask-Login автоматично на основі декількох власних значень конфігурації і використовує функцію альтернативного токена Flask-Login для запам'ятовування користувачів, коли їх сеанс закінчився. Flask-WTF інтегрується з сеансом, а також відразу забезпечує підтримку CSRF (англ. Cross Site Request Forgery – «міжсайтова підробка запиту»).

Flask-Security реалізує дуже просте управління ролями, яке дозволяє зв'язати роль високого рівня або кілька ролей з будь-яким користувачем. Наприклад, можна призначити користувачу такі ролі, як адміністратор, редактор, привілейований користувач або їх комбінацію. Контроль доступу заснований на імені ролі та/або дозволах, що містяться в ролі. Всі ролі повинні мати однозначні імена. Ця функція реалізована за допомогою розширення Flask-Principal.

Хешування паролів включено за допомогою бібліотеки passlib. За замовчуванням паролі хешуються за допомогою функції bcrypt, але можна легко налаштувати алгоритм хешування.

При бажанні, можна включити, щоб нові користувачі підтвердили свою адресу електронної пошти. Flask-Security відправить всім новим користувачам повідомлення електронної пошти з посиланням для підтвердження (рис. 1-3). Після переходу за посиланням підтвердження, користувач автоматично увійде в систему. Також є можливість повторної відправки посилання підтвердження на дану електронну пошту, якщо користувач спробує використовувати прострочений токен або втратив попередній лист. Посилання підтвердження можна налаштувати так, щоб термін дії закінчувався після закінчення певного часу.

doc.in

Создать аккаунт

Спасибо. Инструкция по подтверждению аккаунта отправлена на evvibnvinkfybkbfr@niwghx.online.

Пожалуйста, войдите чтобы получить доступ к этой странице.

Войдите

Почтовый адрес

Пароль

Запомнить меня

Войти

[Забыли пароль?](#)

[Еще нет аккаунта? Зарегистрируйтесь](#)

Рис. 1 – Демонстрація підтвердження електронної адреси при реєстрації нового користувача

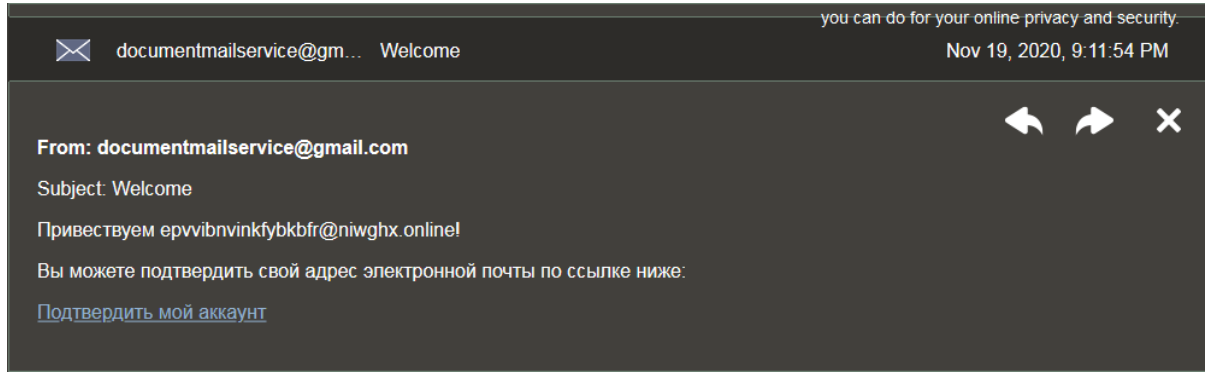


Рис. 2 – Демонстрація отримання листу для активації аккаунту нового користувача

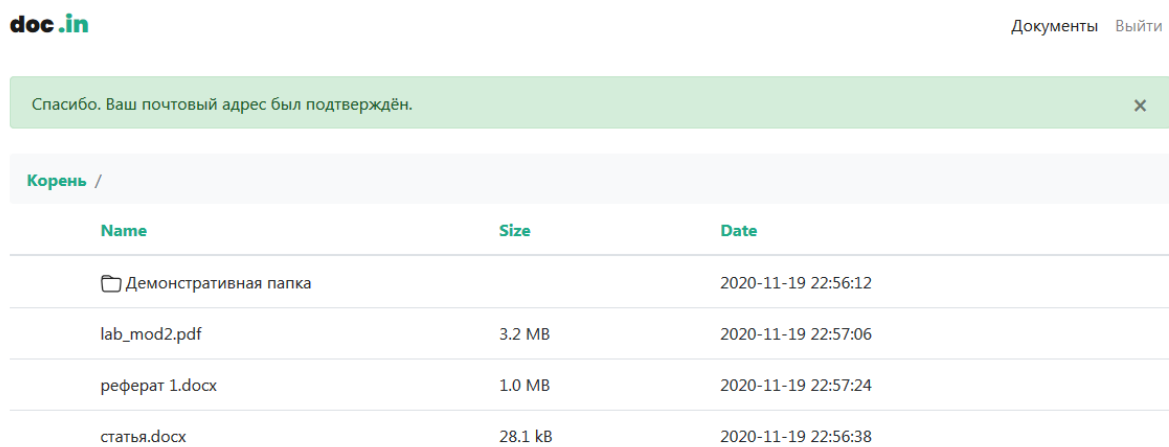


Рис. 3 – Демонстрація підтвердження електронної адреси при реєстрації нового користувача

Flask-Security відправляє користувачу електронного листа з посиланням, в якому він може скинути свій пароль (рис. 4). Після скидання пароля відбувається автоматичний вхід в систему, і з цього моменту буде використовуватися новий пароль. Посилання для скидання пароля можуть бути налаштовані таким чином, щоб термін їх дії закінчувався по заданому часу.

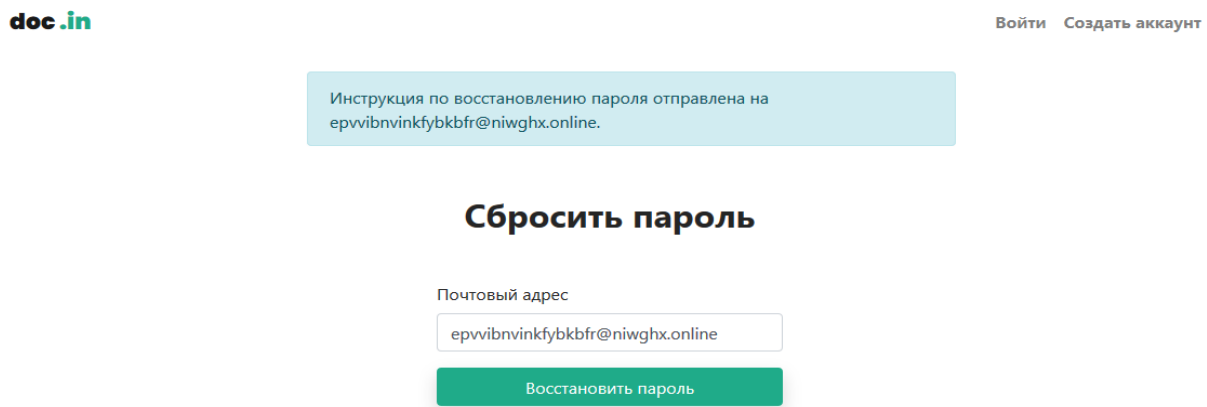


Рис. 4 – Демонстрація скидання пароля користувача

Flask-Security поставляється з базовим представленням реєстрації користувача. Новим користувачам потрібно лише вказати адресу електронної пошти та пароль. Це представлення можна змінити, якщо в процесі реєстрації потрібно більше полів [5].

В свою чергу Flask-Admin вирішує проблему побудови інтерфейсу адміністратора поверх існуючої моделі даних. З невеликими зусиллями воно дозволяє керувати даними веб-служби через зручний інтерфейс [6]. З його допомогою були реалізовані дві версії файлового менеджера. Одна – для викладачів (рис. 5), яка дозволяє завантажувати, редагувати файли та папки, і друга – для звичайних користувачів (рис. 6), яка дозволяє лише перегляд та завантаження файлів.

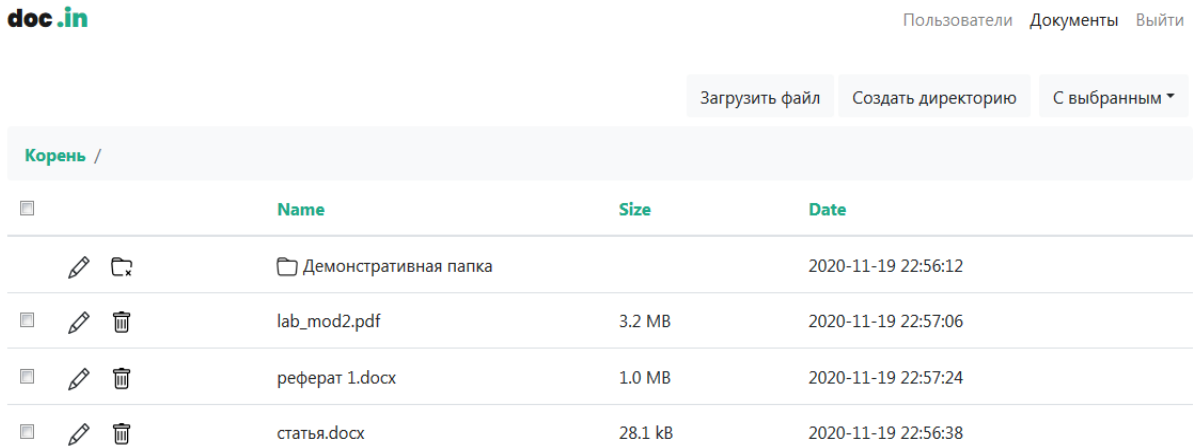


Рис. 5 – Демонстрація версії файлового менеджера з повним функціоналом для викладачів

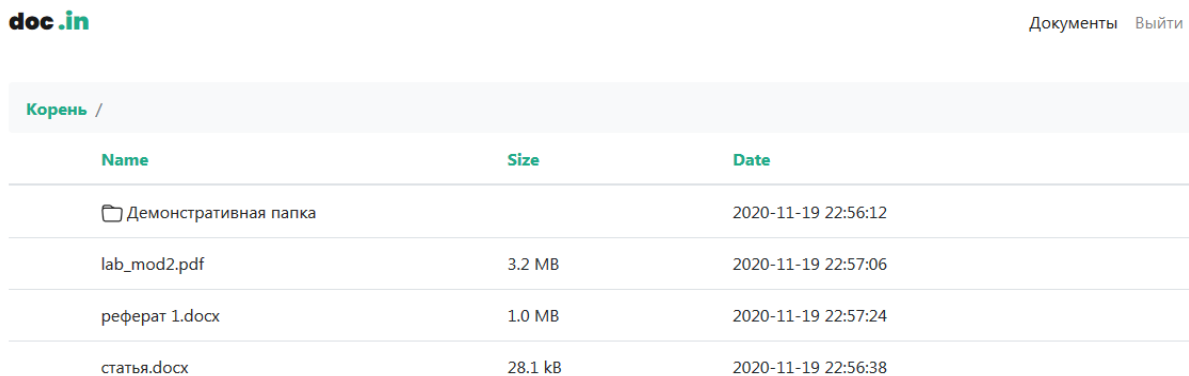


Рис. 6 – Демонстрація версії файлового менеджера з обмеженим функціоналом для звичайних користувачів

Додаток веб-сервера має володіти логікою розуміння нинішнього статусу користувача для генерації йому відповідних посилань та сторінок. Наприклад, якщо користувач не авторизован, йому будуть доступні лише сторінки логіна та реєстрації, в свою чергу при авторизації користувача його переадресують на сторінку файлового менеджера, де в свою чергу відбувається перевірка наявності у нього ролі, яка вирішує яку сторінку додатка відображати. З боку сервера ця логіка реалізується за допомогою вищезгаданого доповнення Flask-Security, а html-шаблони, які генеруються за цією логікою, створюються за допомогою шаблонізатора Jinja2.

Шаблонізатор Jinja2 дозволяє підставляти змінні з Python прямо у html-сторінці в браузері та є стандартом при написанні веб-додатків на Flask. Jinja2 позиціонує себе як інструмент для дизайнерів та верстальників, що спрощує верстку і відокремлює її від розробки, також намагається в міру можливостей ізолювати не-розробників від Python [7].

Для стилізації додатку використовувався CSS-фреймворк під назвою Bootstrap з відкритим вихідним кодом, призначений для гнучкої інтерфейсної веб-розробки, орієнтованої на адаптивний дизайн. Він містить шаблони дизайну на основі CSS і на основі JavaScript для типографіки, форм, кнопок, навігації та інших компонентів інтерфейсу [8].

В якості сервера використовується мініатюрний одноплатний комп'ютер Orange Pi Zero. Це відповідь на Raspberry Pi Zero, який і був найменшим та дешевим комп'ютером. Orange Pi Zero коштує всього 7 доларів США на AliExpress, що трохи дорожче за Raspberry Pi Zero, але він більш просунутий в технічному плані.

Виріб має розміри всього 48×46 мм і важить 26 грамів. Побудований на 4-ядерному процесорі Allwinner H2 з архітектурою ARM (Advanced RISC Machine – вдосконалена RISC-машина) і графічним прискорювачем Mali 400MP2. Міні-комп'ютер може нести на борту 256 або 512 Мбайт оперативної пам'яті DDR3 SDRAM. Для зберігання даних служить змінна карта microSD (рис. 7).

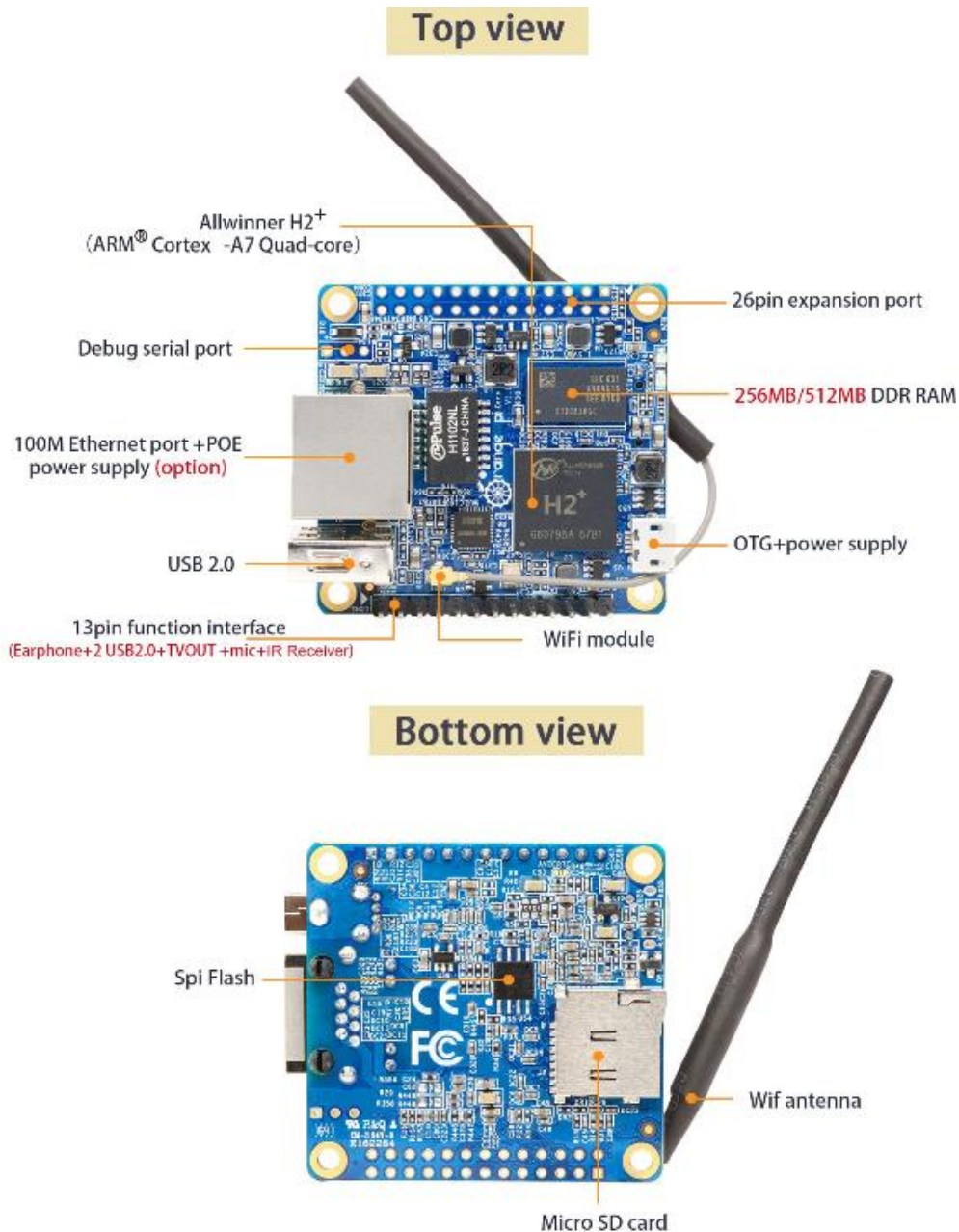


Рис. 7 – Зображення плати Orange Pi Zero

На Orange Pi Zero є адаптер Wi-Fi 802.11b/g/n із зовнішньою антеною і мережевий контролер Ethernet 10/100 з відповідним роз'ємом для підключення кабелю. На платі також є порт USB 2.0 і microUSB 2.0, через який подається живлення, але можна жити й через GPIO-піни та PoE. Є GPIO (General-purpose input/output, Інтерфейс введення/виведення загального призна-

чення) на 26 пінов і додатковий на 13 пінов. 13-піновий інтерфейс дозволяє підключати адаптер для монітора, перехідник на два додаткових USB-порту, мікрофон та навушники. Для полегшення роботи з 13-піновим інтерфейсом є плата розширення, яку можна купити на AliExpress за ціною 2 долари США [9].

Orange Pi Zero досить-таки невимогливий до живлення – сам по собі мікрокомп'ютер споживає приблизно 300 мА, тобто для роботи буде достатньо блоку живлення з вихідним струмом до 1 А.

Цей мікрокомп'ютер має компактні габарити, наднизьке енергоспоживання і здатний виконувати ті завдання, для реалізації яких буде мало платформи Arduino або ESP8266, що робить його дуже привабливим варіантом в якості серверного обладнання для невеликих веб-додатків.

Висновки

1. Запропоновано використовувати для невеликих веб-додатків мікрофреймворк Flask з опціональною можливістю використання розширень.

2. Для швидкої та адаптивної стилізації дизайну додатку запропоновано використовувати Bootstrap.

3. В якості серверного обладнання запропоновано використовувати компактний мікрокомп'ютер Orange Pi Zero.

Перелік використаних джерел:

1. Python Developers Survey 2019 Results [Electronic resource] : [Website]. – Electronic data. – Mode of access : <https://jetbrains.com/lp/python-developers-survey-2019>.
2. Flask vs Django: How to Understand Whether You Need a Hammer or a Toolbox [Electronic resource] : [Website]. – Electronic data. – Mode of access : <https://medium.com/@SteelKiwiDev/flask-vs-django-how-to-understand-whether-you-need-a-hammer-or-a-toolbox-39b8b3a2e4a5>.
3. What is a web server? [Electronic resource] : [Website]. – Electronic data. – Mode of access : https://developer.mozilla.org/en-US/docs/Learn/Common_questions/What_is_a_web_server.
4. Welcome to Flask – Flask Documentation [Electronic resource] : [Website]. – Electronic data. – Mode of access : <https://flask.palletsprojects.com/en/1.1.x>.
5. Flask-Security [Electronic resource] : [Website]. – Electronic data. – Mode of access : <https://pythonhosted.org/Flask-Security>.
6. Flask-Admin – flask-admin 1.5.3 documentation [Electronic resource] : [Website]. – Electronic data. – Mode of access : <https://flask-admin.readthedocs.io/en/latest/>.
7. Jinja – Jinja Documentation (2.11.x) [Electronic resource] : [Website]. – Electronic data. – Mode of access : <https://jinja.palletsprojects.com/en/2.11.x>.
8. Bootstrap 4 – Introduction [Electronic resource] : [Website]. – Electronic data. – Mode of access : <https://getbootstrap.com/docs/4.0/getting-started/introduction>.
9. Orange pi zero – Orangepi [Electronic resource] : [Website]. – Electronic data. – Mode of access : <http://www.orangepi.org/orangepizero>.

References:

1. Python Developers Survey 2019 Results Available at : www.jetbrains.com/lp/python-developers-survey-2019 (accessed 13 August 2020).
2. Flask vs Django: How to Understand Whether You Need a Hammer or a Toolbox Available at : www.medium.com/@SteelKiwiDev/flask-vs-django-how-to-understand-whether-you-need-a-hammer-or-a-toolbox-39b8b3a2e4a5 (accessed 20 August 2020).
3. What is a web server? Available at : www.developer.mozilla.org/en-US/docs/Learn/Common_questions/What_is_a_web_server.
4. Welcome to Flask – Flask Documentation Available at : www.flask.palletsprojects.com/en/1.1.x (accessed 13 August 2020).
5. Flask-Security Available at : www.pythonhosted.org/Flask-Security (accessed 02 September 2020).
6. Flask-Admin – flask-admin 1.5.3 documentation Available at : www.flask-admin.readthedocs.io/en/latest (accessed 02 September 2020).

7. Jinja – Jinja Documentation (2.11.x) Available at : www.jinja.palletsprojects.com/en/2.11.x (accessed 10 September 2020).
8. Bootstrap 4 – Introduction Available at : www.getbootstrap.com/docs/4.0/getting-started/introduction (accessed 11 September 2020).
9. Orange pi zero – Orangepi Available at : www.orangepi.org/orangepizero (accessed 12 September 2020).

Рецензент: Д.С. Міроненко
канд. техн. наук, доц., ДВНЗ «ПДТУ»

Стаття надійшла 30.09.2020