

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ВАСИЛЯ СТУСА

НАГОЛЮК ДМИТРО ОЛЕКСАНДРОВИЧ

Допускається до захисту:

завідувач кафедри

інформаційних технологій,

д-р техн. наук, доцент

_____ Тетяна НЕСКОРОДСЬКА

«_____» _____ 2022р.

РОЗРОБКА CRM СИСТЕМИ ДЛЯ БІЗНЕСУ – BUSINESS STAR

Спеціальність 122 «Комп'ютерні науки»

Кваліфікаційна (бакалаврська) робота

Керівник:

Павло РИМАР, старший викладач

кафедри інформаційних технологій

Оцінка: _____ / _____ / _____

(бали за шкалою ЄКТС / за національною шкалою)

Голова ЕК: _____

(підпис)

Вінниця – 2022

АНОТАЦІЯ

Наголюк Д.О. Розробка CRM системи для бізнесу – Business Star.

Спеціальність 122 «Комп'ютерні науки», освітня програма «Сучасні інформаційні технології та програмування». Донецький національний університет імені Василя Стуса, Вінниця, 2022.

Кваліфікаційна (бакалаврська) робота присвячена розробці CRM системи, яка надає змогу всім приватним підприємцям, яким потрібно організувати роботу колективу, приймати запити на запис до співробітників за допомогою сайту візитки з інтегрованим календарем. Користувач має змогу переглядати календар своїх співробітників та вносити туди зміни. Користувач має змогу зареєструватись як підприємець або як працівник, у працівника – обмежені права доступу до функціоналу.

Ключові слова: CRM система, JavaScript, календар, сайт-візитка
50 сторінок, 50 рис., 30 джерел.

ABSTRACT

Naholiuk D.O Development of CRM system for business - Business Star.

Specialty 122 "Computer Science", educational program "Modern Information Technology and Programming". Vasyl Stus Donetsk National University, Vinnytsia, 2022.

Qualifying (bachelor's) work is dedicated to the development of CRM system, which allows all private entrepreneurs who need to organize the work of the team, to accept requests for registration to employees through the site of a business card with an integrated calendar. The user has the opportunity to view the calendar of their employees and make changes there. The user has the opportunity to register as an entrepreneur or as an employee and depending on this will be provided with certain functionality that can be used.

Keywords: CRM system, JavaScript, calendar, business card site 50 Pages 50
fig 30 sources.

ЗМІСТ

ВСТУП	4
РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ. ОПИС АНАЛОГІВ	7
1.1 Постановка задачі	7
1.2 Аналоги	9
Висновок до розділу 1	10
РОЗДІЛ 2. БІБЛІОТЕКИ ТА ТЕХНОЛОГІЇ	12
2.1 React.....	12
2.2 Redux	13
2.3 React-router-dom	14
2.4 React-redux	15
2.5 Ant design	15
2.6 Moment	15
2.7 Redux thunk	16
2.8 AXIOS та JSON-server	16
2.9 Visual Studio Code	17
Висновок до розділу 2	19
РОЗДІЛ 3. ПРОЦЕС РОЗРОБКИ CRM СИСТЕМИ	20
3.1 Процес реєстрації.....	20
3.2 Основний інтерфейс	31
3.3 Календар та події.....	32
3.4 Сайт візитка	38
3.5 Список співробітників	39
3.6 Авторизація.....	40
3.7 Авторизація як співробітник без прав доступу	41
Висновок до розділу 3	43
ВИСНОВКИ.....	44
СПИСОК ЛІТЕРАТУРИ.....	46

ВСТУП

Підприємці – це двигуни економіки. Як в Україні так і в цілому світі. Частка малого та середнього бізнесу в ВВП України складає приблизно 55% [25]. Для зручного управління бізнес процесами в підприємстві існують CRM-системи.

CRM-система - це програмне забезпечення, яке дозволяє автоматизувати та полегшити бізнес-процеси. Сама аббревіатура CRM - скорочення від Customer Relationship Management, що в перекладі управління взаємовідносинами з клієнтами [1].

Для сучасного підприємця вкрай важливо бути представленим в інтернеті та мати зручний інструмент для керування персоналом, але власний сайт або програма – це дуже складно та дорого [21]. В підприємців і так вистачає клопоту. В них немає часу на вивчення необхідних для цього технологій. Тут на допомогу приходять CRM-системи, які пропонують бізнесу наступне:

1. Реєстрацію компанії в CRM-системі.
2. Календар, в якому є можливість призначати записи до співробітника, редагувати, переглядати, видаляти їх.
3. Сайт шаблон, який можна заповнити своїм текстом та пропонують певний функціонал, такий як онлайн запис. Як правило, за допомогою HTML та CSS-коду шаблони сайтів дозволяють будь-якому налаштувати веб-сторінку без необхідності наймати професійного веб-розробника чи дизайнера [23].
4. Керування персоналом, де можна додати співробітника до компанії, видалити його або змінити дані. Керування персоналом включає в себе – задання графіку роботи, завдань та інше [24].
5. Керування послугами, де так само можна прописати всі види послуг, які надає компанія, вказати орієнтовну тривалість послуги та вартість. Потім список цих послуг буде доступний користувачу на сайті візитці, де він зможе зробити на запис на ту чи іншу послугу, а час завершення запису

автоматично корегується в залежності від заданої завчасно тривалості послуги [22].

Найкраще це продемонструвати на прикладі. Є невелика перукарня, власник якої хоче почати використовувати CRM-систему. Для цього він реєструється в CRM системі, заповнює дані про свою компанію, додає співробітників, призначаючи їм унікальні імена та паролі, потім вводить назву піддомену, на якому буде створено унікальний сайт візитка.

Піддомен – це домен 3-го рівня або вище. Вказується він перед основним доменом [29].

Домен - це адреса сайту в мережі, представлена у вигляді букв і цифр. Він перетворює складно запам'ятовується IP-адреса в звичний користувачеві вид. Кожне найменування унікально - не можна зареєструвати ідентичні послідовності символів. Максимальний розмір комбінації - 63 знака. У найменуванні можна використовувати літери, цифри, а також тире [30].

Після завершення реєстрації власник потрапляє на головну сторінку CRM-системи, де відображається його особистий календар, а в меню посилення на календарі його співробітників. В календарі він має можливість призначити запис на ту чи іншу послугу. При тому будуть враховуватися перерви, робочий час та інші записи, щоб не допустити накладання записів одне на одного.

Таким чином відбувається автоматизація процесу ведення бізнесу. Не потрібно вести паперові журнали обліку та витрачати купу часу на це.

Автоматизація бізнес-процесів — раціональне використання виробничих фондів шляхом застосування новітнього програмного забезпечення. Автоматизаційні рішення завжди скорочують час обробки документації і в цілому підвищують профіт роботи підприємства [28].

Мета бакалаврської роботи: використовуючи стек технологій для розробки веб-додатків, розробити CRM-систему для управління процесами підприємства. Вивчити методику проектування front-end[2] частини CRM-системи на мові програмування JavaScript[3], використовуючи React[4] та ознайомитися з методикою розробки сучасних веб-додатків за допомогою JSON-

server[5] для взаємодії з REST API[6], що дозволяє вести розробку без повноцінної Back-end [2] частини.

Задачами дослідження є:

1. Огляд існуючих CRM систем.
2. Вивчення необхідного стеку технологій для веб-розробки.
3. Розробка CRM системи, яка допоможе підприємцям зручніше керувати персоналом.

Об'єкт дослідження: CRM система для підприємців.

Предмет дослідження: процес розробки CRM-системи.

Структура роботи: кваліфікаційна робота складається зі вступу, трьох розділів, висновків, списку використаних джерел та додатків.

Робота містить 50 сторінок, 50 рисунків та список літератури з 30 джерел.

РОЗДІЛ 1

ПОСТАНОВКА ЗАДАЧІ. ОПИС АНАЛОГІВ

В даному розділі наведена постановка задачі дослідження. Проведена порівняльна характеристика з існуючими аналогами.

1.1 Постановка задачі

За допомогою бібліотеки React розробити front-end частину CRM-системи, а за допомогою JSON-server back-end та базу даних.

В програмі має бути реалізований функціонал реєстрації користувача: для цього потрібно буде реалізувати форму для вводу імені, паролю та підтвердження паролю. В JSON-server для цього потрібно буде прописати надбудову, яка буде перехоплювати відповідний запит та додавати користувача в базу даних, але перед тим перевіряти чи існує користувач з таким іменем, у випадку, коли існує – показувати відповідне сповіщення.

Після успішної реєстрації потрібно реалізувати функціонал заповнення даних про компанію, який повинен складатися з 3 кроків:

1. Заповнення назви, адреси та опису компанії.
2. Додавання співробітників в компанію.
3. Вибір шаблону сайту візитки.

Після успішного заповнення даних користувач має потрапляти на головний екран програми. Головний екран програми має складатися з шапки сайту, календарю, меню.

Шапка сайту. Має знаходитися вгорі і містити такі елементи: праворуч напис «Business Star» - назва програми, поруч ім'я авторизованого користувача, а в правій частині кнопка для виходу з акаунту користувача.

Меню. Меню повинно складатися з декількох підгруп. Календарі – де мають бути посилання на всі календарі співробітників. Списки – де мають бути посилання на список співробітників та список подій. Сайт – у цьому розділі повинно бути посилання на сайт візитку компанії.

На сторінці списку подій користувач повинен мати змогу формувати список подій, які потім можна буде вказувати в календарі.

На сторінці списку працівників користувач повинен мати змогу додавати та видаляти співробітників з компанії.

Календар. В ньому має бути реалізований функціонал відображення подій на місяць, можливість перемикається по датам для відображення подій на інші місяці. Можливість додавання події в календар. Форма для її додавання має включати в себе – назву, працівник, який її виконує, дата початку та дату закінчення, також час завершення події має вираховуватися сам в залежності від часу початку та тривалості обраної події. Також має бути відображення всіх подій у розгорнутому вигляді.

Сайт-візитка – сайт, містить інформацію про підприємство та список його послуг [26]. Сайт візитка має складатися з шапки, форми для запису, загальної інформації. В шапці має бути відображення назви та адреси компанії. Форма для запису – за допомогою неї потенційний клієнт повинен мати змогу записатися до одного із співробітників компанії, потім запис має з'явитися в календарі співробітника. Загальна інформація про компанію має включати в себе список співробітників, подій та опис компанії.

Має бути реалізований процес авторизації – входу до акаунту. Авторизація - це механізм захисту, який використовується для визначення привілеїв користувача / клієнта або рівнів доступу, пов'язаних із системними ресурсами, включаючи комп'ютерні програми, файли, послуги, дані та функції додатків [27]. При чому потрібно ще розробити систему прав доступу, щоб додані працівники могли лише за згоди керівника мати доступ до всіх функцій CRM-системи.

Дана CRM-система має відрізнятись від аналогів своєю універсальністю: кожен має знайти змогу як її застосовувати – від керівника невеликої перукарні до автосервісу.

1.2 Аналоги

На сьогоднішній день існує достатня кількість CRM систем для ведення бізнесу. Розглянемо переваги та недоліки наступних систем.

Система «AmoCRM». Система має змогу показувати кількість продажів, що дозволяє легко оцінювати та контролювати співробітників. Чудово підходить для малого та середнього бізнесу. Є функціонал запису дзвінків, картки клієнтів та мобільний додаток, має чат для співробітників [20].

Плюси:

- Приємний інтерфейс.
- Мобільний додаток.
- Історія переписок та договорів.
- Статистика.
- Чат.
- Можливість програмного масштабування через відкрите API.

Мінуси:

- Функціонал заточений під продажі.
- Не підходить для специфічних ніш.

«Zoho» – CRM. Її важлива особливість, що в ній присутні елементи гри, що допомагає організувати свого роду змагання між менеджерами, мотивуючи їх підвищувати свою продуктивність. Також є конструктор форм, який дозволяє зробити свою унікальну програму [20].

Плюси:

- Ціна підписки.
- Система мотивації.
- Конструктор програм.
- Гнучкість системи.

Мінуси:

- Багато часу витрачається на налаштування.

«aveCRM». Призначена для інтеграції з сайтом. Дозволяє ефективно керувати угодами. Використовує діаграми Ганта для наочного відображення. Підтримує шаблони [20].

Плюси:

- Шаблони.
- Візуалізація.

Мінуси:

- Немає хмарної версії.
- Не можна додавати користувача як гостя.

«OneBox». Система з можливістю масштабування. Пропонує поштові клієнти, сервіси телефонії та інший софт. Є можливість програмного розширення функціоналу.

Її відмінність від інших CRM систем – вона використовує нейронні мережі для навчання. Аналізує бізнес процеси та надає свої пропозиції [20].

Плюси:

- Простота управління.
- Нейронні мережі.
- Масштабованість.
- Інтегровані сервіси.

Мінуси:

- Велика вартість.
- Помилки в роботі.

Висновок до розділу 1

В даному розділі була сформульована постановка задачі з розробки CRM системи для бізнесу. Також наведені аналогічні системи, які використовувалися

під час роботи. В наступному розділі буде описано стек технологій, який використовувався для розробки системи.



РОЗДІЛ 2

БІБЛІОТЕКИ ТА ТЕХНОЛОГІЇ

В даному розділі описано стек технологій, який використовувався під час розробки CRM системи.

2.1 React

React – це JavaScript бібліотека, розроблена компанією «Facebook» для створення інтерфейсів, яка популяризувала ідею використання Virtual DOM. React створює легке дерево із JavaScript об'єктами для імітації DOM-дерева. Потім він створює з них HTML, що відображається у браузері [10].

DOM (об'єктна модель документа) - це інше представлення веб-сторінки ніж HTML код. Браузер по вказаній URL адресі відправляє запит і отримує (завантажує) з сервера веб-сторінку у вигляді HTML коду, який часто називається вихідний код сторінки. І уже з завантаженого з сервера HTML коду браузер формує DOM. Браузер створює DOM для того, щоб за допомогою JavaScript можна було швидко маніпулювати веб-документом: шукати потрібний елемент, додавати нові елементи, отримати наступний дочірній елемент і так далі [8].

Virtual DOM – концепція, коли відбувається взаємодія не з DOM, а з легкою копією. Ви можете вносити зміни в копію відповідно до потреб, а потім вони застосовуються до справжнього DOM. Під час цього процесу відбувається глибоке порівняння DOM та Virtual DOM. Такий підхід працює набагато швидше, оскільки не включає всі важкі частини реального DOM [9].

Важливо розуміти, що писати об'єкти замість звичайного для HTML дуже незручно. Для вирішення цієї проблеми в React був винайдений JSX.

JSX – це зручний формат запису Virtual DOM елементів. Він дуже нагадує HTML, має всі відповідні теги та події, але є відмінності[11].

Стовпами React є компоненти та пропси. Компоненти дозволяють поділити інтерфейс на частини, що дозволяє їх повторно використовувати.

Компоненти – це функції, які повертаються JSX розмітку, та приймаються параметри – пропси, на основі яких відбувається їх відображення.

Кожен раз функція буде повертати однакову розмітку, але з різним наповненням атрибутів тегу зображення. В цьому і є перевага компонент – вони дозволяють шаблонізувати фрагменти розмітки, які повторюються.

Як зазначалося раніше, компонент оновлюється при зміні пропсів. Щоб оновити віддалені одне від одного компоненти необхідно продумати, як і де вони приходять в компоненти.

2.2 Redux

Redux – бібліотека для створення загального контейнера даних і коли ці дані змінюються, змінюються компоненти, які відображають саме ці дані [12].

Кругообіг даних відбувається за такою схемою.

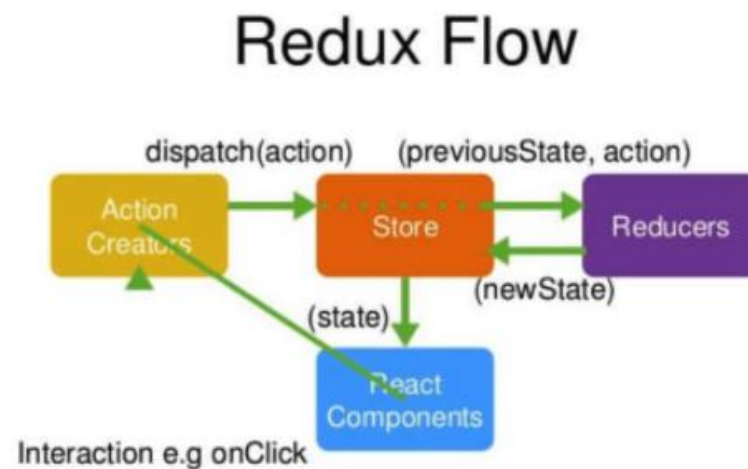


Рисунок 2.1 – Кругообіг даних

На рисунку «store» – це сховище даних, «react-components» – це компоненти, «action creators» – функції, які повертають об’єкт, який повинен мати поле «type» (передавати інші інформацію в цьому об’єкті не заборонено).

Цикл такий:

1. З «store» до компонентів надходять дані, які відображаються на сторінці.

2. Використовуючи спеціальну функцію «dispatch», надану бібліотекою Redux, під час роботи різних подій і методів життєвого циклу в React викликається функція «dispatch» з передається їй «action creator» як параметр.
3. У спеціальній функції «reducer», яка викликає Redux відразу після спрацьовування методу «dispatch», можна змінити «store», відповідно до поля «type» у переданому «action creator» та інших полях, якщо такі є.
4. Коли «store» був змінений, компоненти, які його відображають, оновлюються, і відображаються нові дані.

2.3 React-router-dom

React-router-dom – стандартна бібліотека маршрутизації в React. Вона відслідковує адресну строку браузера і дозволяє відображати або приховувати React компоненти в залежності від значення адресної строки. Тобто по суті дозволяє реалізувати переміщення по сторінках (по суті екранах) веб сайту без перезавантаження сторінки шляхом генерування розмітки в одному HTML файлі [13].

Простіше це зрозуміти на прикладі фрагменту коду розробленої програми в рамках цієї роботи. Є три компоненти «Route», які приймають атрибути «path» - стан адресної строки браузера, при якій буде показуватися компонент

```
<Route path="calendar" element={<EventCalendar />} />
<Route path="listOfWorkers" element={<ListOfWorkers />} />
<Route path="listOfEvents" element={<ListOfEvents />} />
<Route
```

Рисунок 2.2 – Компоненти «Route»

Завдяки спеціальним компонентам – «Router», можна вказати при яких значеннях адресної строки, потрібно відображати ті чи інші компоненти, які передаються в атрибут «element» або «render».

Тобто, якщо користувач перейде за посилання «domain/calendar», то буде відображатися тільки компонента календаря, де «domain» – це домен, на якому розміщений веб-додаток.

2.4 React-redux

React-redux – це бібліотека, яка дозволяє зручно діставати дані з Redux. Вона слугує мостом між React та Redux, саме тому і має таку назву. Для діставання даних використовуються спеціальні функції – селектори.

Селектор приймає в якості параметра весь «store» Redux і повертає обрані дані. Щоб використати селектор в тілі компоненти необхідно, передати його в спеціальну функцію «useSelector» та зберегти результат в змінну.

2.5 Ant design

Ant design – це React бібліотека, яка містить набір готових компонентів. Її завдання – це максимально пришвидшити час на розробку. Це досягається тим, що компоненти вже мають стилі та певний функціонал і розробнику залишається тільки збирати програму наче конструктор [16].

Розробнику не потрібно витрачати час на стилізацію компонентів та прописуванню тривіального функціоналу. Наприклад, такого як меню, календар, компоненти сповіщення і так далі.

2.6 Moment

Moment – це бібліотека для маніпулювання часом [17]. Вона дозволяє:

- Отримувати поточний час.
- Проводити маніпуляції з ним.
- Отримувати час зі строкового формату.
- Форматувати відображення часу.

Дана бібліотека була використана для відображення подій на календарі, задання часу події та обчислення її завершення. Дані часу зазвичай зберігаються у строковому форматі як показано на рисунку 2.3.

Вже потім завдяки цій бібліотеці зчитуються і можна проводити маніпуляції з ним. Час у строковому форматі зберігається так – все що до символу «Т» – то дата, а все що після – час.

```
"startTime": "2022-04-23T09:00",  
"endTime": "2022-04-23T09:15",
```

Рисунок 2.3 – Формат часу

2.7 Redux thunk

Redux Thunk – бібліотека, яка дозволяє Redux працювати асинхронно. Проблема HTTP запитів в тому, що запитам потрібен певний час на виконання, а Redux не може чекати: після отримання «action creator», відразу проходить весь цикл кругообігу даних, не чекаючи відповіді серверу. Тобто, якщо ми будемо робити запити за подіями користувача, ми не зможемо їх додати в «store», щоб відобразити на сторінці. Для вирішення цієї проблеми використовується ця бібліотека [14].

2.8 AXIOS та JSON-server

AXIOS – бібліотека, яка дозволяє робити HTTP запити на JSON-server. Є 4 види запитів на сервер – GET, POST, DELETE, PUT. JSON-server дозволяє зробити імітацію back-end частини і вести front-end розробку у випадках, коли back-end не готовий, а працювати треба. Всі дані зберігаються у форматі JSON. Зазвичай в JSON-server немає авторизації, тому запити може робити будь хто, але є спосіб як її реалізувати. До цієї бібліотеки можна прописати певну надбудову на JavaScript, яка буде видавати унікальний ключ при авторизації, який необхідно буде передавати в якості заголовку при кожному наступному запиту. Заголовок – це певні метадані, які передаються з кожним HTTP запитом.

Ключ авторизації зручніше всього зберігати в COOKIE [15]. COOKIE – це певний унікальний ключ для кожного користувача, який автоматично додається до кожного запиту на сервер. Завдяки їм сервер розрізняє користувачів [15].

Тобто при запиті авторизації він перехоплюється цією надбудовою та перевіряє чи користувач з таким ім'ям та паролем в JSON-server. Якщо є, то у відповіді повертається певний ключ, який і дозволяє робити інші запити.

Також варто зазначити, що всі інші запити перехоплюються іншою частиною надбудови, де перевіряється наявність ключа авторизації в заголовку, якщо ключ є – користувач може робити запити, якщо немає – то йому прийде помилка і програма перенаправить його на сторінку авторизації.

На рисунку 2.4 показана взаємодія всіх вищенаведених компонентів.

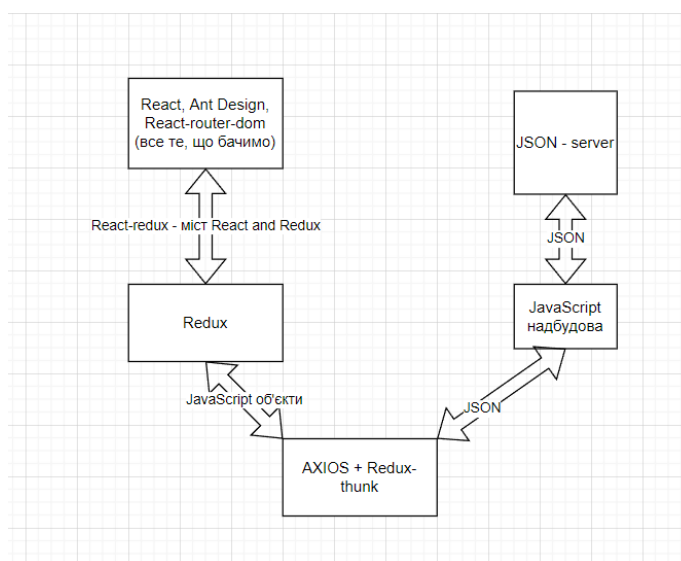


Рисунок 2.4 – Схема взаємодії компонентів

2.9 Visual Studio Code

Середовищем для розробки front-end так і back-end частин було обрано редактор Visual Studio Code.

Visual Studio Code (VS Code) – це спрощений, але потужний редактор вихідного коду, який запускається на комп'ютері й доступний для Windows, macOS і Linux [18].

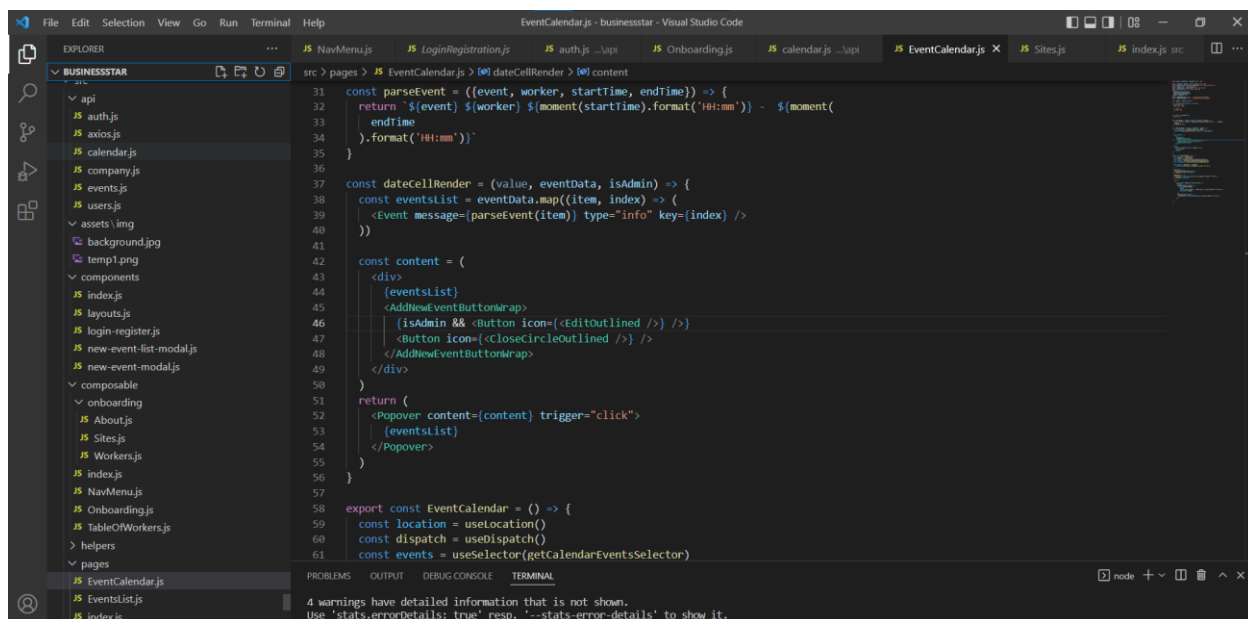


Рисунок 2.5 – Інтерфейс Visual Studio Code

Його переваги в тому, що він універсальний, не займає багато обчислювальних ресурсів комп'ютеру, має багато розширень, які спрощують та прискорюють роботу та має вбудований термінал, а на додачу ще й безкоштовний.

На одному із розширень для цього редактору хотілося б зупинитися більш детально.

«Prettier» - форматувальник коду. Він забезпечує узгоджений стиль, аналізуючи ваш код і передруковуючи його за допомогою власних правил, що враховують максимальну довжину рядка, у разі потреби переносячи код.

На практиці це розширення здатне придати коду читабельного вигляду, не витрачаючи багато часу на його форматування.

На малюнку 2.6 зображений приклад нечитабельного коду. В ньому довжина строки коду занадто довгі, всі вирази в рядок – це дуже сильно ускладнює читання коду, а як наслідок і його доопрацювання – що є невід'ємною частиною розробки.

```

const parseEvent = ({event, worker, startTime, endTime}) => { return `${event} ${worker} ${moment(startTime).format('HH:mm')} - ${moment(endTime).format('HH:mm')}` }

const dateCellRender = (value, eventData, isAdmin) => {const eventsList = eventData.map((item, index) => (<Event message={parseEvent(item)} type="info" key={index} />))

const content = ( <div> {eventsList}<AddNewEventButtonWrap>{isAdmin && <Button icon={<EditOutlined />} />}<Button icon={<CloseCircleOutlined />} />}</div>

return (

```

Рисунок 2.6 – нечитабельний код

Під час зберігання документу розширення аналізує код та форматує його. Після форматування код з рисунку 2.6 став таким, як на рисунку 2.7. Різниця очевидна – в такому коді набагато простіше орієнтуватися.

```

const parseEvent = ({event, worker, startTime, endTime}) => {
  return `${event} ${worker} ${moment(startTime).format('HH:mm')} - ${moment(endTime).format('HH:mm')}`
}

const dateCellRender = (value, eventData, isAdmin) => {
  const eventsList = eventData.map((item, index) => (
    <Event message={parseEvent(item)} type="info" key={index} />
  ))

  const content = (
    <div>
      {eventsList}
      <AddNewEventButtonWrap>
        {isAdmin && <Button icon={<EditOutlined />} />}
        <Button icon={<CloseCircleOutlined />} />
      </AddNewEventButtonWrap>
    </div>
  )

  return (

```

Рисунок 2.7 – відформатований код

Висновок до розділу 2

У цьому розділі були продемонстровані основні бібліотеки для роботи. Було детально розібрано проблематику, яку вирішує React, розробка front-end без повноцінного back-end за допомогою JSON-server.

РОЗДІЛ 3

ПРОЦЕС РОЗРОБКИ CRM СИСТЕМИ

В цьому розділі описано процес користування розроблюваною системою. Наведено основні екрани та описано їх функціонал.

3.1 Процес реєстрації

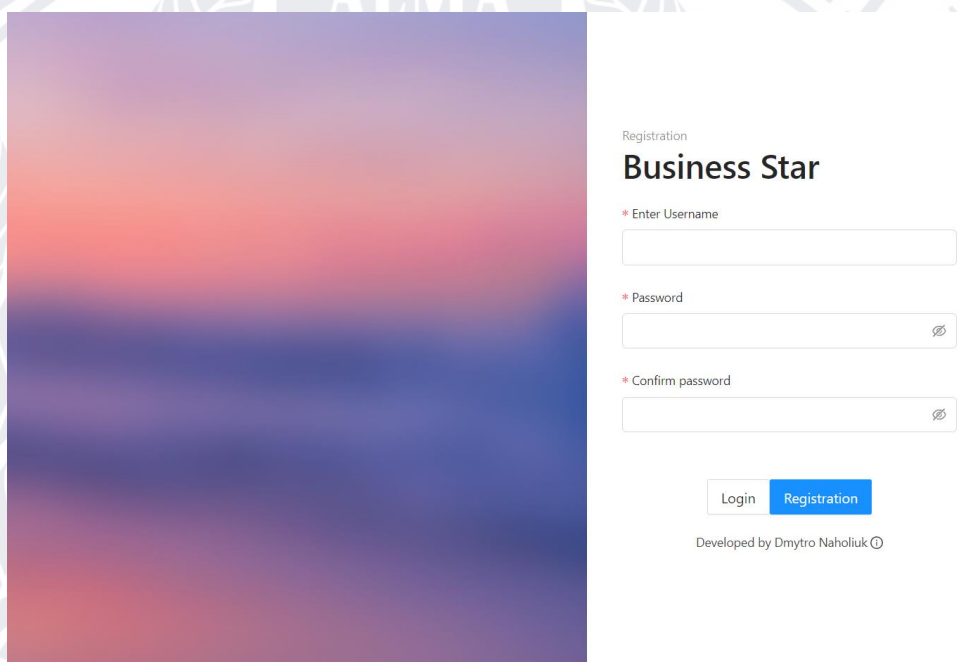
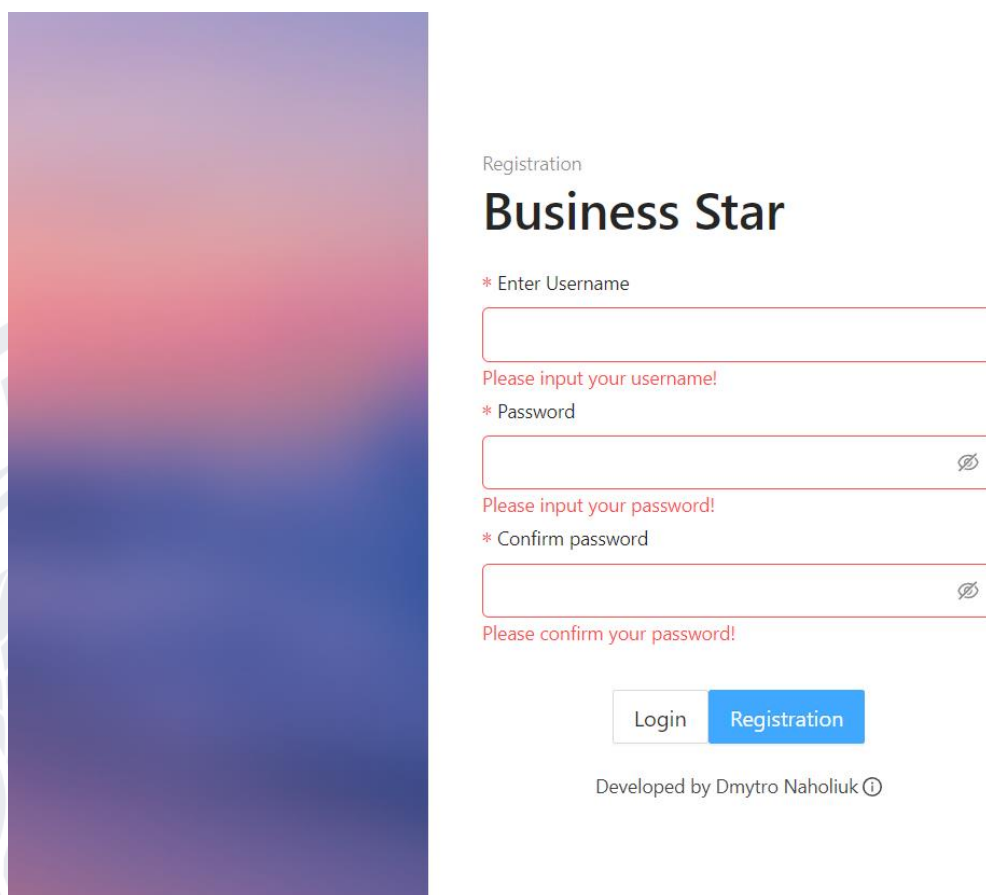


Рисунок 3.1 – Сторінка реєстрації

Інтерфейс сторінки реєстрації представлений декоративним зображенням ліворуч та формою для вводу даних праворуч. У формі є поля для заповнення імені нового користувача, паролю та підтвердження паролю.

При цьому у формі є два види валідації – перша перед відправкою даних на JSON-server, інший – після. Тут варто додати, що валідація – це перевірка даних на коректність. Якщо спрацював перший вид валідації, то відправки форми на JSON-server не відбувається. Розглянемо перший варіант.



Registration

Business Star

* Enter Username

Please input your username!

* Password

Please input your password!

* Confirm password

Please confirm your password!

Login Registration

Developed by Dmytro Naholiuk

Рисунок 3.2 – Валідація пустих полів форми

Якщо користувач нічого не введе і відразу натисне кнопку «Registration» (реєстрація), то спрацюють валідатори полів форми і буде показано повідомленням заповнити відповідні поля форми, що зображено на рисунку 3.2.

У випадку, якщо користувач введе у будь-яке поле значення, що менше 4 символів і натисне кнопку реєстрації, то спрацює валідація, яка попросить його ввести значення з більшою кількістю символів. Зображено на рисунку 3.3.

* Password

'password' must be at least 4 characters

* Confirm password

'confirmPassword' must be at least 4 characters

Login

Registration

Developed by Dmytro Naholiuk ⓘ

Рисунок 3.3 – Валідація недостатньої кількості символів

* Password

'password' cannot be longer than 10 characters

* Confirm password

'confirmPassword' cannot be longer than 10 characters

The two passwords that you entered do not match!

Рисунок 3.4 – Валідація надлишкової кількості символів

У випадку, якщо користувач введе у будь-яке поле значення, що більше 10 символів і натисне кнопку реєстрації, то спрацює валідація, яка попросить його ввести значення з меншою кількістю символів, що зображено на рисунку 3.4.

* Password

* Confirm password

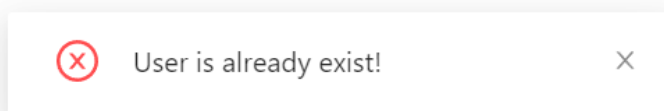
The two passwords that you entered do not match!

Рисунок 3.5 – Валідація поля повторного вводу паролю

У випадку, якщо користувач введе різні значення в поля паролю та підтвердження паролю, з'явиться повідомлення з інформуванням, що паролі не співпадають. Зображено на рисунку 3.5.

Першу частину валідації розглянули, тепер час для другого виду валідації – вже після відправки даних. Дана валідація реалізована у вигляді сповіщень, що з'являються у верхньому правому куту екрану після отримання відповіді від JSON-server. При цьому через декілька секунд сповіщення зникне або можна зробити це швидше, натиснувши на хрестик у верхньому правому кутку сповіщення.

Якщо користувач, під час реєстрації введе ім'я користувача, що вже існує, то з'явиться сповіщення – «User is already exist». В перекладі з англійської – користувач з таким ім'ям вже існує. Воно зображене на рисунку 3.6.



Registration

Business Star

* Enter Username

Gordon

Рисунок 3.6 – Валідація, що користувач з таким ім'ям існує

Тепер розглянемо успішний процес реєстрації в програмі. Заповнивши форму та натиснувши кнопку реєстрації, у разі успіху, користувачу прийде сповіщення – «Successfully created user!», що перекладається – користувач успішно створений.

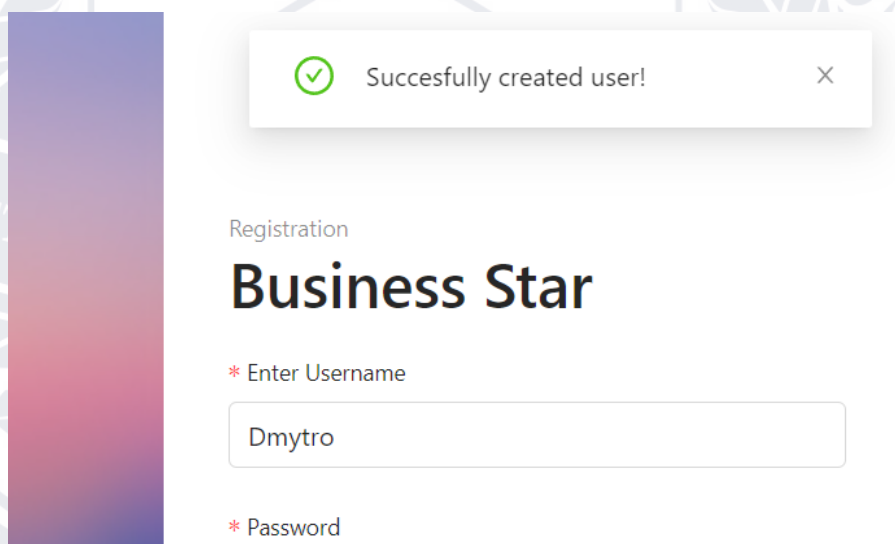
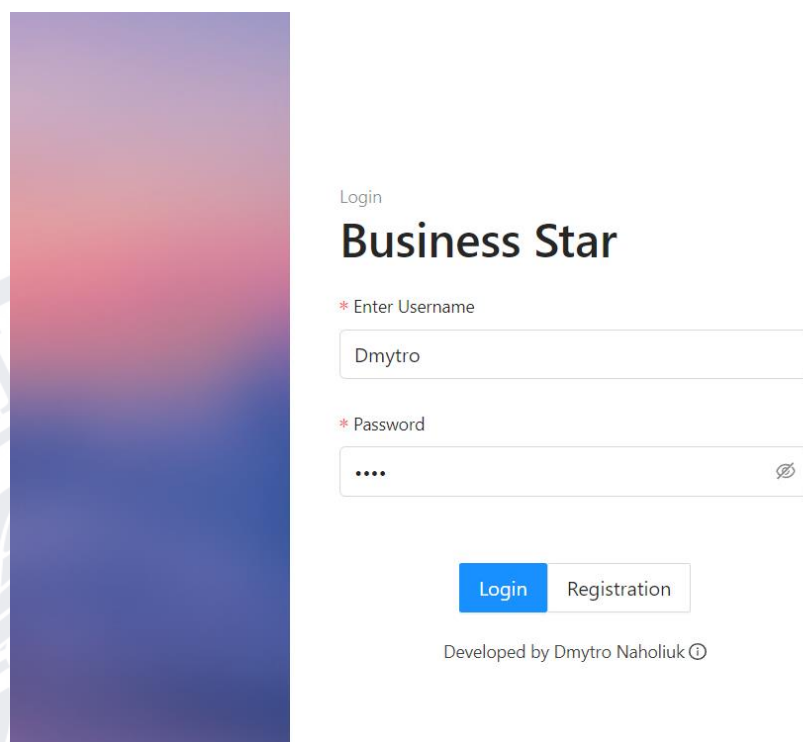


Рисунок 3.7 – Користувач успішно створений

Після того, як користувач був створений, відбудеться перехід на сторінку авторизації. Авторизація – це вхід до акаунту користувача. Де він повинен заповнити поля імені та паролю, а після – натиснути кнопку «Login».



The image shows a login form for 'Business Star'. On the left is a vertical rectangular area with a purple-to-pink gradient. The form itself is on the right, titled 'Login Business Star'. It contains two input fields: 'Enter Username' with the text 'Dmytro' and 'Password' with masked characters '....'. Below these are two buttons: 'Login' (blue) and 'Registration' (white). At the bottom, it says 'Developed by Dmytro Naholiuk' with a small icon.

Login

Business Star

* Enter Username

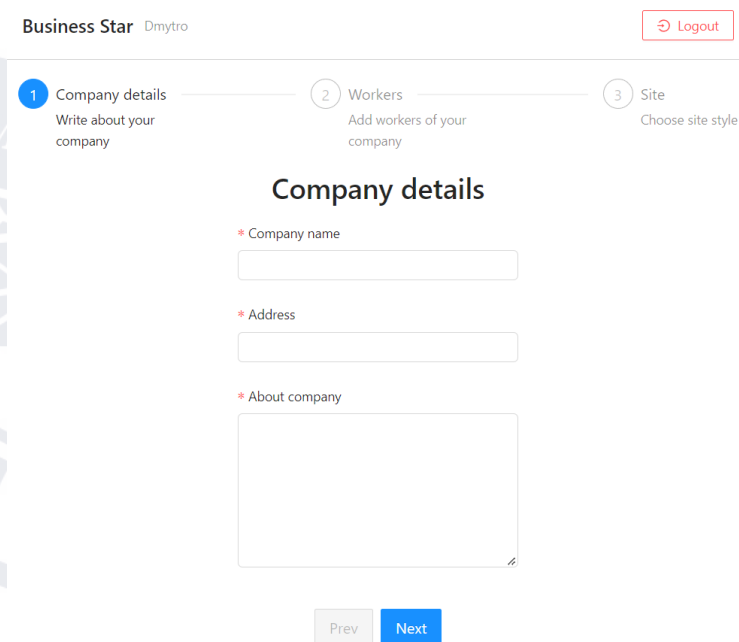
* Password

Login Registration

Developed by Dmytro Naholiuk

Рисунок 3.8 – Сторінка авторизації

Після успішної авторизації нового користувача, перенаправляє на сторінку «Onboarding».



The image shows the 'Onboarding' page for 'Business Star'. At the top, it says 'Business Star Dmytro' and has a 'Logout' button. Below is a progress bar with three steps: 1. Company details (active), 2. Workers, and 3. Site. The 'Company details' section has three input fields: 'Company name', 'Address', and 'About company'. At the bottom are 'Prev' and 'Next' buttons.

Business Star Dmytro Logout

1 Company details Write about your company

2 Workers Add workers of your company

3 Site Choose site style

Company details

* Company name

* Address

* About company

Prev Next

Рисунок 3.9 – «Onboarding» сторінка

«Onboarding» – сторінка, на якій заповнюються дані про компанію. «Onboarding» складається з шапки сайту, де зліва зображено назву програми, а поруч ім'я авторизованого користувача. В правій частині сайту кнопка «Logout», натиснувши на яку, відбувається вихід з акаунту. Сам процес заповнення даних поділений для зручності на 3 кроки. Візуально можна бачити, на якому кроці знаходиться користувач, завдяки візуальному представленню під шапкою сайту.

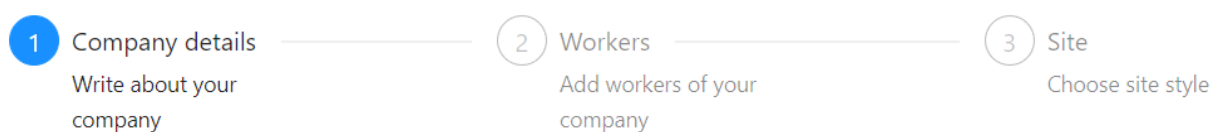


Рисунок 3.10 – Візуальне представлення

Детально розглянемо 3 кроки заповнення інформації про компанію.

«Company details» – на ньому користувач має заповнити такі дані про свою компанію – її ім'я, адресу та опис. Якщо він цього не зробить, а відразу буде намагатися перейти до другого кроку, натиснувши кнопку «Next», то біля відповідних полів з'явиться прохання заповнити їх.

Company details

* Company name

'name' is required

* Address

'address' is required

* About company

'about' is required

Prev

Next

Рисунок 3.11 – Валідація форми вводу даних про компанію

Якщо заповнити дані коректно та натиснути кнопку «Next», то відбудеться зберігання їх та перехід до кроку 2.

Company details

* Company name

Avtobaza

* Address

Вінниця, вул Соборна 1

* About company

Все для вашого авто!

Prev

Next

Рисунок 3.12 – Коректно заповнені дані про компанію

Крок 2 – додавання співробітників до компанії.

Business Star Dmytro [Logout](#)

1 Company details Write about your company
 2 Workers Add workers of your company
 3 Site Choose site style

Workers

[Add new worker](#)

Username	Password	Tags	Action
Dmytro	1111	Director	Delete

< 1 >

[Prev](#) [Next](#)

Рисунок 3.13 – Крок 2

На цьому етапі відображається таблиця з співробітниками компанії. В цій таблиці відображаються дані співробітників – такі як ім'я, пароль та чи є користувач одним із керуючих компанії. Якщо так, то це позначається в стовпчику «Tags».

В таблиці знаходиться кнопка «Add new worker», натиснувши на яку, відкриється діалогове вікно, щоб заповнити дані про нового співробітника.

Рисунок 3.14 – Діалогове вікно для вводу даних користувача

Валідація полів цього вікна така сама, як при реєстрації нового користувача, окрім того, що не потрібно вводити пароль ще раз для підтвердження паролю, тому детально зупинятися на валідації не будемо.

Якщо заповнити форму коректно та натиснути – «Ок», то відбудеться додавання даного користувача до компанії і він відобразиться в таблиці.

Рисунок 3.15– коректно заповнені дані

Add new worker			
Username	Password	Tags	Action
Dmytro	1111	Director	Delete
Andriy	1111		Delete

<
1
>

Рисунок 3.16 – Таблиця з доданим користувачем «Andriy»

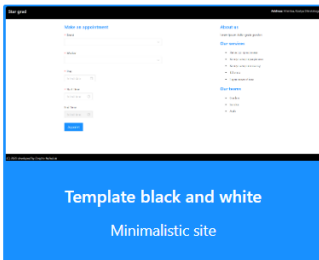
Можна переходити до 3 кроку. На цьому кроці потрібно буде заповнити дані для сайту візитки компанії. Такі як – обрати шаблон сайт візитки та назву піддомену.

✓ Company details Write about your company
 ✓ Workers Add workers of your company
 3 Site Choose site style

Sites

Choose template for your site

* Sub domain name:



Template black and white
Minimalistic site

Prev
Finish

Рисунок 3.17 – Крок 3

Поки що доступний один шаблон сайту візитки, який і обраний автоматично.

Варто зазначити, що за допомогою кнопки «Prev» можна повернутися на попередні кроки і відкоригувати дані. При цьому кнопка перестане бути

доступною, коли користувач повернеться на перший крок. На 3 кроці замість напису «Next» в кнопці відправки форми замінюється на «Finish», що означає фінальний крок. Після натиснення на «Finish» завершується процес реєстрації.

3.2 Основний інтерфейс

Після авторизації користувач потрапляє на головний екран програми. Головний екран являє собою календар авторизованого користувача, де відображаються всі події.

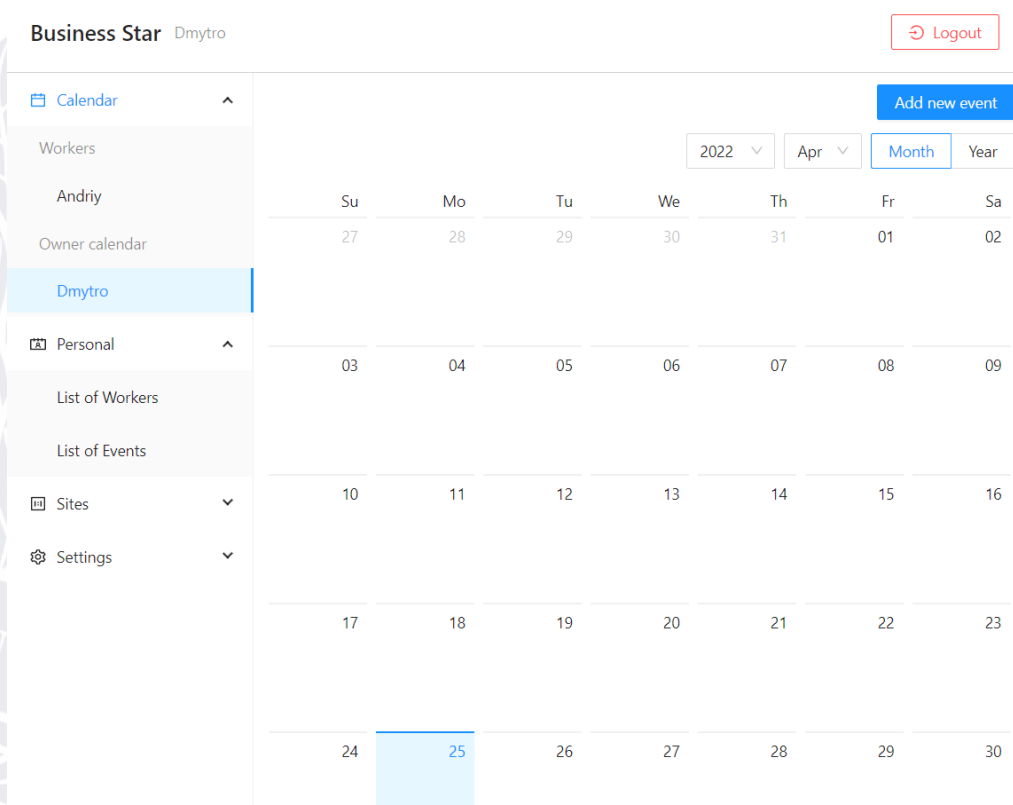


Рисунок 3.18 – Головний екран

Праворуч знаходиться меню. Який складається таких груп:

«Calendar» – у цій групі меню знаходяться посилання на календарі усіх співробітників.

«Personal» – у цьому пункті є посилання на дві списки усіх співробітників та список подій, який може бути призначений на календарі.

«Sites» – у цьому пункті посилання на сайт-візитку.

3.3 Календар та події

Події – це послуги, які компанія надає. Обравши пункт, події, користувачу буде доступний список усіх подій компанії у вигляді таблиці, де вказана назва події та її часова тривалість. У новоствореної компанії будь-які події відсутні.

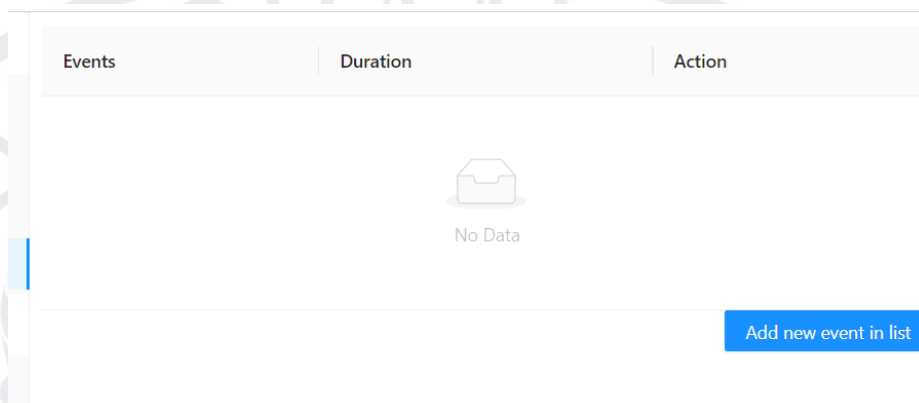


Рисунок 3.19 – Пуста таблиця подій

Щоб додати подію до компанії, необхідно натиснути «Add new event in list» та заповнити форму в діалоговому вікні. Форма складається з поля назви події та її тривалості (в хвилинах).

Add new event in list

* New event:

* Duration (min):

Cancel

OK

Рисунок 3.20 – Діалогове вікно додавання події

При цьому на формі є валідація полів, як в попередніх прикладах.

Add new event in list ⓧ

* New event:
'event' is required

* Duration (min):
'duration' is required

Cancel OK

Рисунок 3.21 – Валідація форми додавання події

Якщо ввести дані коректно, то додані події відобразяться в списку подій.

Events	Duration	Action
Консультація по ОСАГО	30 min	Edit Delete
Заміна мастил	30 min	Edit Delete
Огляд ходової частини авто	60 min	Edit Delete

< 1 >

Add new event in list

Рисунок 3.22 – Таблиця з подіями

Коли заповнена таблиця з подіями, користувач може призначати їх на календарі собі або своїм співробітникам. Для цього потрібно перейти на календар співробітника, якому хочете призначити подію та натиснути кнопку «Add new event».

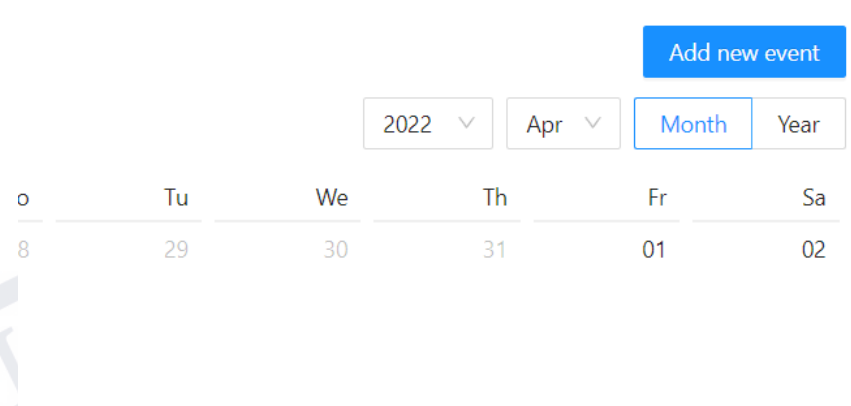


Рисунок 3.23 – Кнопка «Add new event»

Після її натискання відкриється діалогове вікно з формою додавання події на календар.

Add new event (Close icon)

* Event:

* Worker:

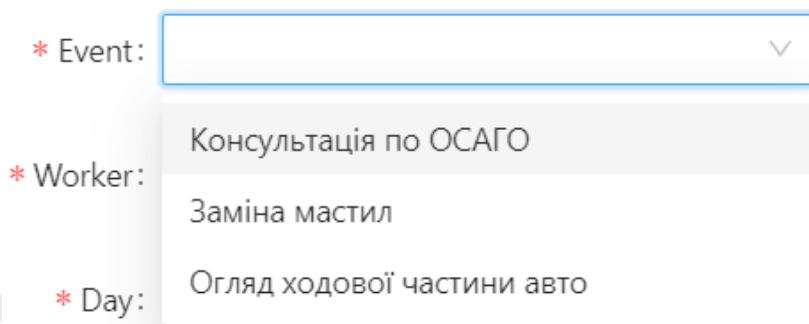
* Day: (Calendar icon)

* Start Time: (Clock icon)

End Time: (Clock icon)

Рисунок 3.24 – Форма додавання події

В формі є поле «Event», де можна обрати подію, яку хочемо додати в календар. Всі раніше додані події доступні у вигляді випадаючого списку.



* Event:

* Worker:

* Day:

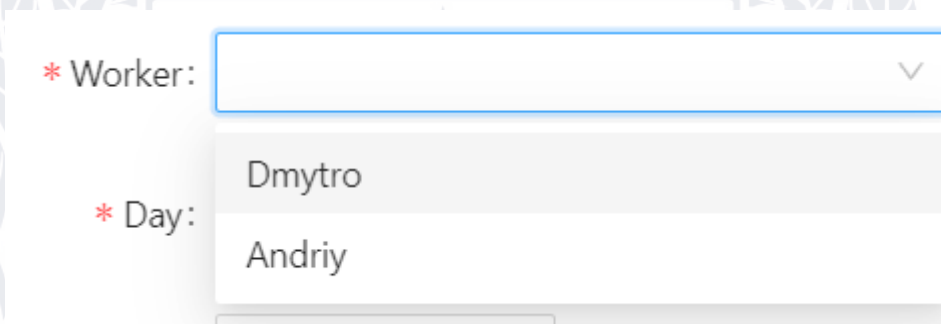
Консультація по ОСАГО

Заміна мастил

Огляд ходової частини авто

Рисунок 3.25 – Випадаючий список подій

Також є поле «Worker», де можемо обрати робітника, якому призначимо подію. Все також у формі випадаючого списку.



* Worker:

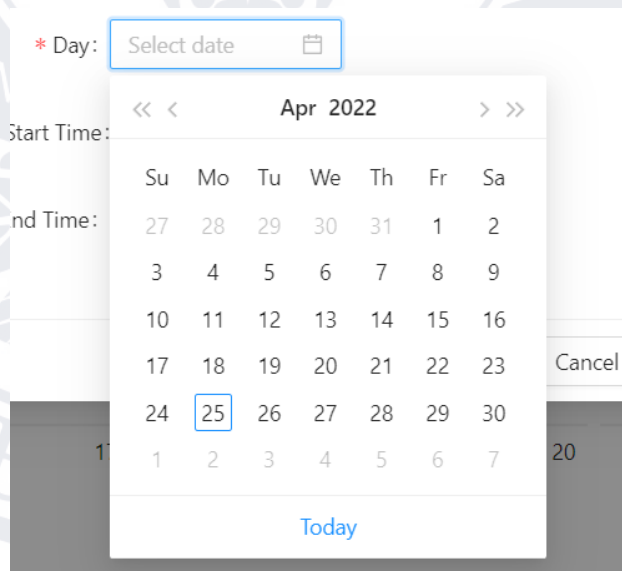
* Day:

Dmytro

Andriy

Рисунок 3.26 – Поле «Worker»

У полі «Day» обирається день події на маленькому календарикі.



* Day:

Start Time:

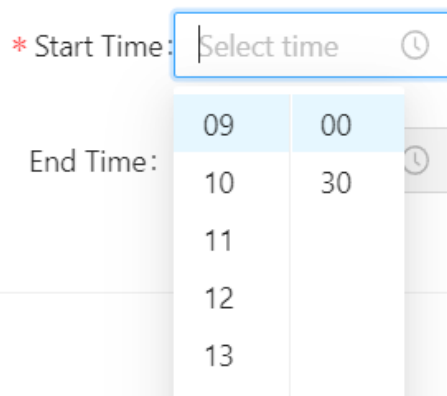
End Time:

Cancel

Today

Рисунок 3.27 – Календар вибору дню події

І врешті-решт, поле «Start time», де обирається точний час події.



* Start Time: Select time

End Time: 09:00

10:30

11:00

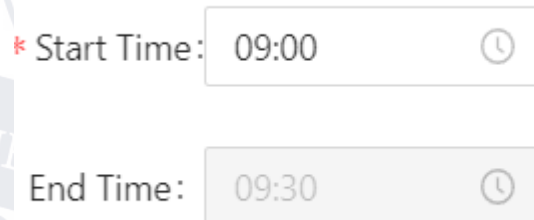
12:00

13:00

Рисунок 3.28 – Вибір часу події

При цьому в залежності від тривалості події, змінюється крок вибору хвилин для події. Наприклад, якщо подія триває 30 хвилин, то крок хвилин буде 9:00 – 9:30, 9:30 – 10:00. А якщо година, то крок буде 9:00 – 10:00.

Обравши час запису, його кінець автоматично вираховується в залежності від тривалості події.



* Start Time: 09:00

End Time: 09:30

Рисунок 3.29 – Часові рамки події

Після натискання кнопки – «Ок» та успішного додавання події, відображається повідомлення про успіх.

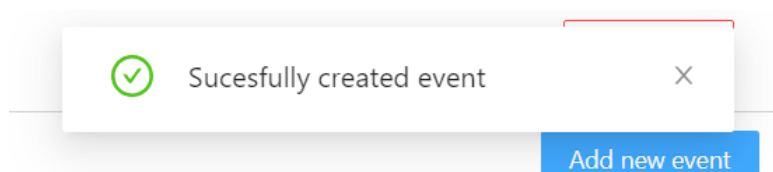


Рисунок 3.30 – Повідомлення про успішно створений запис в календар

І подія починає відображатися на календарі

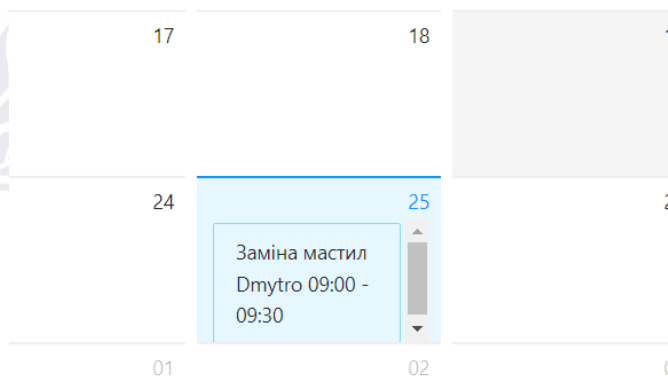


Рисунок 3.31 – Відображення події

Подій може бути багато і всі вони не поміщуються в комірку календаря. Щоб розгорнути список подій необхідно клікнути на комірку.

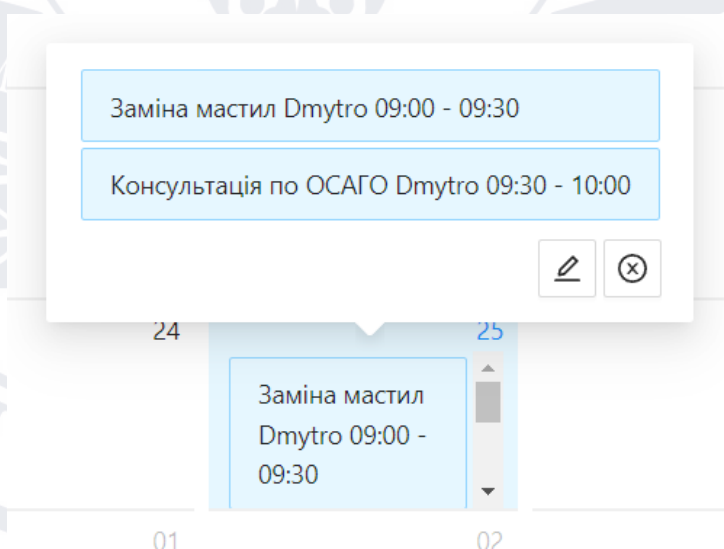


Рисунок 3.32 – Розгорнутий список подій

Якщо перейти на до календарю співробітника «Andriy», то можна побачити, що ніяких записів на 25 число немає, тому що запис був до працівник «Dmytro»

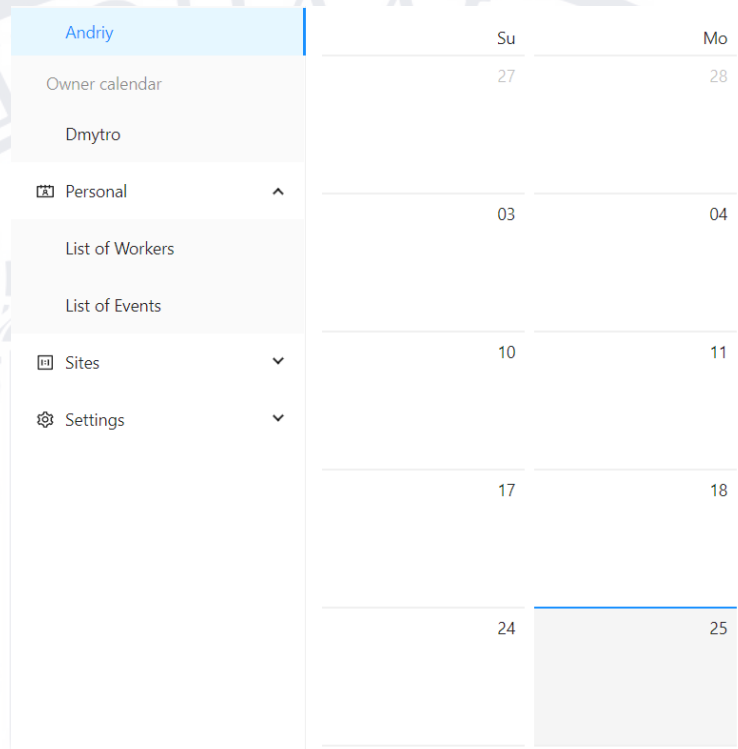


Рисунок 3.33 – Календар працівника «Andriy»

3.4 Сайт візитка

Також генерується сайт візитка, який може бути розміщений у мережі, де будуть відображатися дані про компанію та можливість створення події вже не користувачем, а потенційним клієнтом компанії.

Avtobaza Address: Вінниця, вул Соборна 1

Make an appointment

* Event

* Worker

* Day

* Start Time

End Time

Appoint

About us

Все для вашого авто!

Our services

- Консультація по ОСАГО
- Заміна мастил
- Огляд ходової частини авто

Our teams

- Dmytro
- Andriy

(C) 2022 developed by Dmytro Nahaliuk

Рисунок 3.34 – Сайт візитка

На сайті візитці є шапка, де праворуч назва компанії, а ліворуч адреса. Праворуч також відображається інформація про компанію – така як опис, її послуги (тобто всі додані події) та її співробітники.

Ліворуч потенційний клієнт може зробити запис до певного співробітника аналогічно тому, як це робиться в календарі. За допомогою цього підприємець може контролювати своїх співробітників та автоматизувати процес запису в компанії.

3.5 Список співробітників

Якщо користувач захоче додати ще працівників або видалити, то для цього йому потрібно перейти в пункт меню «List of workers».

Add new worker			
Username	Password	Tags	Action
Dmytro	1111	Director	Delete
Andriy	1111		Delete

< 1 >

Рисунок 3.35 – «List of workers»

Тут він може додавати або видаляти співробітників.

3.6 Авторизація

Щоб вийти з акаунту користувачу необхідно натиснути кнопку «Logout». Після цього відбудеться вихід з акаунту та перенаправлення на сторінку авторизації.

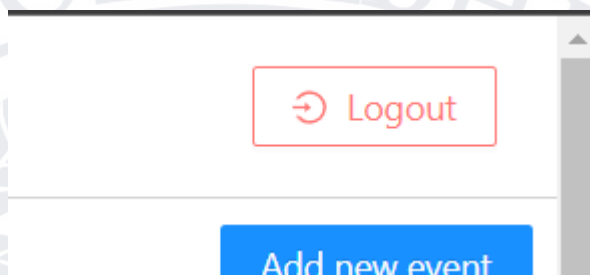


Рисунок 3.36 – Кнопка «Logout»

На сторінці авторизації є кнопки перемикавання між режимами авторизації та реєстрації. Синім кольором виділяється активний режим.

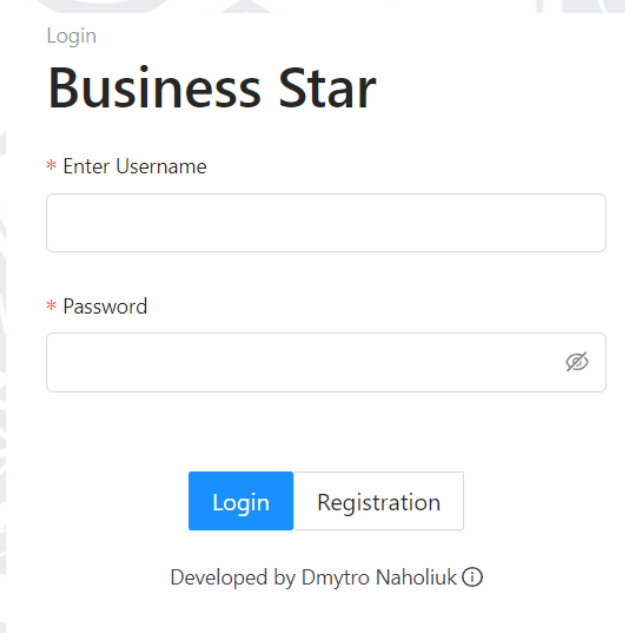


Рисунок 3.37 – Кнопки перемикавання між режимом авторизації та реєстрації

Якщо ввести неправильне ім'я або пароль, то відобразиться сповіщення, що пароль або ім'я користувача введено неправильно.

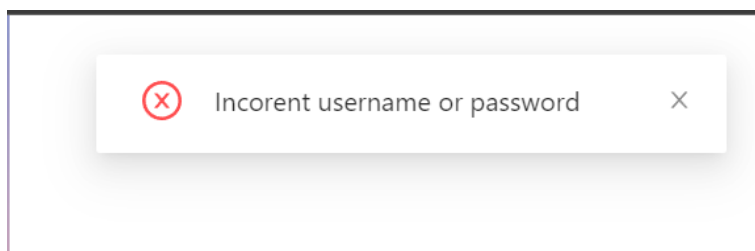


Рисунок 3.38 – Сповіщення неправильного паролю або імені

Також натиснувши на іконку «око» на полю вводу паролю, можна подивитися, що було введено.



Рисунок 3.39 – Поле вводу паролю

3.7 Авторизація як співробітник без прав доступу

У системі реалізовано права доступу. У того, хто зареєстрував компанію права доступу автоматично розширені, а щоб наділити цими правами своїх співробітників потрібно під час додавання співробітника поставити галочку навпроти пункту «Admin rights».

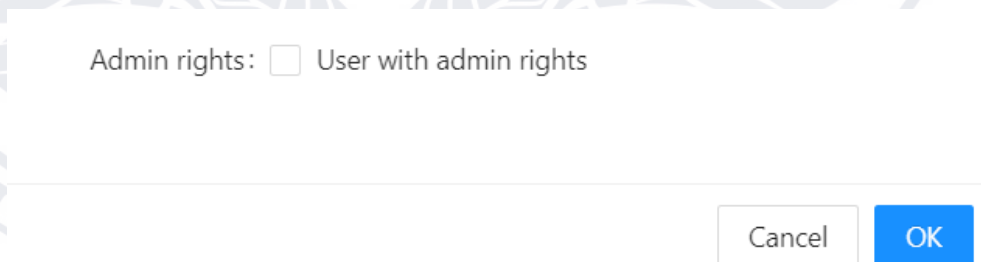


Рисунок 3.40 – Поле вводу паролю

Для працівників, які не мають цього права доступу до повного функціоналу CRM, інтерфейс програми дещо відрізняється. На рисунку 3.41 видно, що в меню доступні лише пункти з календарями співробітників.

Business Star Soroka

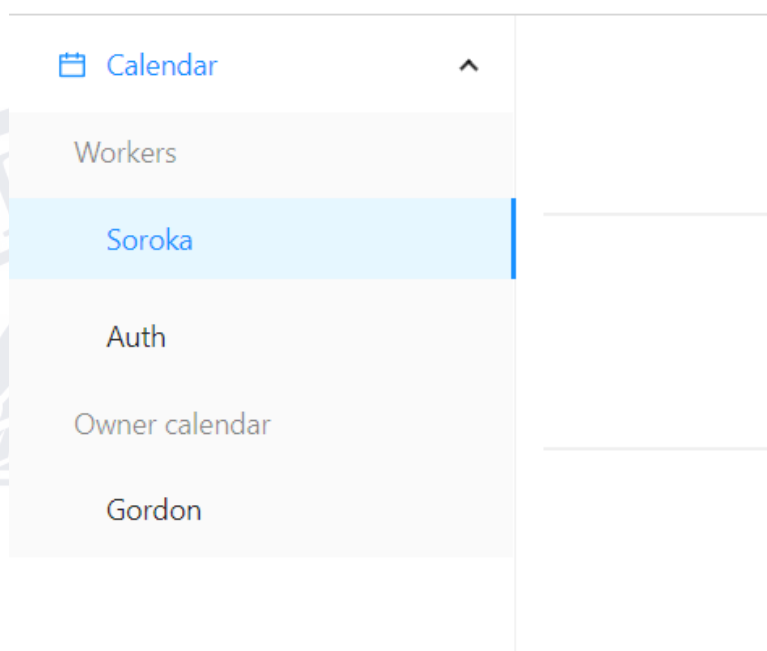


Рисунок 3.41 – меню працівника без права доступу

Всі інші пункти – налаштування, список співробітників подій, сайт не відображаються.

Також у такого співробітника відсутня кнопка додавання події на календар, яка зазвичай знаходилась над ним з правого боку, він може тільки переглядати.

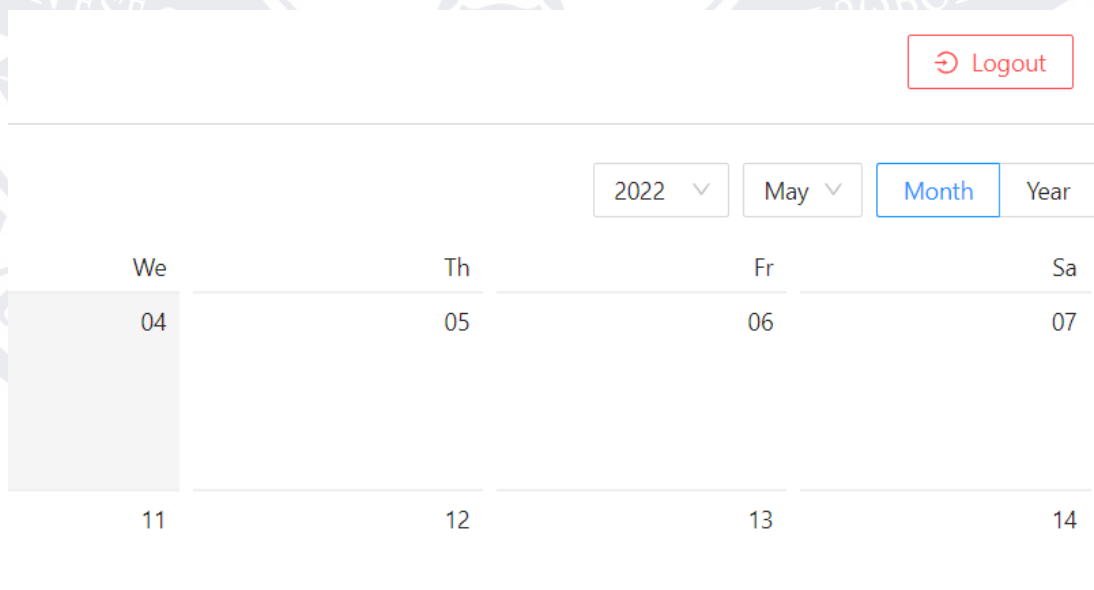


Рисунок 3.42 – календар працівника без права доступу

Якщо розгорнути список подій, то можна побачити, що іконка, редагування зникла.

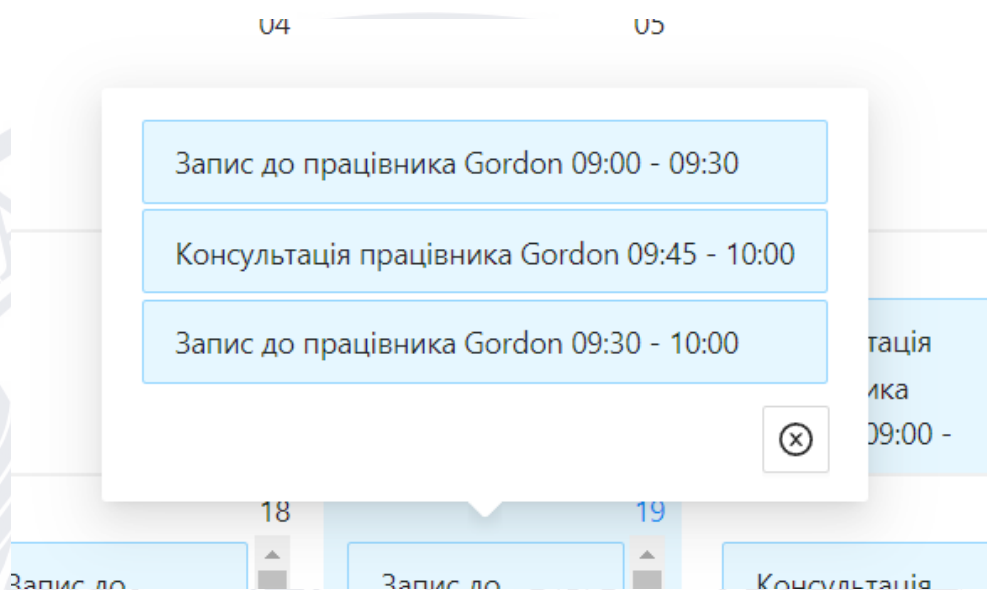


Рисунок 3.43 – розгорнутий список подій працівника без права доступу

Висновок до розділу 3

В цьому розділі було продемонстровано результати виконаної роботи, вдалося реалізувати функції календарю, авторизації та реєстрації, додавання користувачів, подій та шаблонний сайт візитка.

ВИСНОВКИ

Використовуючи стек технологій для розробки веб-додатків, було розроблено CRM-систему для управління процесами підприємства. Було вивчено методику проектування front-end частини CRM-системи на мові програмування JavaScript, використовуючи React та ознайомлено з процесом розробки сучасних веб додатків без повноцінного back-end за допомогою JSON-server.

Було проведено огляд існуючих CRM-систем, вивчено їх недоліки та переваги, була наведена відмінність розроблюваної CRM-системи від аналогів.

Також було детально розглянуто необхідний стек технологій для розробки. Було детально розібрано проблематику DOM в сучасних реаліях. Принципи віртуального DOM дерева. Основи, на яких побудована бібліотека React та архітектура організації потоку даних за допомогою Redux. Також було розібрано було розібрано велику кількість бібліотек для розробки та зображено взаємозв'язок між ними. Було реалізовано можливість авторизації в JSON-server.

В CRM-системі був реалізований весь запланований функціонал:

1. Реєстрація нового користувача.
2. Процес заповнення даних про компанію.
3. Додавання співробітників до компанії.
4. Вибір шаблону сайту візитки.
5. Меню з посиланнями на календарі співробітників.
6. Формування списку подій
7. Календар з можливістю відображення подій для кожного працівника компанії на вибрану дату та додавання подій на нього.
8. Сайт візитка з можливістю запису до співробітника компанії та відображення інформації про компанію.
9. Повторний вхід користувача.
10. Права доступу для користувачів.
11. Можливість виходу з акаунту.

Розроблена CRM-система відрізняється від аналогів своєю універсальністю: кожен може її застосовувати – від керівника невеликої перукарні до автосервісу.

В ході виконання роботи, було підвищено навички в програмуванні JavaScript та бібліотек до нього.



СПИСОК ЛІТЕРАТУРИ

1. Що таке CRM-система? Автор оригіналу: Retailcrm. URL: <https://www.retailcrm.ru/blog/chto-takoe-crm-i-zachem-ona-nuzhna>
2. Хто такі frontend- і backend-розробники?. Автор оригіналу: te.itsep.org. URL: [https://te.itstep.org/blog/frontend-and-backend-developers#:~:text=Frontend%20-%20це%20розробка%20користувачького%20інтерфейсу,i%20з%20чим%20%](https://te.itstep.org/blog/frontend-and-backend-developers#:~:text=Frontend%20-%20це%20розробка%20користувачького%20інтерфейсу,i%20з%20чим%20%20)
3. What is JavaScript? Автор оригіналу: Hack Reactor. URL: <https://www.hackreactor.com/blog/what-is-javascript-used-for>
4. Початок роботи. Автор оригіналу: uk.reactjs.org. URL: <https://uk.reactjs.org/docs/getting-started.html>
5. Создание фейкового REST API с помощью json-server-a. Автор оригіналу: Roy Agasthyan. URL: <https://code.tutsplus.com/ru/tutorials/fake-rest-api-up-and-running-using-json-server--cms-27871>
6. What is HTTP? Автор оригіналу: Cloudflare. URL: <https://www.cloudflare.com/learning/ddos/glossary/hypertext-transfer-protocol-http/>
7. What is JSON? Автор оригіналу: json.org. URL: <https://www.json.org/json-en.html>
8. What is DOM. Автор оригіналу: Freecodecamp. URL: <https://www.freecodecamp.org/news/what-is-the-dom-document-object-model-meaning-in-javascript/>
9. Что такое virtual DOM. Автор оригіналу: Webdevblog. URL: <https://webdevblog.ru/chto-takoe-virtual-dom/>
10. What is React. Автор оригіналу: Alesia Sirotko. URL: <https://flatlogic.com/blog/what-is-react/>
11. Знакомство с JSX. Автор оригіналу: React. URL: <https://ru.reactjs.org/docs/introducing-jsx.html>

12. What is Redux. Автор оригіналу: Logrocket. URL: <https://blog.logrocket.com/why-use-redux-reasons-with-clear-examples-d21bffd5835/>
13. React Маршрутизация. Автор оригіналу: Metanit. URL: <https://metanit.com/web/react/4.1.php>
14. Как делать асинхронные Redux экшены используя Redux-Thunk?. Автор оригіналу: Alligator.io. URL: <https://habr.com/ru/post/483314/>
15. Cookie HTTP. Автор оригіналу: Cloudflare. URL: <https://www.cloudflare.com/learning/privacy/what-are-cookies/>
16. Getting Started. Автор оригіналу: Ant design. URL: <https://ant.design/docs/react/getting-started>
17. Moment. Автор оригіналу: Moment. URL: <https://momentjs.com/>
18. Зведення. Автор оригіналу: Microsoft. URL: <https://docs.microsoft.com/uk-ua/power-apps/maker/portals/vs-code-extension>
19. Prettier. Автор оригіналу: Prettier. URL: <https://marketplace.visualstudio.com/items?itemName=esbenp.prettier-vscode>
20. Кращі CRM-системи. Автор оригіналу: UGUIDE. URL: <https://uguide.ru/rejting-luchshie-crm-sistemy-dlja-biznesa>
21. Персоналізація та автоматизація процесів. Автор оригіналу: Unicon. URL: https://ubpo.ru/press/publications/it_na_sluzhbe_hr_personalizatsiya_i_avtomatizatsiya_protsesov/
22. Бізнес Online. Автор оригіналу: Бізнес Online. URL: <https://www.business-gazeta.ru/article/538344#:~:text=CRM-системы%20предлагают%20максимальную%20информативность,и%20позволяют%20объективно%20оценить%20момент>
23. Що таке шаблон сайту? Автор оригіналу: Templatemonster. URL: <https://www.templatemonster.com/ru/faq/chto-takoe-template/>
24. Управління персоналом. Автор оригіналу: Gekoms. URL: <https://gekoms.org/2012/11/10/note-16/>

25. Частка малого та середнього бізнесу. Автор оригіналу: ZN, UA. URL: https://zn.ua/ukr/ECONOMICS/chastka-malogo-i-serednogo-biznesu-u-vvp-ukrayini-55-mert-315597_.html
26. Як зробити сайт візитку? Автор оригіналу: Weblium. URL: <https://ru.weblium.com/blog/kak-sdelat-sayt-vizitky-istrukciya/>
27. Авторизація. Автор оригіналу: uk.theastrologypage.com. URL: <https://uk.theastrologypage.com/development/1>
28. Автоматизація бізнес процесів. Автор оригіналу: www.softex.if.ua. URL: <https://www.softex.if.ua/services/proektne-vprovadzhennya/avtomatizatsiya-biznes-protseviv/#:~:text=Відповідно%2C%20автоматизація%20бізнес-процесів%20—,цілому%20підвищують%20профїт%20роботи%20підприємства>
29. Піддомен. Автор оригіналу: CITYHOSY.UA. URL: <https://cityhost.ua/uk/blog/chto-takoe-poddomen-i-dlya-chego-on-nuzhen.html>
30. Домен. Автор оригіналу: Домени. URL: <https://regery.ua/uk/blog/post/chto-takoe-domen>