УДК 004.9
Крашівський А.І. магістрант
(Тернопільський національний технічний університет імені Івана Пулюя)

РОЗРОБКА ВЕБ-СИСТЕМИ З ВИКОРИСТАННЯМ NODE.JS ТА MONGODB НА ПРИКЛАДІ СИСТЕМИ АВТОМАТИЗАЦІЇ HR-ПРОЦЕСІВ

UDC 004.9
Krashivskyi A.

WEB SYSTEM DEVELOPMENT USING NODE.JS AND MONGODB ON EXAMPLES OF HR-PROCESS AUTOMATION SYSTEM

I. Постановка проблеми

Дана робота присвячена процесу проектування, розробки та розгортання на віддаленому сервері веб-додатку для автоматизації HR-процесів з використанням фреймворку Angular.js, платформи Node.js і системою управління базами даних MongoDB. Цей додаток повинен зберігати інформацію про людські ресурси, освіту всіх співробітників, їх останній досвід, відвідуваність, управління відпустками та поточне управління проектами. В рамках даної системи повинна бути реалізована адміністративна панель для управління всією інформацією про працівників. Система міститиме статистичні дані про ставку найманих працівників з різних установ та середню плинність працівників з різних установ. Особлива увага приділена огляду принципів роботи, основних можливостей і переваг використаних інструментів і технологій.

II. Мета роботи

Метою дослідження є розробка інформаційної системи автоматизації HR-процесів на базі Node.js та MongoDB.

III. Вибір програмних інструментів для реалізації системи

Для вирішення поставлених завдань було вирішено створити SPA (Single Page Application – вебсайт або веб-сайт, що використовує єдиний HTML-документ як оболочку для всіх веб-сайтів та організує взаємодію з користувачем за допомогою динамічно завантаженого HTML,CSS, JavaScript без перезагрузки всієї сторінки) [1]. Для створення сучасних SPA на стороні клієнта існує багато фреймворків, одним з яких є Angular.js, ми його використовуємо для клієнтської частини нашого додатку, надає більші можливості для реалізації складних веб-додатків. Основними можливостями є зв’язування даних моделейта презентації, розбиття додатків на модулі, підтримка шаблонів, функції для зручної роботи з promise, http-запитами.

Клієнт додатку на етапі розробки представляє сукупність розділених CSS, HTML, JSфайлів, щоб мати можливість використовувати їх усі разом необхідна система збору фронтенда, ми будемо використовувати Webpack, за допомогою його ми зможемо зібрати всі js-файли в одному бандлі,використовуємо css препроцесори, такі як Sass, і так вже у нас з’являється можливість для модульного розбиття html і підключення його в js функцією require().

Для написання серверної логіки використовується Node.js – програмна платформу,заснована на движку V8 (транслятор JavaScript в машинний код). Node.js надає можливість JavaScript взаємодіяти з пристроями вводу-виводу через свій API (написаний на C++), підключити інші зовнішні бібліотеки, написані на різних мовах, забезпечують виклики до них із JavaScript-коду [2]. Також Node.js розміщується разом із пакетним менеджером NPM за допомогою якого ви можетешвидко отримати доступ до модулів під різні потреби. Використовуючи NPM можназручно керувати всіма зовнішніми залежностями додатків, надаючи можливість автоматичного завантаження їх у каталог node_modules у папку проекту за допомогою команд npm install.

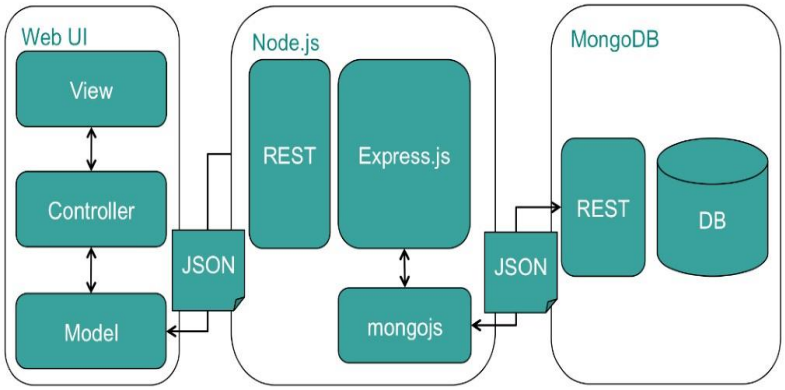
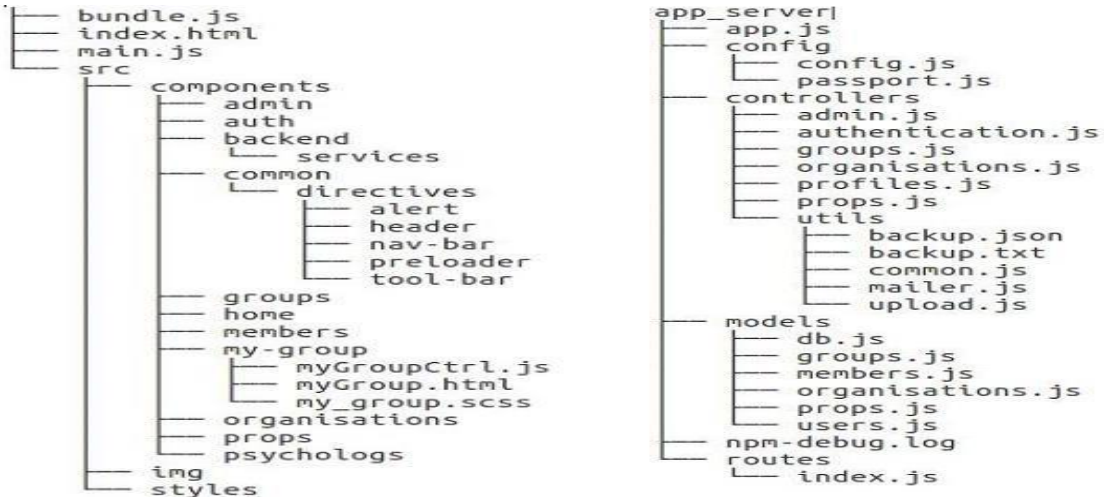


Рисунок 1. Архітектура системи

Серед переваг Node.js можна виділити асинхронність, неблокуючий ввід-вивід, можливість написання ізоморфних додатків, використовуючи один код на клієнтах і на серверах. Також нам потрібна база даних, вибір здійснено на користь MongoDB – документована база даних NoSQL даних, яка зберігає інформацію в документах JSON. У порівняні з реляційними базами дані документованих БД мають перевагу в швидкості роботи та менше займають пам'яті. Для аутентифікації користувачів буде використовуватися концепція JWT (JSON Web Token). Для спілкування між клієнтами та сервером ми будемо використовувати REST - це стиль архітектури програмного забезпечення, при якому сервер не зберігає стан клієнта, а також кожен запит від клієнта до сервера містить у собі вичерпну інформацію про бажаний доступ до сервера. Дії над даними завданнями виконуються за допомогою методів: GET (отримати), PUT (додати, замінити),POST (додати, змінити, видалити), DELETE (видалити). Таким чином, дії CRUD (Create-Read-Updtae-Delete) можуть виконуватися як з усіма 4-ма методами, так і лише за допомогою GET і POST. Використання методології REST забезпечує переваги за рахунок простоти, масштабності, надійності та продуктивності сервісів.

IV. Реалізація системи

Для підтримки масштабованості, зручності внесення змін і збереження чіткої логічної структури всіх модулів клієнтської частини нашого додатку дуже важливо правильно вибрати метод розбиття файлів на директорії і піддиректорії (див. рис. 2).



а) б)

Рисунок 2. Структура каталогу файлів клієнтської частини (а) та серверної частини (б) системи

Для того щоб розгорнути нашу систему в інтернеті було вирішено скористатися безкоштовними функціями хмарної інтернет-платформи Heroku, яка крім Node.js підтримує також мови Java, Ruby, Scala, PHP та інші.

Програми, що працюють на Heroku, використовують також DNS-сервер Heroku(зазвичай додатки мають доменне ім’я виду «імя_додатку.herokuapp.com»). Для кожного додатку виділяється кілька незалежних віртуальних процесів, які називаються «dynos».Вони розподілені по спеціальній віртуальній сітці («dynos grid»), яка складається з декількох серверів. Крім цього є зручна можливість налаштувати автоскладання додатку з git-репозиторію [2]. В результаті розроблена система була успішно розміщена і тепер доступна по домену. Система планується до практичного використання в одній з IT компаній міста Тернопіль для управління HR процесами.

Висновок

В рамках проведеного дослідження реалізована веб-система управління HR-процесами з використанням сучасних інформаційних технологій. Здійснено апробацію системи на прикладі автоматизації основних HR бізнес процесів водній з провідних компаній Тернополя.

Література.

1. Міковскі Майкл, Пауелл Джош. Розробка односторінкових веб-додатків // ДМК Пресс,2014. 512 с.
2. Lambert M. Surhone, Mariam T. Tennoe, and Susan F. Henssonow. 2010. Node.js. Betascript Publishing, Beau Bassin, MUS.