

**РІВНЕНСЬКИЙ ДЕРЖАВНИЙ ГУМАНІТАРНИЙ УНІВЕРСИТЕТ**

Факультет математики та інформатики

Кафедра інформатики та прикладної математики

«До захисту допущено»

Завідувач кафедри

\_\_\_\_\_ Батишкіна Ю.В.

« \_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ р.

**Дипломний проект (робота)**

ступеня «Магістр»

з напрямку підготовки (спеціальності) 122 – Комп'ютерні науки

на тему: Використання інформаційних технологій для створення віртуальних ярмарок

Виконав: студент 2 курсу, групи М-КН-21

\_\_\_\_\_ Бурнасенков Олег Андрійович \_\_\_\_\_

Керівник: к.ф.-м.н., доц. Шахрайчук М. І. \_\_\_\_\_

Рецензент: канд. тех. наук, доц. Шинкарчук Н.В. \_\_\_\_\_

Рецензент: д-р. пед. наук, проф. Войтович І.С. \_\_\_\_\_

Засвідчую, що у цьому дипломному проекті немає запозичених матеріалів з праць інших авторів без відповідних посилань.

Студент \_\_\_\_\_

(підпис)

Рівне – 2020 року

	2
СПИСОК УМОВНИХ ТЕРМІНІВ ТА ПОЗНАЧЕНЬ .....	3
ВСТУП.....	4
РОЗДІЛ 1. ОГЛЯД АНАЛОГІВ ПРОГРАМНОГО ПРОДУКТУ ТА ІНСТРУМЕНТАРІЮ СТВОРЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ .....	7
1.1. Огляд аналогів .....	7
1.2. Огляд інструментарію .....	9
РОЗДІЛ 2. МОДЕЛЮВАННЯ ПРОГРАМНОГО ПРОДУКТУ .....	22
2.1. UX-проектування.....	22
2.2. UI-дизайн.....	23
2.3. Верстка сторінок.....	28
2.4. Розробка бази даних .....	29
2.5. Основні функції додатку.....	37
2.5.1. Реєстрація та аутентифікація користувачів .....	37
2.5.2. Реалізація подання користувача на вакансію .....	40
2.5.3. Налаштування інструментарію для відеодзвінків.....	42
2.5.4. Налаштування інструментарію для чату між користувачами .....	43
РОЗДІЛ 3. ІНСТРУКЦІЯ З ВИКОРИСТАННЯ .....	45
3.1. Сторінка реєстрації користувача та вхід у систему .....	45
3.1.1. Сторінка реєстрації .....	45
3.1.2. Функція ”Забув пароль”.....	49
3.2 Адміністраторська панель .....	49
3.2.1. Адміністраторська панель адміністратора ярмарки .....	50
3.2.2. Адміністраторська панель менеджера компанії.....	58
3.2.3. Адміністраторська панель користувача .....	61
3.3 Огляд ярмарки .....	61
ВИСНОВКИ.....	67
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	68

## СПИСОК УМОВНИХ ТЕРМІНІВ ТА ПОЗНАЧЕНЬ

Фронтенд – все, що браузер може читати, виводити на екран і/або запускати. Тобто це HTML, CSS і JavaScript.

Бекенд – все, що працює на сервері, тобто «не в браузері» або на комп'ютері, підключеному до мережі (зазвичай до Інтернету), який відповідає на повідомлення від інших комп'ютерів.

Фреймворк – платформа для розробки програмного продукту. Ця платформа забезпечує основу, на якій розробники програмного забезпечення створюють додатки для певних платформ.

HTML (HyperText Markup Language) – це мова розмітки, яка використовується для розробки веб-сайтів та веб-програм.

CSS (Cascading Style Sheets) – мова дизайну, яка відповідає за візуальне оформлення веб сторінки.

Javascript – це динамічна мова програмування. Вона не є ресурсозатратною і найчастіше використовується як частина веб-сторінок, реалізації яких дозволяють клієнтському сценарію взаємодіяти з користувачем та створювати динамічні сторінки.

SQL (англ. Structured Query Language) – мова структурованих запитів, яка використовується для модифікації та управління даними.

## ВСТУП

Пошук роботи для студента завжди був важливим та відповідальним етапом в подальшому кар'єрному розвитку, незалежно від того чи це робота за спеціальністю чи сезонний заробіток. З іншого боку, роботодавці завжди відчують нестачу молодих, цілеспрямованих спеціалістів. Саме для цього керівництво компаній регулярно організовують так звані “дні кар'єри” або “ярмарки кар'єри”. Ці заходи спрямовані на популяризацію компаній для студентів, проведення семінарів, на яких висвітлюються основні напрями роботи, надання інформації про актуальні та затребувані професії, а також вакансії. Такі заходи є дуже важливими та корисними для обох сторін: керівництво компаній отримує актуальну інформацію про стан підготовки студентів та їх зацікавленість в отриманні робочих місць. У свою чергу студенти отримують актуальну інформацію про те, в якому напрямку потрібно розвиватись задля досягнення поставлених цілей.

Нині суспільство зіткнулося з проблемою глобального масштабу – пандемією корона вірусу. Тому по всьому світу були впровадженні карантинні заходи. Підприємства та різноманітні заклади змушені відмовитись від проведення подібних заходів задля уникнення масових скупчень людей та загроз, які можуть слідувати. З цією метою проведення віртуальних заходів в даний час є дуже актуальним.

**Актуальність.** Розробка віртуальної платформи «StudYar» в даний час є актуальною, бо не дивлячись на складну ситуацію у світі, пов'язану з коронавірусом та карантинними заходами, компанії та організації потребують молодих кваліфікованих спеціалістів, а ті, в свою чергу, достойне робоче місце.

**Проблема** даної кваліфікаційної роботи полягає у вивченні та розв'язанні наступних питань:

- Ретельне вивчення структури побудови систем віртуальних заходів.
- Визначення вимог до проекту.
- Розробка продукту згідно з отриманими даними.

**Тема.** «Використання інформаційних технологій для створення віртуальних ярмарок».

Сутність поставленого завдання полягає в розробці віртуальної платформи «Студентський ярмарок» (надалі будемо використовувати скорочену назву «StudYar»). Цей програмний продукт після виконання всіх етапів роботи повинен давати можливість проводити віртуальні заходи пошуку робіт студентам, мати простий та дружній інтерфейс, зрозумілий для опанування потенційним користувачем.

**Мета.** На підставі актуальності та проведених досліджень створити сучасну програмну систему з використанням веб-технологій для організації комунікації між роботодавцями та здобувачами освіти.

**Об'єкт.** Технологічний процес розробки віртуальної платформи «StudYar».

**Предмет.** Віртуальна платформа «StudYar» дозволяє створювати місця для проведення віртуальної ярмарки пошуку робіт, надання шляхів комунікації між студентом та компанією, інтерфейс передбачає можливість швидкого пошуку вакансій, їх зручне відображення, легкої подачі заявок на вакансію компанії, виведення повідомлення про успішну відправку, проведення вебінарів для студентів.

**Цілі роботи:**

1. Підібрати інструментарій для аналізу предметної області та з'ясування функціональних вимог та проектування програмного виробу.
2. Розробити матеріали, для використання у навчальному процесі.
3. Використовуючи підібрані технології розробки, спроектувати віртуальну платформу «StudYar».

**Завдання.** Для досягнення поставлених цілей потрібно:

- Розробити вимоги до програмного виробу.
- Спроектувати та розробити базу даних.
- Спроектувати та розробити інтерфейс користувача.

- Реалізувати засобами веб-технології, зокрема PHP Laravel Framework та VueJS, програмний виріб
- Навчитися оперувати елементами інтерфейсу через код програмної частини.

**Гіпотеза.** Практична реалізація віртуальної платформи спростить організацію подібних заходів та спростить пошук студентам у пошуку роботи, а для роботодавців кваліфікованих працівників.

**Апробація.**

- Звітна науково-практична конференція професорсько-викладацького складу Рівненського державного гуманітарного університету, яка відбулась 14-15 травня 2020 року, м. Рівне.
- XIII Всеукраїнська науково-практична конференція «Інформаційні технології в професійній діяльності» яка відбулась 18 листопада 2020 року.
- Науковий семінар кафедри інформатики та прикладної математики факультету математики та інформатики Рівненського державного гуманітарного університету, який відбувся 1 грудня 2020 року, м.Рівне.

**Обсяг роботи.** Дипломна робота складається зі вступу, трьох розділів, висновку та списку використаних джерел. Повний обсяг роботи складає 69 сторінок машинного тексту, в тому числі 42 рисунків.

## РОЗДІЛ 1. ОГЛЯД АНАЛОГІВ ПРОГРАМНОГО ПРОДУКТУ ТА ІНСТРУМЕНТАРІЮ СТВОРЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### 1.1. Огляд аналогів

На даному етапі розвитку інформаційних технології шляхи пошуку роботи для сучасного співробітника чи студента стрімко змінюються. Все менше і менше людей в наш час купує газети з оголошеннями робіт. Більшість намагається знайти роботу за допомогою мережі Інтернет. Серед розповсюджених ресурсів пошуку праці є такі відомі ресурси як **work.ua** або **rabota.ua**. Ці ресурси спеціалізуються на наданні вакансії якоїсь компанії з коротким описом компанії та описом вакансії. Іншим можливим способом пошуку роботи можуть бути ярмарки вакансій робочих місць. Це хороший спосіб не тільки довідатися про актуальні вакансії, але й почути про основні напрями розвитку компанії, проаналізувати ринок праці в цілому. Нині віртуальні ярмарки набирають все більшої популярності як серед користувачів, так і серед організаторів заходів. Це і не дивно, адже на це є кілька причин:

- Карантинні заходи пов'язані з пандемією корона вірусу. Велике скупчення людей є небезпечним у такий час.
- Висока гнучкість віртуальних ярмарок. Проведенням таких заходів можуть займатись люди, які не обов'язково мають досвід у проведенні офлайн заходів. Також штат організаторів зменшується в декілька разів.
- Для того, щоб відвідати ярмарку праці, не потрібно витратити кошти на переїзди до інших міст.
- За допомогою віртуальних ярмарок, відвідувачі мають змогу ознайомитись з набагато більшою кількістю наявних представників компаній, ніж, наприклад, на офлайн заходах.
- Віртуальний ярмарок може приймати набагато більше користувачів, ніж традиційний захід. Це серйозний аргумент для переконання керівництва

компаній в його актуальності, адже величезна аудиторія людей зможе ознайомитись з діяльністю компанії.

- Віртуальна платформа значно заощаджує час на організацію проведення, пошук найкращого місця проведення. Сучасний віртуальний ярмарок може бути створений за лічені хвилини, коли на створення офлайн ярмарку можуть піти тижні.
- Використовуючи функціонал продукту адміністрація та менеджмент заходу має змогу збирати різноманітну інформацію про майбутній ярмарок або той, який вже відбувся.

Як і на справжніх ярмарках, після реєстрації кожен відвідувач має змогу перейти на стенд та ознайомитись з інформацією про компанію (зразки продукції, макети, фото, відео), поспілкуватись з менеджером компанії за допомогою чату, подати резюме на вакансію, яка цікавить відвідувача, або залишити вакансію у розділі “вибране”. Після подання резюме, відвідувач та менеджер компанії отримують лист про успішну відправку.

Розглянемо декілька існуючих програм для створення віртуальних ярмарок.

### **vFairs**

Однією з основних платформ для проведення віртуальних заходів є платформа під назвою **vFairs**. Вперше платформа провела віртуальний захід у 2016 році та в подальшому набула великої популярності серед організаторів. Ця платформа була першою в даному класі платформ.

З кожним роком функціонал розширювався, але, не дивлячись на немалий досвід роботи компанії, на даний момент часу програмний продукт має певні недоліки:

- Доволі дорогу ціну проведення віртуального заходу.
- Застарілий інтерфейс користувача.
- Не логічний інтерфейс адміністраторської частини платформи.



## Eventcube

Наступною популярною платформою для проведення заходів є Eventcube. Дана система має одну відмінність від інших платформ. Вона включає систему квитків, яка має схожу логіку з сайтом “booking.com”. Також платформа дає змогу проводити конференції які можна транслювати в інших соціальних мережах. На даний момент програмний продукт має свої недоліки:

- Дороге обслуговування
- Не до кінця продуманий інтерфейс користувача

## Svent

Ще однією платформою для проведення таких заходів є Svent. Вона має потужний функціонал для mail маркетингу широкий спектр наявних вбудованих статистик. Ключовою відмінністю цієї системи є адміністративна панель для менеджера офлайн заходів, тобто цю систему можна використовувати поза мережею Інтернет. До недоліків можна віднести відсутність інтеграції систем для проведення вебінарів.

## 1.2. Огляд інструментарію

Перед початком огляду основного інструментарію, треба розглянути поняття **front-end, back-end, framework**.

Фронтенд – все, що браузер може читати, виводити на екран і/або запускати. Тобто це HTML, CSS і JavaScript.

Бекенд – все, що працює на сервері, тобто «не в браузері» або на комп'ютері, підключеному до мережі (зазвичай до Інтернету), який відповідає на повідомлення від інших комп'ютерів.

Різницю між бекендом та фронтендом можна пояснити на прикладі годинника. Циферблат, стрілки, цифри, механізм управління стрілками – все те, що бачить користувач – це є фронтенд. Ми бачимо конкретний результат – котра година, ми можемо цей час підкоригувати. Але от як працюють шестерні,

як працює сам механізм, процесор чи пружинка користувач не бачить. Це і є бекенд (рис.1.2.1 ).

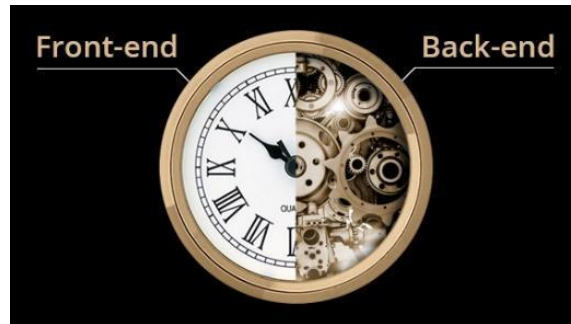


Рис. 1.2.1. Принцип роботи бекенду та фронтенду на прикладі годинника

Front end – це візуальна частина web-додатку, з якою користувач може взаємодіяти напряму. До цієї частини відноситься відображення функціональних задач, інтерфейсу користувача, який виконується на стороні клієнта, а також обробка запитів користувача. Тобто це все те, що бачить користувач при відкритті web-сторінки.

Ця частина відповідає за правильне розташування всіх елементів на сторінці, цілісний вигляд сторінки, т. т. кожен елемент цієї частини на сторінці виконував поставлені задачі (наприклад, при натисканні кнопки “далі” користувача перенаправляло на наступну сторінку)

Основним інструментами розробки front-end є HTML, CSS, JavaScript. Кожен з них відіграє важливу роль в роботі веб-сторінки [1].

HTML (HyperText Markup Language)– це мова розмітки, яка використовується для розробки веб-сайтів та веб-програм. Зацікавлені сторони бізнесу, керівництво проектами та розробники програм надають перевагу саме HTML перед іншими альтернативними варіантами розробки програм через вигідні характеристики HTML.

Оскільки HTML є мовою розмітки, тому вона використовує прості теги для розмітки та форматування вмісту. Ці теги обмежуються кутовими дужками, наприклад тег <html>. Майже кожен тег має свій закриваючий тег. Таким тегом буде </html>. Відкриваючий тег повідомляє браузеру про те, що документ

HTML було запущено, а закриваючий – про закінчення документа. Все, що буде знаходитись всередині цих двох тегів, буде відображатись в браузері.

HTML – один з найкращих варіантів розробки веб-сторінки, або веб-сайту для малого та середнього бізнесу, який не має змоги інвестувати більше в придбання програмного забезпечення, або в ліцензію і не вимагає розширеного програмування для своїх веб-сайтів.

Переваги HTML:

- HTML легко вивчати та використовувати. Для тієї людини, яка вивчає веб-розробку, HTML є першою і найголовнішою мовою, з якою вона знайомиться. Вона має прості теги, немає великої чутливості до регістру. Можна легко зрозуміти чужий код і, якщо потрібно, внести в нього зміни.
- HTML є безкоштовною. Однією з найбільших переваг HTML є те, що вона безкоштовна і немає потреби купувати платне програмне забезпечення. Не має потреби використовувати сторонні плагіни для роботи з ним. Її також легко редагувати, бо немає потреби у спеціальному інтерфейсі, або платформі для її редагування. Код HTML можна редагувати навіть у простому Блокноті або, наприклад в більш просунутому Notepad++.
- HTML підтримується всіма браузерами. Майже всі браузери по всьому світу використовують HTML. Тож не потрібно турбуватись про продукт, написаний у форматі HTML, бо він легко з'явиться у всіх браузерах, якщо розробник оптимізував програму під всі браузери. HTML надає легкі способи оптимізації, відповідно до браузерів.
- HTML є однією з найзручніших пошукових систем у порівняно зі всіма мовами програмування, доступними на ринку. Веб сайти, написані з HTML, легше читати та отримувати до них доступ веб-сканерам, отже, покращує його ефективність
- Можна легко інтегрувати з іншими мовами. Наприклад в JavaScript, NodeJS, PHP та багатьох інших мовах ми пишемо код цих мов між HTML.

- Одразу виводить зміни. Однією з найбільших переваг HTML є те, що ми можемо миттєво бачити зміни, просто зберігаючи їх та перезавантажуючи попередню сторінку HTML. На відміну від інших мов програмування, немає потреби запускати весь код і з'ясувати де помилка. Наприклад, якщо ви зробили слово курсивом, то воно одразу з'явиться на сторінці після збереження та перезавантаження[2].

CSS (Cascading Style Sheets) – мова дизайну, яка відповідає за візуальне оформлення веб сторінки. Це документ, в якому описується як буде виглядати кожен з елементів сторінки. Перевагами використання CSS є:

- Значне заощадження часу. При великому числі наявних сторінок на сайті, їх дизайн можна змінити лише змінивши стиль в одному файлі і при потреба всі сторінки будуть посилатися на один CSS-файл.
- Допомагає підвищити швидкість завантаження сторінки. Використання CSS сприяє зменшенню обсягу коду, що, в свою чергу, пришвидшує завантаження сторінки браузером.
- Робить сторінку респонсивною (легко підлаштовуваною під екрани з різними розмірами). Все частіше для перегляду веб-сторінок використовують смартфони, планшети та інші гаджети з екранами, які відрізняються від стандартного екрана монітору. При розробці сторінки на це потрібно зважати. CSS полегшує розробку[3].

Javascript – це динамічна мова програмування. Вона не є ресурсозатратною і найчастіше використовується як частина веб-сторінок, реалізації яких дозволяють клієнтському сценарію взаємодіяти з користувачем та створювати динамічні сторінки. Ця інтерпретована мова програмування включає в себе об'єктно-орієнтовані можливості.

Javascript спочатку була відома як LiveScript, але, Netscape, компанія-розробник мови, змінила назву на Javascript. Вперше Javascript з'явилася в

Netscape 2.0 у 1995. Ядро мови загального призначення було вбудовано в Netscape, Internet Explorer та інші веб-браузери.

Javascript клієнтської частини – найпоширеніша форма мови. Скрипт повинен бути включений в документ HTML, або мати посилання на нього, щоб код інтерпретувався браузером. Це означає, що сторінка не повинна бути статичним HTML, а може включати програми, які взаємодіють з користувачем, керують браузером та динамічно створюють вміст HTML.

Будучи мовою сценаріїв, JavaScript не може працювати самостійно. Насправді браузер відповідає за запуск коду JavaScript. Коли користувач дає запит на HTML-сторінку з JavaScript, скрипт надсилається на браузер і браузер сам повинен його виконати. Головною перевагою є те, що всі сучасні веб-браузери підтримують Javascript. Крім того, JavaScript працює на будь-якій операційній системі, включаючи Windows, Linux, Mac. Отже, JavaScript усуває основні недоліки наразі застарілого VBScript, який має обмежене використання – лише Internet Explorer та Windows.

Клієнтський механізм Javascript надає багато переваг перед традиційним CGI скриптом. Наприклад, можна використовувати JavaScript, з метою перевірки вводу користувачем правильної адреси електронної пошти у поле форми. Код Javascript виконується, коли користувач надає форму, і лише якщо всі записи є дійсними, вони будуть надіслані на веб-сервер.

JavaScript може використовуватись для захоплення ініційованих користувачем подій, таких як натискання на кнопку, навігація посиланнями на інші події, які користувач ініціював явно або неявно.

Інструментами для роботи з JavaScript можуть стати будь-які текстові редактори для написання коду та браузер для відображення розроблених веб-сторінок. Такими текстовими редакторами є NotePad++, Visual Studio Code, Sublime Text, Atom.

Переваги Javascript:

- Мінімізація взаємодії з сервером. За допомогою Javascript ми маємо змогу перевіряти введення даних користувача, перш ніж відправляти сторінку

на сервер. Це економить серверний трафік, а це означає менше навантаження на наш сервер.

- Швидкий відгук відвідувачам – їм не потрібно чекати, поки сторінка перезавантажиться, щоб побачити, чи не забули вони щось ввести, наприклад в полях для реєстрації.
- Підвищена інтерактивність. За допомогою Javascript ми маємо змогу створювати інтерфейси, які реагують, коли користувач наводить на них курсор, або активує за допомогою клавіатури.
- Великий інструментарій для розширення можливостей інтерфейсу. Ми можемо використовувати Javascript, щоб включити такі елементи, як компоненти перетягування та повзунки, щоб надати розширений інтерфейс відвідувачам нашого сайту [4].

Наступним поняттям для ознайомлення є поняття фреймворку.

Фреймворк – платформа для розробки програмного продукту. Ця платформа забезпечує основу, на якій розробники програмного забезпечення створюють додатки для певних платформ. Наприклад, фреймворк може включати заздалегідь визначені класи та функції, які можна використовувати для обробки введення, управління апаратними пристроями та взаємодії з програмним забезпеченням системи [5].

Фреймворки полегшують життя програмістам, дозволяючи їм контролювати весь процес розробки лише з однієї платформи.

Переваги використання фреймворків:

- Код стає безпечнішим.
- Уникнення повторного та надлишкового коду.
- Допомагає послідовно розробляти код з меншим числом помилок.
- Полегшує роботу зі складними технологіями.
- Функціональні можливості та сегменти коду були протестовані раніше.  
Тож це робить фреймворк більш надійним інструментом
- Значне скорочення часу

Коли відбувається інсталяція фреймворка, перше, про що потрібно подбати - це системні вимоги. Після того, як фреймворк встановлений і налаштований, він створює структуру каталогів. Наприклад на рисунку показано структуру каталогів Laravel Framework (рис. 1.2.2). Кожна з цих папок може мати додаткові файли, класи, процедури, шаблони, тощо.

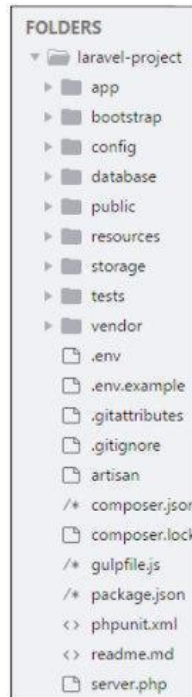


Рис.1.2.2. Структура каталогів Laravel Framework[6]

При розробці нашого програмного продукту основними фреймворками фронтенд частини є Bootstrap та VueJs [7].

Bootstrap – це відкритий та безкоштовний фреймворк для HTML, CSS та JS, який використовується для створення адаптивного веб-дизайну швидко та ефективно. Основною сферою застосування є розробка компонентних інтерфейсів сайтів та інтерфейсів адміністратора. Серед подібних систем (Foundation, UIKit, SemanticUI, InK) Bootstrap є найпопулярнішою.

Bootstrap – це сукупність файлів CSS та JavaScript. Коли веб-розробник підключить ці файли до сторінки, велике число класів та готових компонентів стане доступним до макету дизайну.

Класи Bootstrap можуть бути розділені на 3 частини:

- Класи для створення компонента Grid (адаптивна верстка сторінки).
- Класи для стилів вмісту (тексту, таблиць та іншої інформації).
- Сервісні класи (для вирішенні найпоширеніших допоміжних проблем, таких як вирівнювання, контроль дисплею, додавання меж тощо) [8]

Система сіток Bootstrap дозволяє розділити рядок на 12 стовпців однакової ширини (рис. 1.2.3). Однак, ці стовпці можна комбінувати для створення ширших стовпців та різного позиціонування. Сітка також може реагувати та змінювати порядок залежно від ширини екрана та розміру вікна.

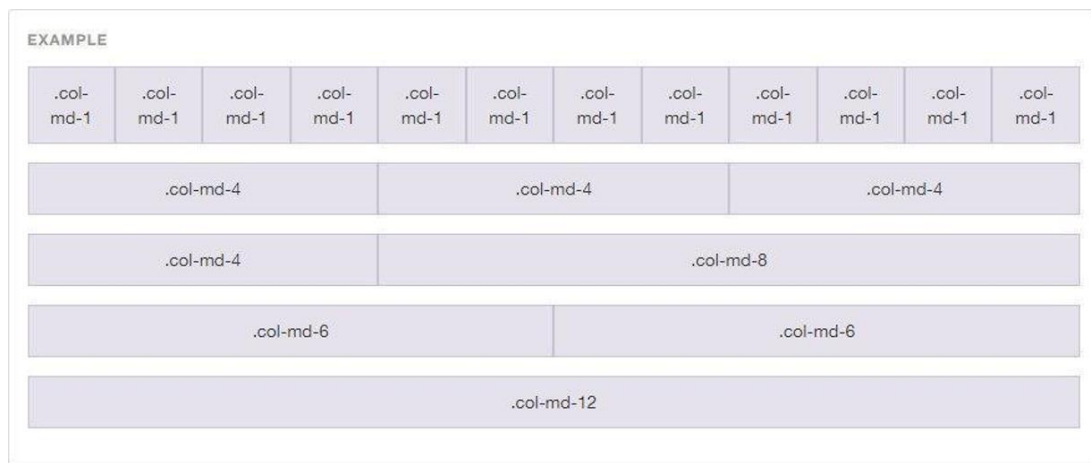


Рис. 1.2.3. Структура сітки Bootstrap Grid[9]

У Bootstrap наявні п'ять класів для визначення адаптивності елемента Grid (рис. 1.2.4 ).

- .col- \* - для екранів, вужчих ніж 576 px.
- .col-sm- \* -для екранів ширших 576 px.
- .col-md- \* - для екранів ширших 768 px.
- .col-lg- \* - для екранів ширших 992 px.
- .col-xl- \* - для екранів ширших 1200 px [10].

Переваги використання Bootstrap:

- Кросбраузерність та адаптивність. Веб-сторінки, створені за допомогою цього фреймворка ідентично відображаються на всіх сучасних пристроях.
- Великий обсяг документації. Документація доволі детальна, містить ілюстрації, які допомагають краще зрозуміти фреймворк.



- Має в собі модулі JavaScript, які спрощують розробку та значно економлять час [12].

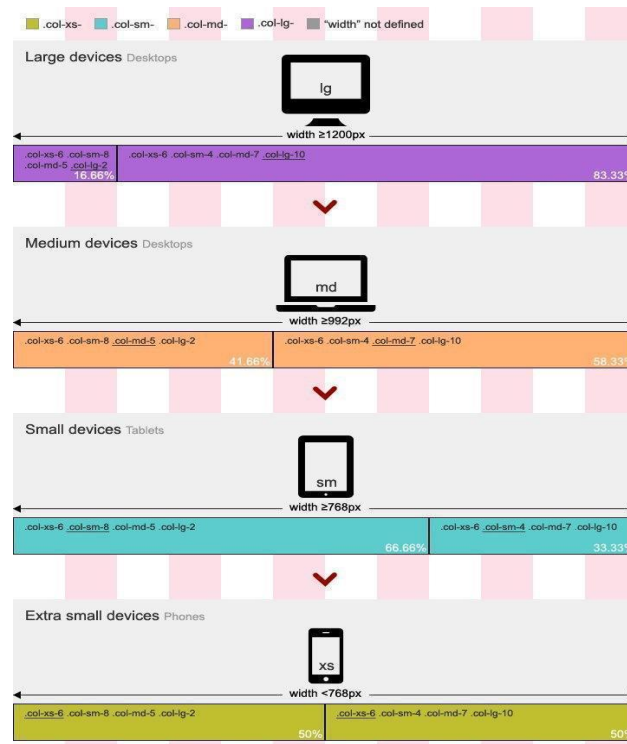


Рис. 1.2.4 Наявні класи Grid в Bootstrap [11]

VueJS – прогресивний фреймворк JavaScript для побудови веб-інтерфейсу, який останнім часом набув великої популярності. На відміну від багатьох інших фреймворків, VueJS можна використовувати для вдосконалення існуючого HTML. Це дозволяє використовувати VueJS як заміник такої бібліотеки як jQuery.

До початку 2000-х браузері не мали таких можливостей, як зараз. Вони були менш потужними, побудова складних додатків всередині них була неможлива з погляду продуктивності. Все змінилось, коли Google надали Google Maps і Gail, дві програми, які працювали всередині браузера.

Бібліотеки як jQuery та MooTools були першими великими проектами, які базувались на JavaScript і були дуже популярними. Вони надавали якісніший API для взаємодій з браузерами та надавали способи рішень помилок невідповідностей між різними браузерами.

Такі фреймворки як Backbone, Ember, Knockout, AngularJs були першою хвилею сучасних фреймворків JavaScript. Друга хвиля, яка є поточною, включає в себе такі фреймворки як React, Angular, VueJS.

VueJS називають прогресивним фреймворком. Це означає, що він адаптується до потреб розробника. В той час як інші фреймворки вимагають певного вкладу від розробника або команди, VueJS легко розгортається за допомогою просто теги сценарію [13].

Переваги VueJS перед іншими фреймворками:

- Малий розмір фреймворка. Завантажений zip-файл із фреймворком має розмір 18 КБ. Перевагою такого фреймворка є не тільки швидке завантаження та встановлення бібліотеки, це також позитивно впливає на SEO оптимізацію.
- Віртуальний DOM. Модель об'єкта документа (DOM) – це представлення HTML-сторінок з їх стилями, елементами та вмістом сторінки як об'єктами. Об'єкти, які зберігаються як деревоподібна структура, створюються браузером під час завантаження сторінки(рис. 1.2.5).

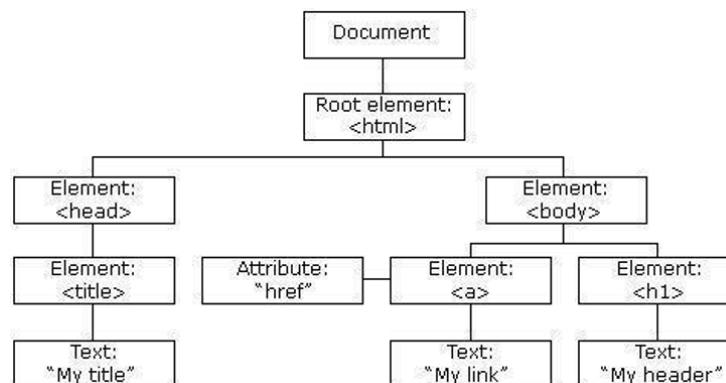


Рис. 1.2.5. Схема віртуального DOM [14]

- Однофайлові компоненти та зручність читання. Кожна частина програми/веб-сторінки у VueJS є компонентом. Компоненти представляють інкапсульовані елементи інтерфейсу. У VueJS компоненти можна писати у форматі HTML, CSS та JavaScript, не розділяючи їх на

окремі файли. Розбиття коду програми насправді є архітектурним підходом, який називається Component Based Architecture (CBA)

- Інтеграційні можливості та гнучкість. Важливим аспектом будь-якої нової технології є можливість інтеграції з існуючими додатками. Так як фреймворк Vue.js покладається лише на JavaScript і не потребує жодних інших інструментів для роботи, він легкий для інтеграції. Vue.js також дозволяє писати шаблони за допомогою HTML, JS або JSX [15].

Тепер перейдемо до огляду бекенд технологій. Для початку з'ясуємо, що означає термін бекенд.

Бекенд – поєднання технологій та програмування, яке забезпечує веб-сайт. Він складається з трьох частин, які користувач ніколи не бачить: сервера, програм та бази даних. Бек-енд розробник – це той, хто створює та підтримує технологію, потрібну для роботи компонентів, які дозволяють існувати стороні веб-сайту, яка спрямована на користувача. Їх внутрішній код додає ефективності до всього, що створює розробник інтерфейсу. Такі розробники несуть відповідальність за створення, підтримку, тестування та налагодження цілого інтерфейсу. Це включає основну логіку додатків, бази даних, інтеграцію даних, API та інші фонові процеси. Це поєднання бази даних та програмного забезпечення, написаного мовою на сторінці сервера, яке запускається на веб-серверах, хмарних сервісах.

Веб-сайт працює за рахунок бекенду – користувач не бачить його та не взаємодіє з ним, адже він завжди працює у фоновому режимі, забезпечуючи функціональність та роботу [16].

Бекенд фреймворк який використовувався при створенні нашого додатку є Laravel.

Laravel – це PHP фреймворк, який побудований на архітектурі MVC (Model-View-Controller). Був створений, щоб замінити тоді вже популярний фреймворк CodeIgniter. На той момент цей фреймворк не пропонував таких функцій як підтримка вбудованої аутентифікації клієнтів та правильної

авторизації користувачів. Головною ціллю даного фреймворка є підвищення якості коду, зменшення вартості розробки веб-сайту. Фреймворк складається з пакетів, міграцій, та Artisan CLI (інтерфейс командного рядка). Цей фреймворк надає заздалегідь визначені куски коду, розроблені за стандартом останніх практик програмування [17].

Особливості Laravel:

- Модульність. Laravel надає 20 вбудованих бібліотек та модулів, які допомагають вдосконалити програму. Кожен модуль інтегрований з менеджером залежностей Composer, який полегшує оновлення.
- Стабільність. Laravel включає функції та помічники, які допомагають у тестуванні на різних тестових кейсах. Ця функція допомагає підтримувати код відповідно до вимог.
- Маршрутизація. Laravel забезпечує гнучкий підхід до користувача для визначення маршрутів у веб-програмі. Маршрутизація допомагає краще масштабувати та підвищує її продуктивність.
- Управління конфігурацією. Веб-програма, розроблена на Laravel, буде працювати в різних середовищах, а це означає що її конфігурація буде постійно змінюватися. Фреймворк забезпечує послідовний підхід до ефективного управління конфігурацією.
- Конструктор запитів ORM. Laravel включає конструктор запитів, який допомагає робити запити до бази даних за допомогою різних простих ланцюгових методів. Він забезпечує реалізацію ORM (Object relational Mapper) та ActiveRecord під назвою Eloquent.
- Конструктор схем. Schema Builder підтримує визначення бази даних та схему у PHP коді. Він також підтримує відстеження змін щодо міграції баз даних
- Template Engine. Laravel використовує механізм Blade Template, полегшену мову шаблонів, яка використовується для проектування ієрархічних блоків та макетів із заздалегідь визначеними блоками, які включають динамічний вміст.

- Аутентифікація. Аутентифікація користувачів – загальна функція веб-додатків. Laravel полегшує розробку аутентифікації, бо включає такі функції, як реєстрація, нагадування забутого паролю.
- Черги. Laravel включає сервіси черги, такі як надсилання електронної пошти великому числу користувачів, або визначений cron job. Ці черги допомагають легше виконувати завдання, не чекаючи завершення попереднього.

Laravel надає наступні переваги:

- Веб-додаток стає більш масштабованим завдяки фреймворку. При розробці значно економиться час, бо Laravel повторно використовує компоненти з інших середовищ для розробки. Він включає простори імен та інтерфейси, які допомагають організувати ресурси та керувати ними.
- Composer – це інструмент, який включає всі залежності та бібліотеки. Це дозволяє користувачеві створювати проект щодо даного фреймворка. Сторонні бібліотеки також можна легко встановити за допомогою composer.
- Artisan – це інтерфейс командного рядка, який використовується в Laravel. Він включає в себе набір команд, які допомагають створювати веб-додаток. Ці команди інтегровані з фреймворка Symfony, що приводить до додаткових функцій у Laravel [18].

## РОЗДІЛ 2. МОДЕЛЮВАННЯ ПРОГРАМНОГО ПРОДУКТУ

### 2.1. UX-проектування

Перед тим як приступати до розробки програмного продукту, досить важливо сконструювати прототип програмного продукту – його інтерактивну модель. Він дозволяє впевнитись, що розробник та замовник розуміють потреби замовлення, та краще зрозуміти потреба користувачів та клієнта. Побудова прототипу, або, так званого wireframe є дуже важливою, адже на початкових етапах під час його розробки легше внести зміни ніж у вже в готовий продукт.

Розробка прототипу розпочинається з обговорення гіпотез про сам продукт, оскільки саме їх потрібно буде перевіряти на наступних стадіях розробки.

1. Яка цільова аудиторія додатку?

При обговоренні цієї гіпотези нас цікавить: вік, стать, соціальний статус;

2. Чим додаток стане корисним для користувача?

Відповідь на це питання допоможе зрозуміти чим в даний час незадоволений користувач щоб задовольнити його потреби.

Після того як ми довідалися більше про користувачів додатку, ми маємо продумати сценарії користувача – кроки, які робить користувач на шляху до своєї цілі. Для того, щоб виявити ці сценарії, та побачити можливі проблеми на шляху до цілі треба провести воркшоп – процедуру, яка триває близько 1,5 години та в яку включається вся команда розробників, разом з менеджером. Під час цієї процедури команда дотримується чіткого плану:

1. Виявлення цілей додатку. На протязі 15 хвилин команда проводить так званий “мозковий штурм”. В кінці цього кроку фіксується основна ідея додатку, не більше 140-150 символів.

2. Обмеження. На цьому етапі впродовж 15 хвилин команда намагається виявити та задокументувати всі можливі обмеження, які можуть завадити продукту стати успішним.
3. Опис цільової аудиторії. На протязі цього виділеного часу ми описуємо 1-3 персони, які будуть користуватись додатком.
4. Створюємо Story Map (рис. 2.1.1).

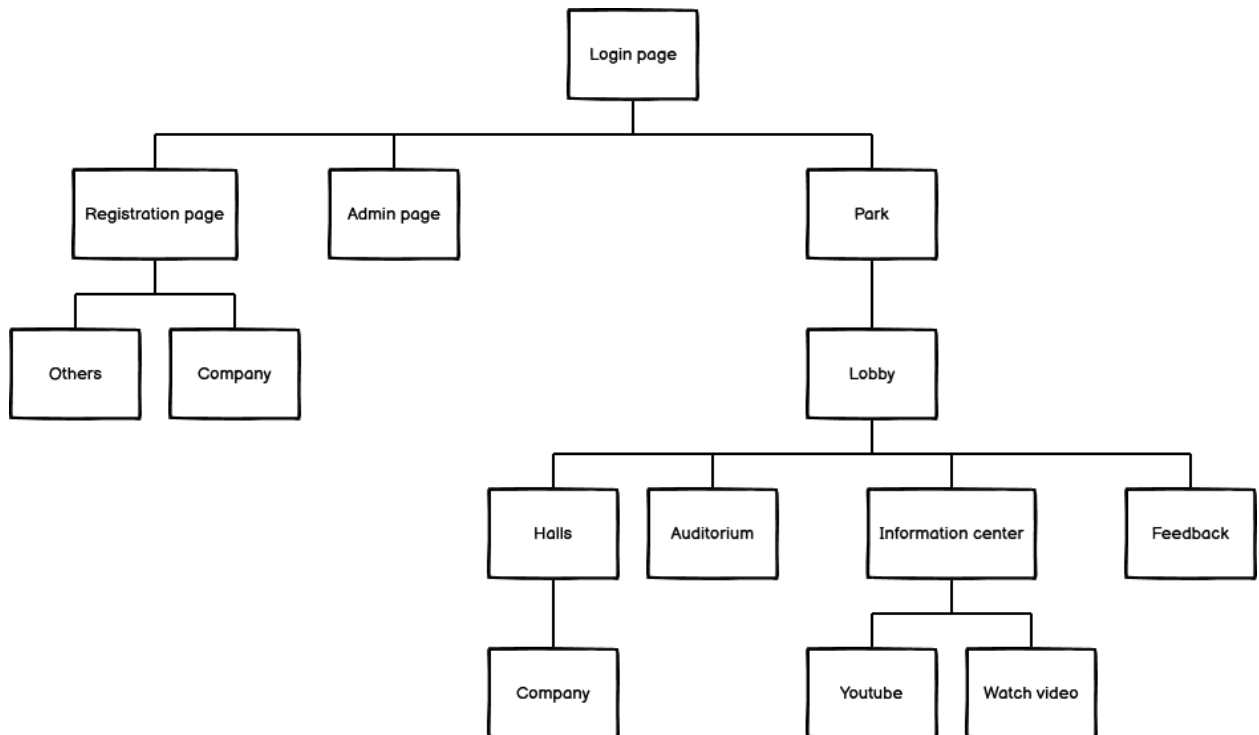


Рис. 2.1.1. Приклад Story Map платформи

Ця карта описує набір кроків, які повинен пройти користувач до досягнення поставленої цілі[19].

## 2.2. UI-дизайн

Наступним кроком при розробці програмного продукту є розробка UI дизайну (User Interface). Для цього спочатку треба спроектувати wireframe – наглядний ескіз розташування всіх елементів на сторінках сайту. Для цього на звичайному папері, або з використанням спеціальних інструментів команда або

дизайнер при мозковому штурмі розташовує всі елементи без особливих позначень (рис. 2.2.1). Це допомагає наповнити сайт елементами та не враховувати стилі та наповнення елементів.

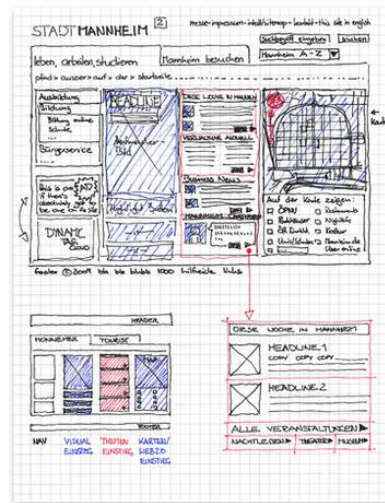


Рис. 2.2.1. Приклад розташування елементів на wireframe [20]

Далі wireframe переноситься у програму для створення прототипів. Для створення готового прототипу дизайнер переносить вміст wireframe з усіма потрібними розмірами та тою ж структурою без реальних картинок та тексту. Кольорова гама прототипу витримана у темно та світло-сірих кольорах задля фокусу на розташуванні елементів та оцінки логіки структури сторінки (Рис. 2.2.2).



Рис. 2.2.2. Приклад прототипу сторінки[21]



Існують різні інструменти для створення прототипів. Найпопулярнішими є Moqups, Axure RP, Mockflow та Balsamiq. Серед них найпростішою є програма Balsamiq. Вона є найпростішою, включає великий набір готових шаблонів та макетів елементів, які найчастіше використовуються при розробці сторінки сайту(рис. 2.1.4).

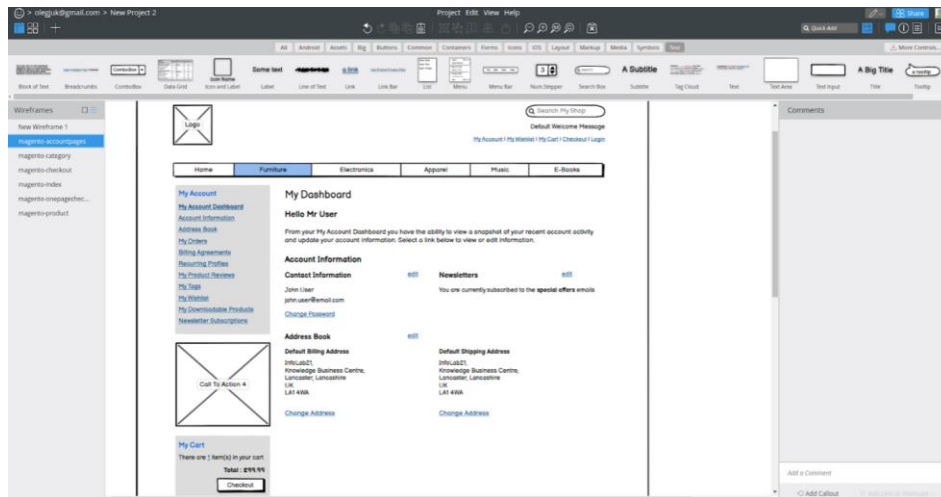


Рис. 2.2.3. Приклад одного з прототипів адміністративної панелі сайта

Далі за допомогою інструменту для розробки інтерфейсів під назвою Figma було створено точну копію прототипу дизайну але вже з врахуванням усіх кольорів та попередньо поставлених вимог до сторінок (розташування елементів на заданих місцях, точні розміри у пікселях, як були затвержені під час побудови прототипу) (Рис. 2.2.4).

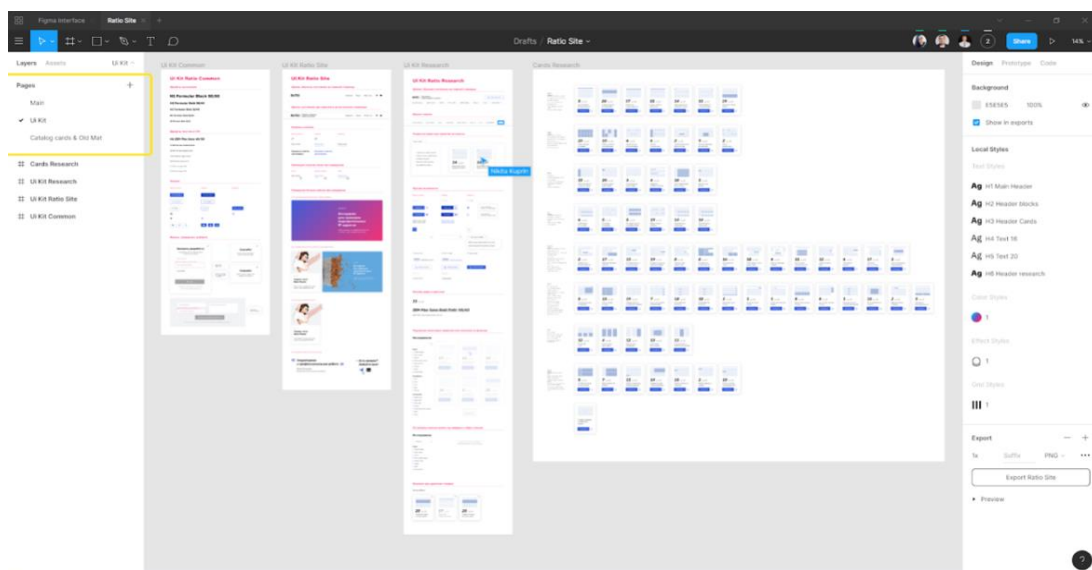


Рис. 2.2.4. Приклад розгорнутого проекту у програмі Figma[22]

Після створення всіх головних сторінок сайту, весь матеріал зберігається на ресурсі з назвою Zerpin. Це інструмент для зберігання та розповсюдження готових дизайнів між членами команди (Рис. 2.2.5). Головною особливістю цього інструмента є легкий перегляд всіх розмірностей кожного елемента на сторінці. Це набагато пришвидшує подальшу розробку, особливо верстку сторінки, адже вся потрібна інформація про розмірність та розташування елементів відносно інших елементів вже є наявною. Для отримання детальної інформації про якийсь конкретний елемент треба натиснути на нього курсором і ми можемо побачити всю інформацію про нього (Рис. 2.2.6).

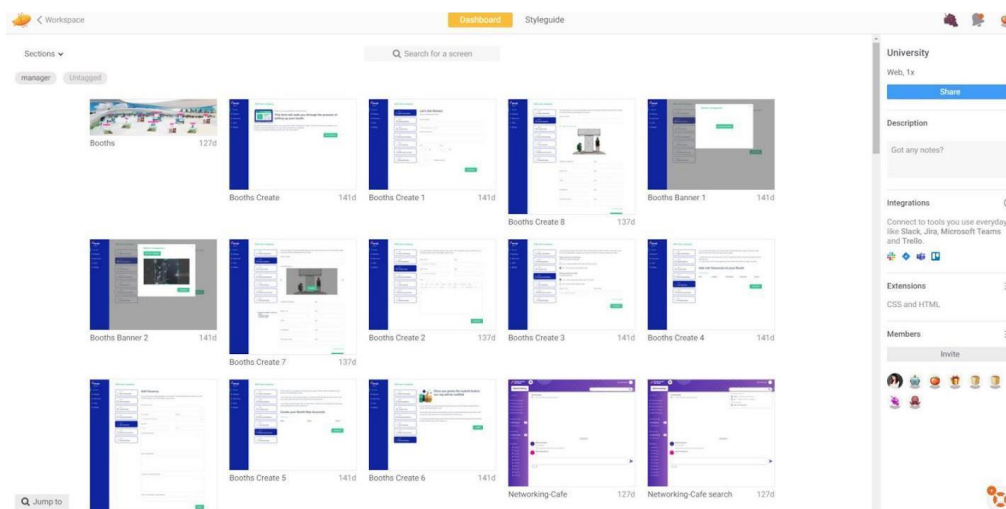


Рис. 2.2.5 Зовнішній вигляд інструменту Zerpin та вміст проекту в ньому

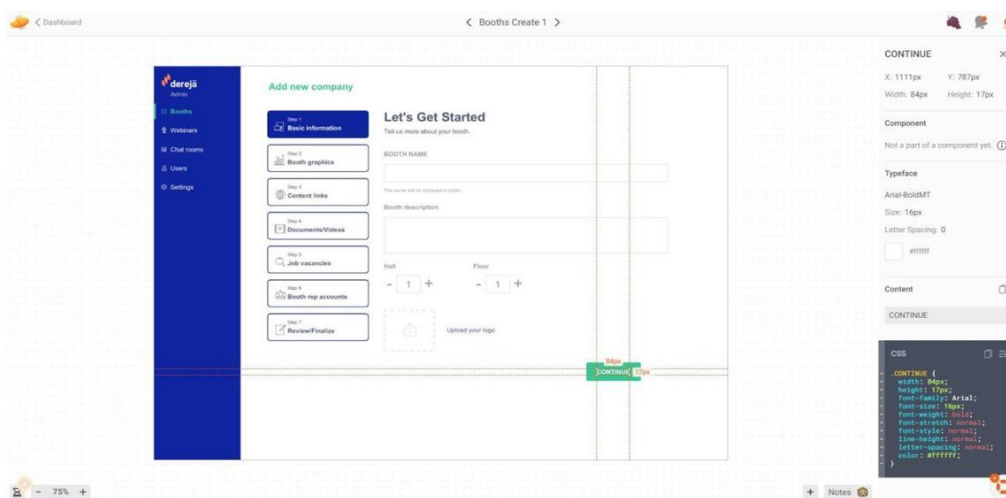


Рис. 2.2.6. Приклад дизайну сторінки сайту з наявною інформацією про кожен елемент

Окремим пунктом у розробці дизайну є розробка 3D моделей для оформлення сторінок з місцем проведення ярмарки. Дизайн будівель, холів та інших елементів був розроблений за допомогою інструменту 3D Max (рис.2.2.7, рис.2.2.8, рис.2.2.9.). Він є найпопулярнішим інструментом 3D дизайнерів, адже має потужні можливості для розробки моделей.

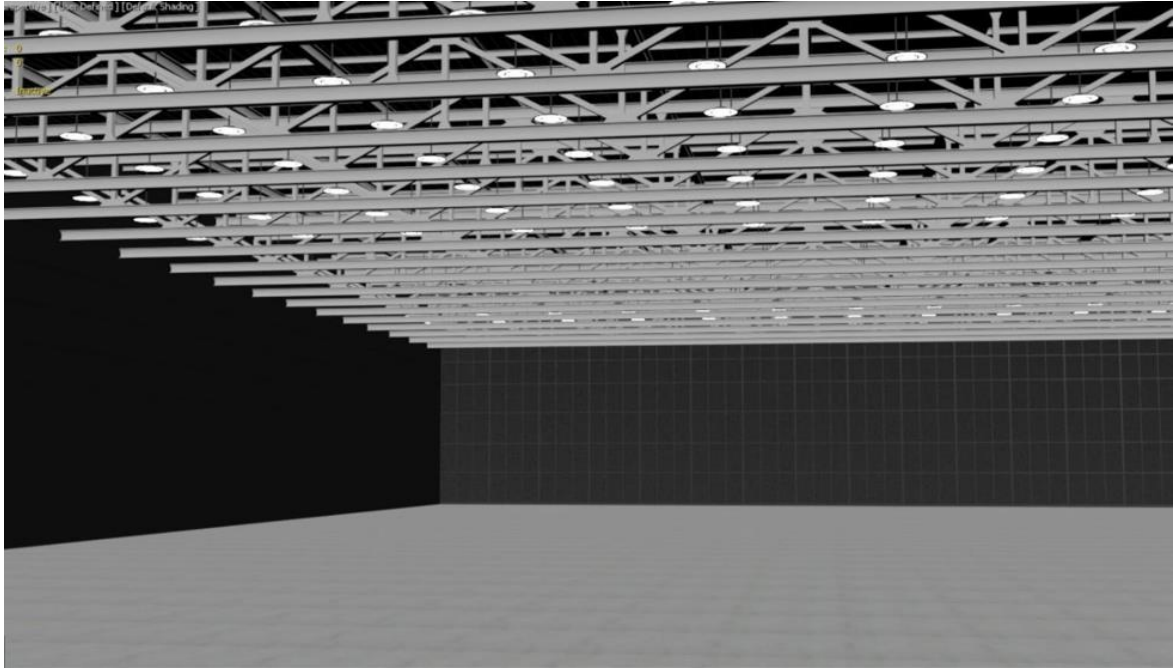


Рис. 2.2.7. Приклад розробки холу в програмі 3D Max

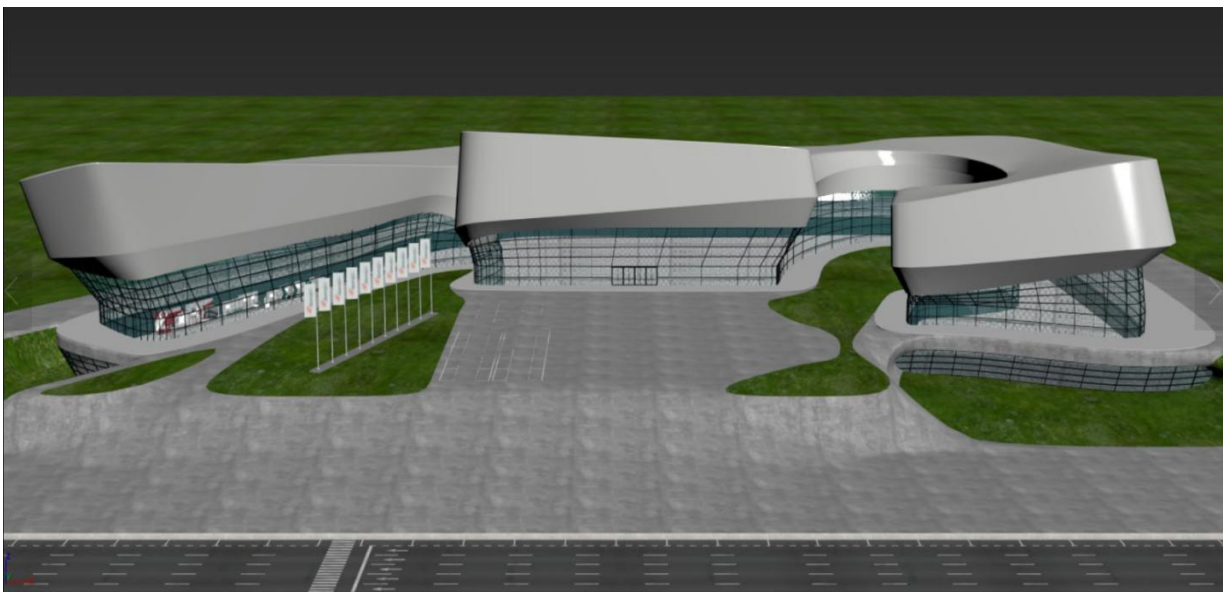


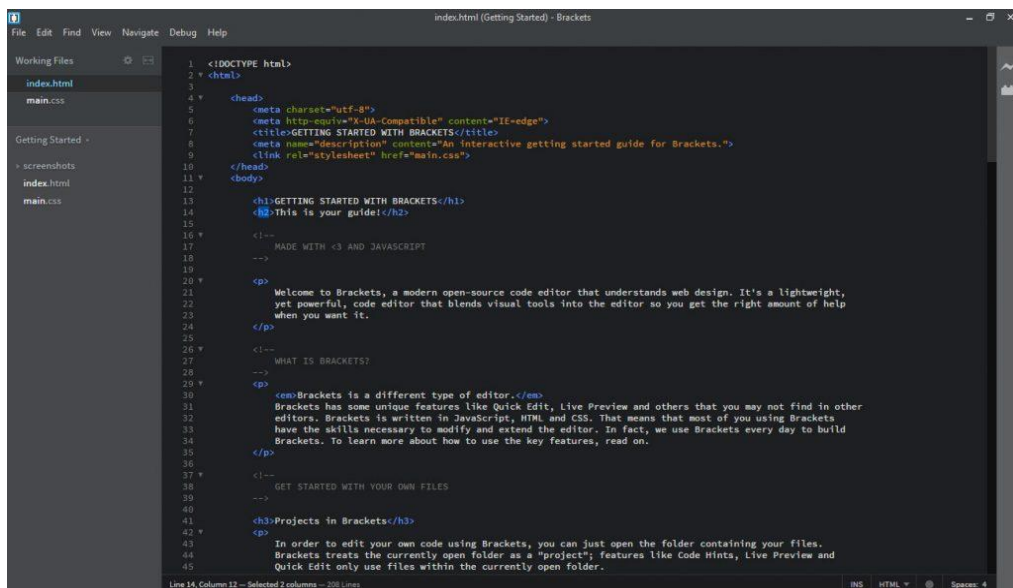
Рис. 2.2.8. Приклад розробки головної будівлі в програмі 3D Max



Рис.2.2.9. Приклад готової головної будівлі в програмі 3D Max

### 2.3. Верстка сторінок

Наступним кроком в розробці сайту була верстка сторінок з використанням технологій HTML5+CSS3 та фреймворка Bootstrap 4. Саме цей крок дає нам кінцевий результат того як сторінки будуть виглядати в браузері. Для верстки використовувався інструмент Brackets, так як саме він розрахований на версту веб-сторінок та пришвидшує цю роботу (рис. 2.3.1).



Також ми тестуємо сторінки в різних браузерах і пристроях (як мобільних, так і десктопних) та виправляємо недоліки. Далі переходимо до створення динамічної версії.

Для верстки адміністративної панелі платформи використовувався AdminLTE – шаблон для фреймворка Bootstrap 4. Використання цього шаблону обумовлено тим, що цей шаблон є готовим скелетом адміністративної частини, що значно пришвидшує розробку, адже враховує вже всі нюанси валідності, SEO оптимізації, респонсивності та виглядає так, як звично виглядає адміністративна панель.

В кінці був проведений аналіз верстки на відповідність стандартам, респонсивності, перевірка графіки (чи всі картинки стиснуті та знаходяться в правильних форматах ), код протабульований та сторінки функціональні при вимкнених картинках та вимкненому js коді.

## **2.4. Розробка бази даних**

Першим кроком розробки бази даних є створення її архітектури (Рис.2.4.1). Для цього кожен учасник розробки повинен врахувати всі нюанси взаємозалежностей в записах таблиць бази даних. Далі потрібно створити UML діаграму за допомогою інструмента для створення блок схем. Серед таких інструментів можна відмітити draw.io, LucidChart та Microsoft Office Visio. Для наших цілей був використаний draw.io, який має найпростіший і найзручніший інтерфейс.

При розробці бази даних була використана технологія Eloquent ORM (Object-Relation mapper)

ORM – це назва, яка описує ряд фреймворків, які дозволяють автоматично зв'язати базу даних з кодом. Такі фреймворки мають на меті якомога більше приховати сутність класичної бази даних, наскільки це можливо. Взамін розробнику надається зручний інтерфейс роботи та



маніпулювання даними. Наприклад, замість створення SQL запитів, програміст викликає звичайні методи, а всю іншу роботу бере на себе ORM.

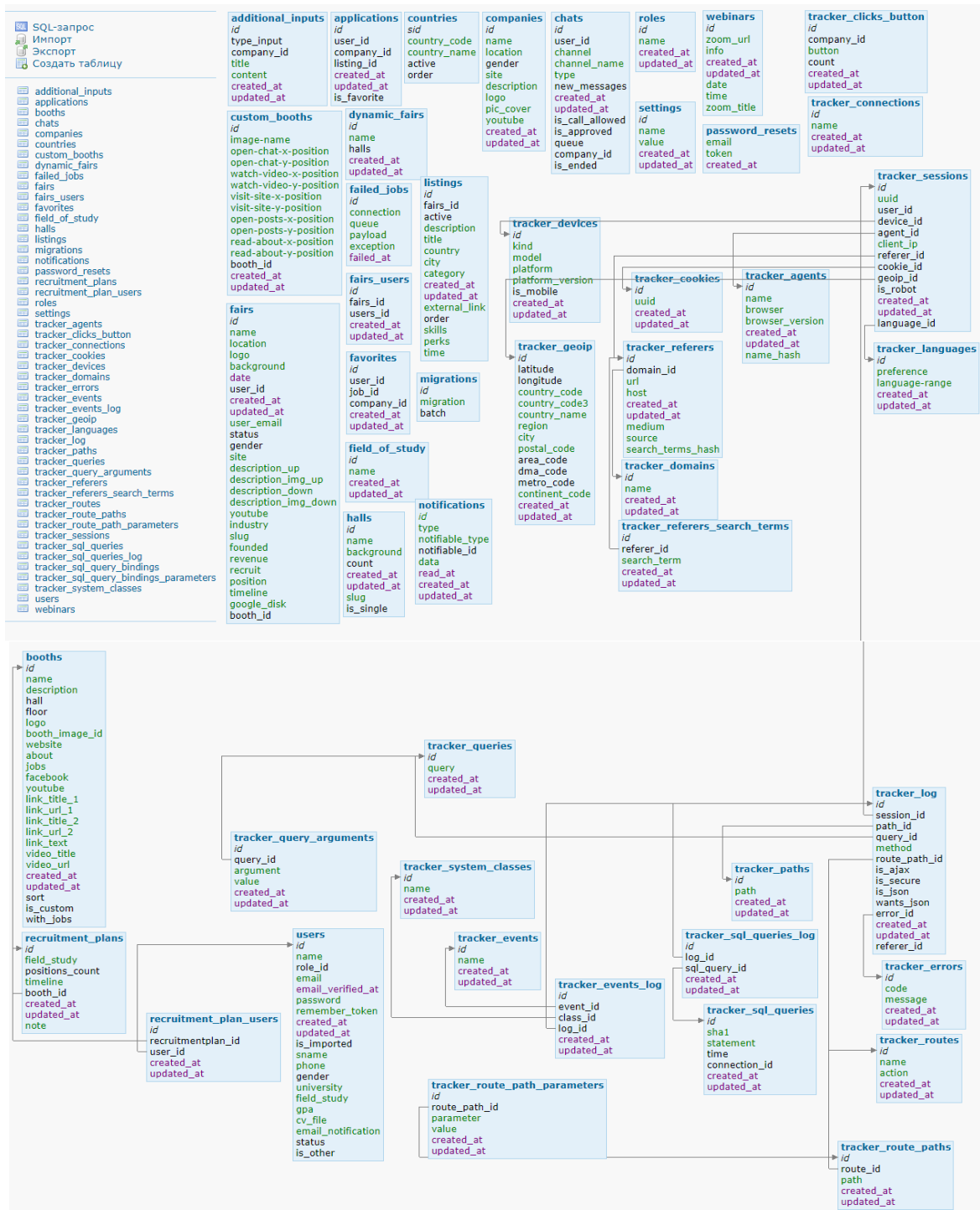


Рис. 2.4.1. Схема бази даних додатку

Eloquent відноситься до низки ORM під назвою Active Record. Це найбільш розповсюджений та найпростіший шаблон проектування, який створений на ідеї, що кожній таблиці відповідає якийсь клас, тобто модель.

Перед початком роботи з Eloquent треба налаштувати з'єднання з базою даних. Всі дані для налаштування зберігаються в директорії `config/database.php`. Тут можна вказати всі з'єднання, які будуть використовуватись базою даних. Ось приклад налаштування з'єднання нашої системи з базою даних `mysql`.

```
<?php
use Illuminate\Support\Str;
return [
    'default' => env('DB_CONNECTION', 'mysql'),
    'connections' => [
        'sqlite' => [
            'driver' => 'sqlite',
            'url' => env('DATABASE_URL'),
            'database' => env('DB_DATABASE',
database_path('database.sqlite')),
            'prefix' => '',
            'foreign_key_constraints' => env('DB_FOREIGN_KEYS',
true),
        ],

        'mysql' => [
            'driver' => 'mysql',
            'url' => env('DATABASE_URL'),
            'host' => env('DB_HOST', '127.0.0.1'),
            'port' => env('DB_PORT', '3306'),
            'database' => env('DB_DATABASE', 'forge'),
            'username' => env('DB_USERNAME', 'forge'),
            'password' => env('DB_PASSWORD', ''),
            'unix_socket' => env('DB_SOCKET', ''),
            'charset' => 'utf8mb4',
            'collation' => 'utf8mb4_unicode_ci',
            'prefix' => '',
            'prefix_indexes' => true,
            'strict' => true,
            'engine' => null,
            'options' => extension_loaded('pdo_mysql') ?
array_filter([
                PDO::MYSQL_ATTR_SSL_CA => env('MYSQL_ATTR_SSL_CA'),
            ]) : [],
        ],

        'tracker' => [
            'driver' => 'mysql',
            'url' => env('DATABASE_URL'),
            'host' => env('DB_HOST', '127.0.0.1'),
            'port' => env('DB_PORT', '3306'),
            'database' => env('DB_DATABASE', 'forge'),
            'username' => env('DB_USERNAME', 'forge'),
            'password' => env('DB_PASSWORD', ''),
            'unix_socket' => env('DB_SOCKET', ''),
            'charset' => 'utf8mb4',
            'collation' => 'utf8mb4_unicode_ci',
```

```

        'prefix' => '',
        'prefix_indexes' => true,
        'strict' => false,
        'engine' => null,
        'options' => extension_loaded('pdo_mysql') ?
array_filter([
    PDO::MYSQL_ATTR_SSL_CA => env('MYSQL_ATTR_SSL_CA'),
]) : [],
],

'pgsql' => [
    'driver' => 'pgsql',
    'url' => env('DATABASE_URL'),
    'host' => env('DB_HOST', '127.0.0.1'),
    'port' => env('DB_PORT', '5432'),
    'database' => env('DB_DATABASE', 'forge'),
    'username' => env('DB_USERNAME', 'forge'),
    'password' => env('DB_PASSWORD', ''),
    'charset' => 'utf8',
    'prefix' => '',
    'prefix_indexes' => true,
    'schema' => 'public',
    'sslmode' => 'prefer',
],

'sqlsrv' => [
    'driver' => 'sqlsrv',
    'url' => env('DATABASE_URL'),
    'host' => env('DB_HOST', 'localhost'),
    'port' => env('DB_PORT', '1433'),
    'database' => env('DB_DATABASE', 'forge'),
    'username' => env('DB_USERNAME', 'forge'),
    'password' => env('DB_PASSWORD', ''),
    'charset' => 'utf8',
    'prefix' => '',
    'prefix_indexes' => true,
],
],
'migrations' => 'migrations',
'redis' => [
    'client' => env('REDIS_CLIENT', 'phpredis'),
    'options' => [
        'cluster' => env('REDIS_CLUSTER', 'redis'),
        'prefix' => env('REDIS_PREFIX',
Str::slug(env('APP_NAME', 'laravel'), '_').'_database_'),
    ],
    'default' => [
        'url' => env('REDIS_URL'),
        'host' => env('REDIS_HOST', '127.0.0.1'),
        'password' => env('REDIS_PASSWORD', null),
        'port' => env('REDIS_PORT', '6379'),
        'database' => env('REDIS_DB', '0'),
    ],
],

```



```

        'cache' => [
            'url' => env('REDIS_URL'),
            'host' => env('REDIS_HOST', '127.0.0.1'),
            'password' => env('REDIS_PASSWORD', null),
            'port' => env('REDIS_PORT', '6379'),
            'database' => env('REDIS_CACHE_DB', '1'),
        ],
    ],
];

```

Після цього можна почати створювати моделі Eloquent. Найпростішим способом створення моделі є створення через консоль командою `php artisan make: model`. Ось приклад створеної моделі Hall:

```

<?php
namespace App;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Support\Str;
use Illuminate\Support\Facades\Storage;
use App\Booth;
class Hall extends Model
{
    protected $guarded = [];
    public function getHallBackground()
    {
        if($this->background == null)
        {
            return ;
        }
        return Storage::disk('hallbackground')->url($this->
background);
    }
    public function nextRecord()
    {
        $nextHall = $this->where('id', '>', $this->id)-
>orderBy('id')->first();
        if($nextHall) return $nextHall;
    }
    public function previousRecord()
    {
        $prevHall = $this->where('id', '<', $this->id)-
>orderBy('id', 'DESC')->first();
        if ($prevHall) return $prevHall;
    }

    public function hasBooths()
    {
        return $this->hasMany(Booth::class, 'hall', 'id');
    }

    public function boothsWithActiveCompanies()

```

```

    {
        return $this->hasMany(Booth::class, 'hall', 'id')
            ->whereHas('hasCompany', function ($q){
                $q->where('status', 1);
            });
    }

    public function getLink()
    {
        return '/hall-' . $this->slug;
    }
}

```

Наступним пунктом, треба висвітлити міграції в РНР Laravel. Міграції – це один із способів системи контролю версії бази даних. Міграції дозволяють всім учасникам команди розробки бачити та ділитися змінами, які вносяться в поточну версію бази даних. Для створення міграції в консолі треба викликати команду `php artisan make: migration` та вказати назву міграції[24]. Ось перелік міграцій, які використовувались при розробці нашої платформи:

```

2020_07_14_114233_add_starts_on_to_webinars
2020_08_13_063639_create_recruitment_plans_table
2020_07_27_084627_create_tracker_buttons_table
2020_08_11_094104_add_is_ended_to_chats
2020_07_04_081953_create_booths_table
2020_08_21_171411_add_note_to_recruitment_plans
2020_08_25_092730_create_recruitment_plan_users_table
2020_07_02_080004_create_notifications_table
2020_07_16_125112_add_company_id_to_chats
2020_08_20_110848_add_zoom_url_2_and_zoom_titles_to_webinars
2020_06_25_090705_add_columns_to_listings_table
2020_07_20_172922_add_is_custom_to_booths_table
2020_07_13_093727_change_column_type_table_booth
2020_08_21_123349_remove_zoom_title_2_and_zoom_link_2_from_webinar
s
2020_07_08_113925_add_sort_column_to_booths_table
2020_07_31_091328_create_dynamic_fairs_table
2020_08_11_055445_add_is_other_to_users
2020_07_20_171225_create_custom_booths_table
2020_07_20_145145_add_column_to_application_table
2020_07_06_093310_change_column_type_table_booths_
2020_07_12_094111_add_queue_number_to_chats
2020_07_11_192241_add_is_approved_to_chats
2020_07_21_091853_add_column_to_halls_table
2020_06_22_071101_add_order_column_to_listings
2020_07_14_094702_create_users_fairs_table
2020_06_30_133736_create_chats_table
2020_07_06_123436_add_column_booth_to_fairs_table

```

2020\_08\_12\_195907\_add\_with\_jobs\_to\_booths  
2020\_07\_02\_070449\_add\_email\_notific\_to\_users\_table  
2020\_07\_09\_082555\_create\_settings\_table  
2020\_07\_27\_162101\_add\_status\_to\_users  
2020\_06\_29\_091759\_create\_halls\_table  
2020\_07\_11\_092852\_add\_is\_call\_allowed\_to\_chats\_table  
2020\_08\_10\_102053\_create\_study\_fields\_table  
2020\_08\_20\_085807\_add\_is\_single\_column\_to\_halls  
2020\_06\_27\_131331\_create\_applications\_table  
2020\_06\_19\_203046\_add\_external\_link\_to\_listings  
2020\_06\_19\_090432\_create\_favorites\_table  
2020\_06\_19\_084122\_create\_additional\_table\_table  
2020\_06\_18\_124829\_create\_webinars\_table  
2020\_06\_17\_120946\_add\_columns\_to\_fairs\_tables  
2020\_06\_15\_131032\_add\_columns\_to\_users\_table  
2020\_06\_12\_120323\_create\_listings\_table  
2020\_06\_11\_151626\_add\_columns\_to\_fairs\_table  
2020\_06\_11\_144342\_create\_companies\_table  
2020\_05\_07\_164535\_create\_fairs\_table  
2020\_05\_06\_101822\_create\_roles\_table  
2019\_08\_19\_000000\_create\_failed\_jobs\_table  
2017\_12\_13\_150000\_fix\_query\_arguments  
2017\_06\_20\_311102\_add\_agent\_name\_hash  
2017\_01\_31\_311101\_fix\_agent\_name  
2015\_11\_23\_311100\_add\_nullable\_to\_tracker\_error  
2015\_11\_23\_311099\_add\_tracker\_language\_foreign\_key\_to\_sessions  
2015\_11\_23\_311097\_create\_tracker\_languages\_table  
2015\_11\_23\_311098\_add\_language\_id\_column\_to\_sessions  
2015\_11\_23\_311096\_add\_tracker\_referer\_column\_to\_log  
2015\_03\_13\_311095\_add\_tracker\_referer\_columns  
2015\_03\_13\_311094\_create\_tracker\_referer\_search\_term\_table  
2015\_03\_07\_311093\_create\_tracker\_tables\_relations  
2015\_03\_07\_311092\_create\_tracker\_connections\_table  
2015\_03\_07\_311090\_create\_tracker\_sql\_query\_bindings\_parameters\_table  
2015\_03\_07\_311091\_create\_tracker\_sql\_queries\_log\_table  
2015\_03\_07\_311089\_create\_tracker\_sql\_query\_bindings\_table  
2015\_03\_07\_311088\_create\_tracker\_sql\_queries\_table  
2015\_03\_07\_311087\_create\_tracker\_events\_log\_table  
2015\_03\_07\_311086\_create\_tracker\_events\_table  
2015\_03\_07\_311085\_create\_tracker\_log\_table  
2015\_03\_07\_311084\_create\_tracker\_system\_classes\_table  
2015\_03\_07\_311083\_create\_tracker\_errors\_table  
2015\_03\_07\_311082\_create\_tracker\_sessions\_table  
2015\_03\_07\_311081\_create\_tracker\_geoip\_table  
2015\_03\_07\_311080\_create\_tracker\_referers\_table  
2015\_03\_07\_311079\_create\_tracker\_domains\_table  
2015\_03\_07\_311078\_create\_tracker\_devices\_table  
2015\_03\_07\_311077\_create\_tracker\_cookies\_table  
2015\_03\_07\_311076\_create\_tracker\_agents\_table  
2015\_03\_07\_311075\_create\_tracker\_route\_path\_parameters\_table  
2015\_03\_07\_311074\_create\_tracker\_routes\_paths\_table  
2015\_03\_07\_311073\_create\_tracker\_routes\_table

```

2015_03_07_311072_create_tracker_queries_arguments_table
2015_03_07_311071_create_tracker_queries_table
2015_03_07_311070_create_tracker_paths_table
2014_10_12_100000_create_password_resets_table
2014_10_12_000000_create_users_table

```

### Приклад міграції 2020\_06\_29\_091759\_create\_halls\_table

```

<?php
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;
class CreateHallsTable extends Migration
{
    public function up()
    {
        Schema::create('halls', function (Blueprint $table) {
            $table->id();
            $table->string('name');
            $table->string('background')->nullable();
            $table->integer('count')->default(20);
            $table->timestamps();
        });
    }
    public function down()
    {
        Schema::dropIfExists('halls');
    }
}

```

## 2.5. Основні функції додатку

### 2.5.1. Реєстрація та аутентифікація користувачів

Реєстрація користувачів була виконана за допомогою внутрішніх інструментів фреймворка Laravel. Цей фреймворк передбачає створення контролерів даних без втручання розробника, але при потребі їх можна кастомізувати під власні потреби. Для виконання реєстрації, зміни паролю та іншого функціоналу при реєстрації наявні такі контролери:

- `ConfirmPasswordController` – відповідає за логіку підтвердження паролю;
  - `ForgotPasswordController` – відповідає за логіку відновлення електронної пошти;
  - `LoginController` – відповідає за аутентифікацію користувача та перенаправлення його на подальшу домашню сторінку;
  - `RegisterController` – відповідає за валідацію та реєстрацію користувачів;
  - `ResetPasswordController` – відповідає за логіку сценарію при відновленні пароля;
  - `VerificationController` – відповідає за верифікацію електронної пошти
- Всі ці контролери знаходяться у директорії `\app\Http\Controllers\Auth\`.

Для прикладу покажемо реалізацією одного з головних контролерів – `RegisterController`.

```
class RegisterController extends Controller
{
    use RegistersUsers;
    protected $fair;
    public function __construct(Fair $fair)
    {
        //$this->middleware('guest');
        $this->fair = $fair;
    }
    protected function validator(array $data)
    {
        return Validator::make($data, [
            'name' => ['required', 'string', 'max:255'],
            'email' => ['required', 'string', 'email', 'max:255',
'unique:users'],
```

```

        'password' => ['required', 'string', 'min:8', 'confirmed'],
    ]);
}
protected function create(array $data)
{
    try{
        return User::create([
            'name' => $data['name'],
            'email' => $data['email'],
            'password' => Hash::make($data['password']),
        ]);
    }catch (Exception $e){
        $errorCode = $e->errorInfo[1];
        if($errorCode == 1062){
            return redirect()->route('register_company');
        }
    }
}
protected $redirectTo;
public function redirectTo()
{
    switch(Auth::user()->role_id){
        case 1:
            $this->redirectTo = '/admin';
            return $this->redirectTo;
            break;
        case 2:
            $this->redirectTo = '/admin';
            return $this->redirectTo;
            break;
        case 3:
            $this->redirectTo = '/manager';
            return $this->redirectTo;
            break;
        case 4:
            $this->redirectTo = '/user';
            return $this->redirectTo;
            break;
        default:
            $this->redirectTo = '/user';
            return $this->redirectTo;
    }
}
public function showRegistrationFormEmployer()
{
    return view('auth.registration_employer_user');
}
public function registrationCompany(Request $request)
{
    $this->validate($request, [
        'name' => 'required',
        'location' => 'required',
        'logo' => 'required|image',
    ]);
}

```

```

        'gender' => 'numeric',
        'founded' => 'nullable|numeric',
        'revenue' => 'nullable|numeric',
        'phone' => 'required|numeric',
        'website' => 'required',
        'industry' => 'required'
    ]
);
try{
    $data = [
        'name'=>$request->name,
        'email'=>$request->email,
        'password'=>Hash::make($request->password),
        'role_id' => 5,
        'phone'=> $request->phone,
    ];

    User::create($data)->notify(new
CompanyCreate(['name'=>$request->name, 'user_email' => $request-
>email]));

        $filename = Str::random(10) . '.' . $request-
>file('logo')->extension();
        Storage::disk('logos')->put($filename,
file_get_contents($request->file('logo')));

        $check = $this->fair->where('slug', Str::slug($request-
>company, '-'))->get();

    $this->fair->create([
        'name'=> $request->company,
        'location'=> $request->location,
        'logo'=> $filename,
        // 'user_id'=> User::where('role_id',3)->first()-
>id,
        'user_email'=> $request->email,
        'status'=> 0,
        'gender'=> $request->gender,
        'site'=> $request->website,
        'youtube'=> $request->youtube,
        'industry'=> $request->industry,
        'founded'=> $request->founded,
        'revenue'=> $request->revenue,
        'recruit'=> $request->recruit,
        'position'=> $request->position,
        'timeline'=> $request->timeline,
        'google_disk'=> $request->google_disk,
        'slug'=> count($check)>0? $this->slug =
Str::slug($request->company.'-'.time(), '-'): $this->slug =
Str::slug($request->company, '-'),])->notify(new
CompanyCreate(['name'=>$request->company, 'user_email'=>$request-
>email]));

```

```

$text = "The company - {$request->company} was created. Contact
person - {$data['email']} / {$data['phone']}";
$text_to_company=Setting::where('name','company_template')->first;
    if($text_to_company){
        $text_to_company = preg_replace('/{{name}}/',
$request->company, $text_to_company->value);
    }
    try{
        Mail::send(['text'=>'mail_notification'], compact('text'),
function($message->to(User::where('role_id',1)->first()-
>email)>subject('New company was created');
$message->sender(env("MAIL_FROM_ADDRESS",'test@medaxpo.com'),
parse_url(config('app.url'), PHP_URL_HOST));});
        if($text_to_company){
            Mail::raw($text_to_company, function($message)
use($request) {
                $message->to($request->email)->subject('Welcome to ').
parse_url(config('app.url'), PHP_URL_HOST);
                $message->sender(env("MAIL_FROM_ADDRESS",
'test@medaxpo.com'), parse_url(config('app.url'), PHP_URL_HOST));
            });
        }
    }
    catch (\Exception $e) {
        \Log::error($e->getMessage());
    }

    $this->fair->notify(new CompanyCreate($fair));
}catch(Exception $e){
    Log::error($e->getMessage().'/'. $e->getLine());
}
return redirect()->route('login')->with('message', 'Thank
you for registration, now you can login!');
}
}

```

### 2.5.2. Реалізація подання користувача на вакансію

За логікою платформи користувач може подати свою кандидатуру на вакансію, яку завчасно створив менеджер компанії. Для цього користувачу потрібно зайти на стенд компанії та вибрати вакансію, яка його зацікавила. Після цього натиснути на кнопку “Apply this job”. Після виконання цього кроку в адміністративній панелі користувача менеджера компанії з’явиться відповідна



інформація про успішне подання на вакансію. Розглянемо як була реалізована ця частина програми.

Кожен користувач має свій id, який зв'язується з id вакансії та заноситься в таблицю бази даних під назвою Application та спрацьовує функція Mail, яка надсилає на пошту повідомлення про успішне подання на вакансію.

```
class ContactController extends Controller
{
    public function index(Request $request)
    {
        $data = Auth::user();
        $candidate = Application::where([
            'user_id' => Auth::user()->id,
            'company_id' => $request->get('companyID'),
            'listing_id' => $request->get('id')
        ])->first();
        $vacancy = Listing::where('id', $request->id)-
>with('company')->get();
        if($candidate) {
            $candidate->touch();
        } else {
            $application = new Application([
                'user_id' => Auth::user()->id,
                'company_id' => $request->get('companyID'),
                'listing_id' => $request->get('id')
            ]);
            $application->save();
            $application->notify(new MailUserApplyVacancy(Auth::user(),
            $vacancy[0]->company->name));
        }
        try{
            Mail::send(['text'=>'mail'], compact('data',
            'vacancy'), function($message) use ($vacancy) {
                $message->to($vacancy[0]->company->user_email)-
>subject('Interested vacancy');
                $message->sender(env("MAIL_FROM_ADDRESS", 'test@medaxpo.com'),
                parse_url(config('app.url'), PHP_URL_HOST));
            });
        }
        catch (Exception $e) {
            \Log::error($e->getMessage());
        }
        return response()->json(['message' => 'Success']);
    }
}
```

### 2.5.3. Налаштування інструментарію для відеодзвінків

Для того, щоб не розробляти свій функціонал, який займе багато часу та ресурсів, було прийнято рішення використати готове рішення для відеодзвінків. Для цього була використана API Vonage. Для початку роботи з цією технологією треба встановити NodeJS. Ця технологія включає в себе багато модулів, серед яких socket.io. Цей модуль є інструментом для використання відкритих з'єднань з метою полегшення зв'язку в режимі реального часу. Socket.IO дозволяє двонаправлений зв'язок між клієнтом і сервером. Двонаправлений зв'язок вмикається, коли клієнт має Socket.IO у браузері, а сервер також інтегрував пакет Socket.IO. Після встановлення NodeJS треба вказати директиви для модуля socket.io у файлі server.js.

Для подальшої роботи з цією API треба зареєструватись на сайті Vonage та створити проект. В запропонованих полях потрібно ввести назву проекту та URL проекту (рис. 2.5.3.1).

Після успішної реєстрації вкажемо який порт повинна використовувати API у файлі ChatComponent:

```
created() {  
  
    this.socket = io('http://studyar.local.ua:3000');
```

Також для роботи треба запустити сервер NodeJS за допомогою команди в терміналі “node server.js”.

Після того як розробник вказав правильні дані порту та запустив сервер, користувачі зможуть користуватись відеодзвінками.

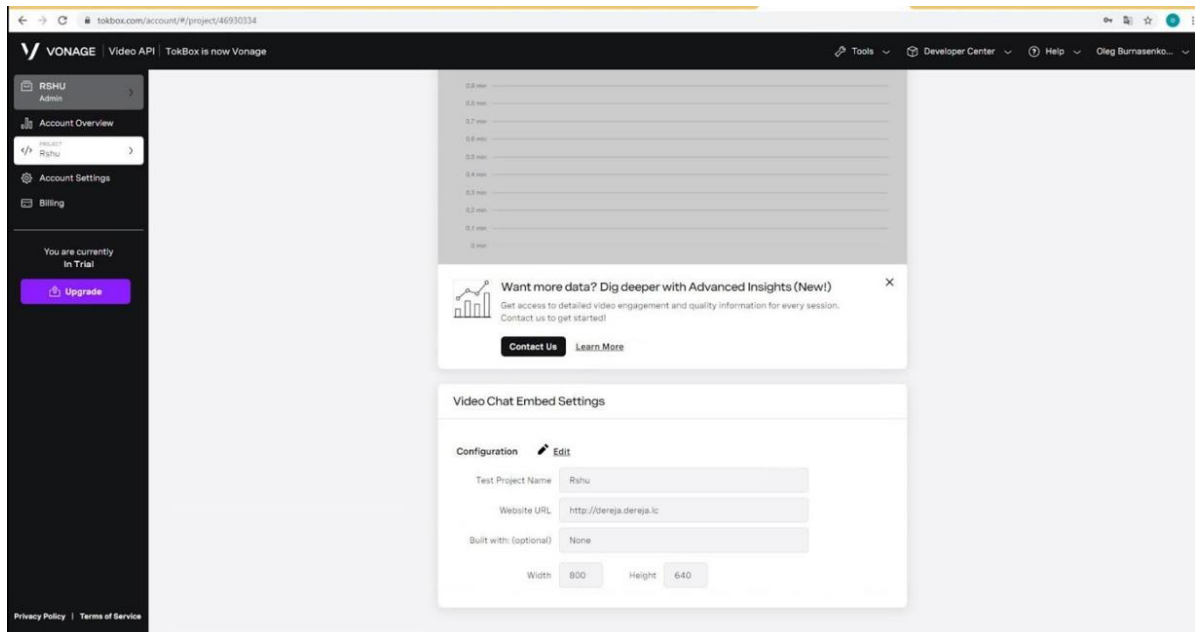


Рис. 2.5.3.1. Створення проекту на сайті Vonage

## 2.5.4. Налаштування інструментарію для чату між користувачами

Для додавання функціоналу чату було прийнято рішення використовувати готовий інструмент для чату. Він реалізований за допомогою API PubNub. Це легкий у використанні API, який не бере за основу використання сокетів, а використовує власні ресурси на власному сервері. Для початку для роботи потрібно зареєструватись на сайті [pubnub.com](http://pubnub.com) та створити проект на цій платформі(рис. 2.5.4.1).

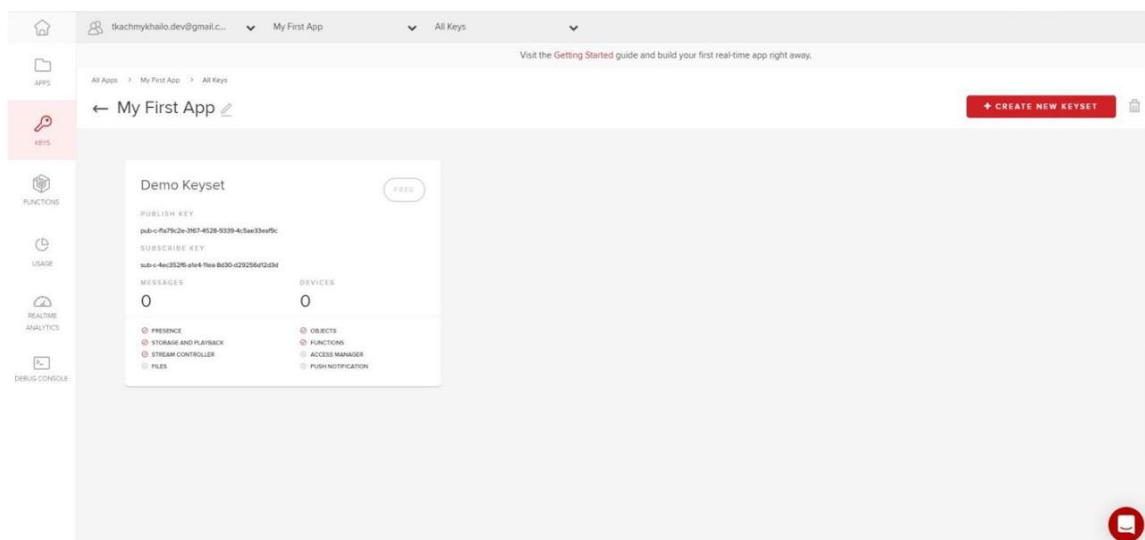


Рис. 2.5.4.1. Створення проекту на сайті PubNub

Після цього отримаємо набір ключів для підключення інструменту до програми (рис. 2.5.4.2).

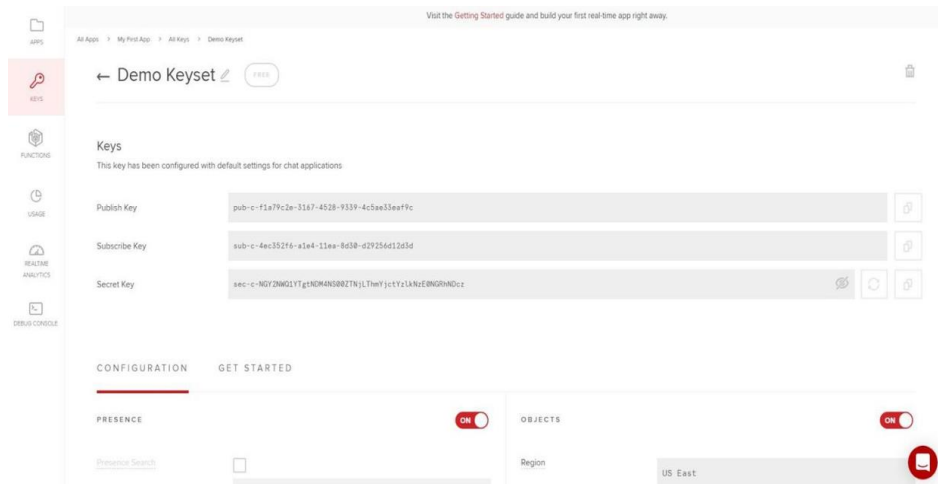


Рис. 2.5.4.2. Отримання ключів на сайті PubNub

Далі налаштуємо середовище та вставимо отримані ключі у файл `app.js`.

```
Vue.use(PubNubVue, {
  subscribeKey: 'sub-c-7286729e-2507-11eb-862a-82af91a3b28d',
  publishKey: 'pub-c-f91c5c08-aff7-44cc-9fe2-ad8b5bc1279f'
});
```

Після цього скопіюємо проект та отримаємо чат, в якому можуть спілкуватись користувачів

## РОЗДІЛ 3. ІНСТРУКЦІЯ З ВИКОРИСТАННЯ

Цей програмний продукт умовно можна розділити на 3 частини:

- сторінка реєстрації користувача та вхід у систему;
- панель адміністратора;
- власне сам ярмарок.

Розглянемо кожну частину окремо.

### 3.1. Сторінка реєстрації користувача та вхід у систему

Зайшовши на сторінку бачимо вітальний напис. В ньому пропонується зареєструватись на даній платформі або ж залогінитись під існуючими доступами. У правій частині екрана в білому блоці наявні два поля: пошти та паролю, кнопка “Забув пароль” та кнопка реєстрації (рис. 3.1.1).

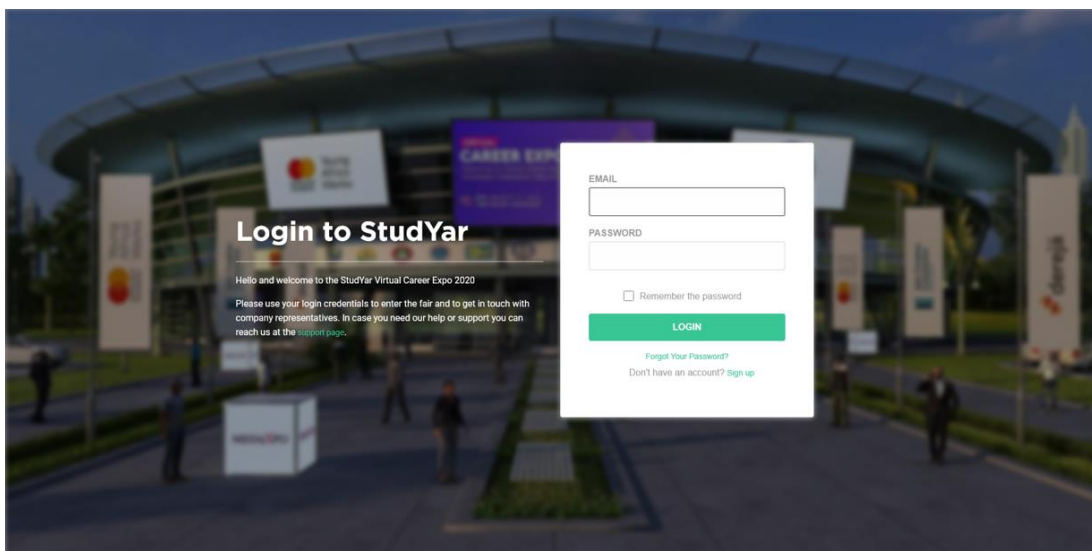


Рис. 3.1.1. Вітальна сторінка платформи

#### 3.1.1. Сторінка реєстрації

Для входу на сторінку реєстрації, натиснемо кнопку “Sign up”. Перейдемо на сторінку реєстрації студента. Зверху наявні 3 види реєстрації:

- Graduate
- Other
- Register Company

Першою з'являється сторінка реєстрації студента (рис. 3.1.1.1). Тут наявні такі поля для реєстрації:

- First name – ім'я
- Last Name – прізвище
- Email – електронна пошта
- Phone number – номер телефону
- Choose Password – пароль
- Gender – стать
- Name of University – назва університету
- Field of Study – спеціальність
- GPA – середній бал
- Upload Your CV – завантаження резюме

Рис. 3.1.1.1. Сторінка реєстрації студента

Також на сторінці наявна перевірка CAPTCHA (Completely Automated Public Turing test to tell Computers and humans apart) – автоматизований

публічний тест Тьюринга для того, щоб була можливість відрізнити користувачів-людей від комп'ютерів. Після заповнення всіх полів потрібно натиснути кнопку Register.

Сторінка реєстрації користувача **other** розрахована на представника компанії, який може вести перемовини в чаті, але не має можливості вносити зміни у свою компанію (рис. 3.1.1.2.). На цій сторінці наявні такі поля для реєстрації:

- First name – ім'я
- Last Name – прізвище
- Email – електронна пошта
- Phone number – номер телефону
- Choose Password – пароль
- Gender – стать
- Name of Organization – назва організації

The screenshot shows a registration form with the following elements:

- Title:** Registration
- Link:** Already have an account? [Sign in](#)
- Tabs:** Graduate, Other (selected), Register Company
- Fields:**
  - First Name\* (text input)
  - Last Name\* (text input)
  - Email\* (text input)
  - Phone number\* (text input, example: +251-9xxx)
  - Choose Password\* (text input)
  - Gender\* (radio buttons for Male and Female, Male is selected)
  - Name of the organization\* (dropdown menu, currently showing "-- Please Select --")

Рис.3.1.1.2. Сторінка реєстрації студента (продовження)

Перейдемо на сторінку реєстрації компанії. На цій сторінці наявні такі поля для реєстрації(рис. 3.1.1.3):

- Organization name – назва організації
- Contact Person – контактна людина
- Email – електронна пошта
- Phone – номер телефону
- Password – пароль
- Office Location – місцезнаходження офісу
- Website – веб сайт
- Industry – індустрія
- Founded year – рік створення
- Revenue \$m – прибуток компанії
- Youtube link – посилання на канал компанії на Youtube
- Logo – логотип компанії
- Gender – покажчик який показує співвідношення статей у компанії
- Link on your Google Disk – посилання на Google Disk компанії
- Recruiting this Season(Yes/No) – чи наймає компанія цього сезону
- Number of available positions(1-5/5-10/10+) – число наявних вакансій
- Recruiting timeline(0-3 months/3-6 months/6+ months) – термін найму

## Employer Registration

If you are interested in participating in the Virtual Career expo, please register for a demo below.

Organization name *			
Contact Person *	Email *		
Phone *	Password *		
Office Location *			
Website *			
Industry *	Founded year *		
Revenue \$m *	Youtube link *		
Logo * <input type="text" value="Upload your logo"/>	Gender * <input checked="" type="radio"/> Select gender ratio <input type="range" value="50"/>	<input type="radio"/> Do not state this information	
Link on your Google Disk *			
Recruiting this Season? * <input type="radio"/> Yes <input type="radio"/> No			
Number of available positions * <input type="radio"/> 1-5 <input type="radio"/> 5-10 <input type="radio"/> 10+			
Recruiting timeline * <input type="radio"/> 0-3 months <input type="radio"/> 3-6 months <input type="radio"/> 6+ months			
<input type="checkbox"/> I am a job seeker		<input type="button" value="Register"/>	



### Рис. 3.1.1.3. Сторінка реєстрації студента

Ці поля можна використовувати для виводу їх на екрані або стенді під час проведення ярмарки.

### 3.1.2. Функція ”Забув пароль”

Ця функція використовується користувачами, якщо вони забули пароль та не можуть зайти в систему(рис. 3.1.2.1).

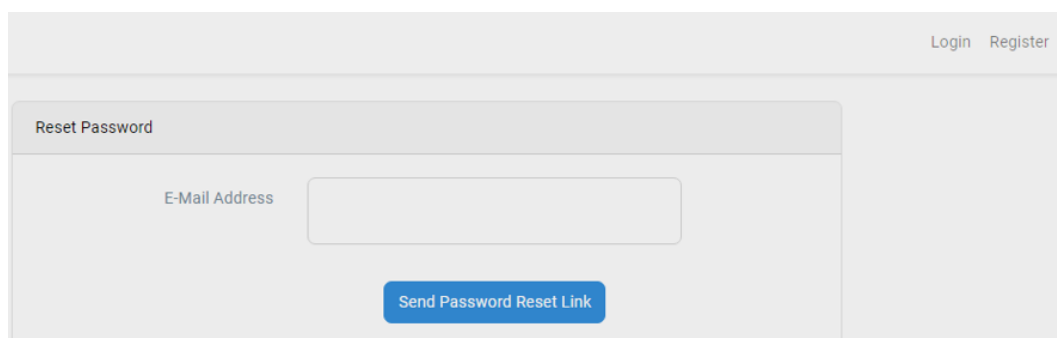


Рис. 3.1.2.1. Функція ”Забув пароль”

Для того щоб скинути пароль, потрібно натиснути кнопку “Forgot Password”. Наступним кроком є ввід своєї електронної пошти. При натисненні кнопки “Send Password Reset Link” система відправляє на пошту користувача посилання на зміну паролю. Після переходу за цим посиланням користувач може встановити новий пароль і увійти в систему за допомогою нового паролю.

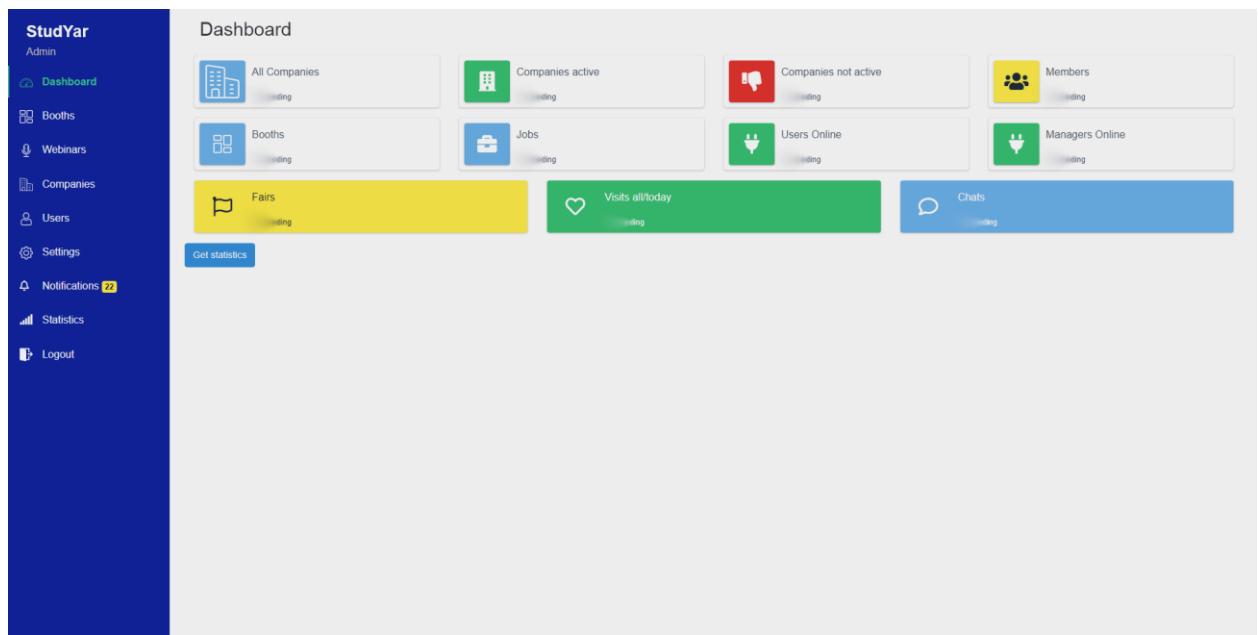
## 3.2 Адміністраторська панель

Одразу після входу в систему користувача здійснюється перенаправлення на адміністраторську панель. У кожній ролі користувача наявна така адміністраторська панель, пов’язана з його роллю. Це означає що у адміністратора найбільший перелік пунктів меню адміністраторської панелі, а у

студента – найменше. Розглянемо всі пункти меню адміністраторської панелі адміністратора та їх функціонал.

### 3.2.1. Адміністраторська панель адміністратора ярмарки

Перше що бачить адміністратор після того як виконав вхід в систему є



дошка (рис. 3.2.1.1).

Рис.3.2.1.1. Сторінка Dashboard

На цій дошці розміщені віджети основних витягів зі статистики а саме:

- Кількість компаній
- Кількість активованих компаній
- Кількість неактивованих компаній
- Кількість учасників ярмарки
- Кількість стендів
- Кількість наявних вакансій на ярмарку
- Кількість користувачів онлайн
- Кількість менеджерів компаній онлайн

- Кількість загальних візитів/ візитів за сьогодні
- Кількість чатів

Другим пунктом є меню стендів. Перейшовши на нього ми бачимо вкладки, які відповідають за холи (рис. 3.2.1.2).

Перейшовши на вибрану вкладку ми бачимо таблицю з переліком стендів, наявних в цьому холі. Ця таблиця складається з таких полів:

- Booth name – назва стенда
- Booth ID – ідентифікатор стенда в базі даних
- Hall – номер хола, в якому знаходиться даний стенд
- Floor – номер поверха, на якому знаходиться стенд
- Background – вигляд стенда
- Action – набір кнопок який складається з кнопки переходу на стенд самої компанії в холі, редагування, видалення стенда

Booth name	Booth ID	Hall	Floor	Background	Company	Action
SoftServe	7	1	1		SoftServe	
Honeycomb	13	1	1		Honeycomb	
Beetroot Academy	14	1	1		Beetroot Academy	
IT Step	15	1	1		IT Step	
DataArt	11	1	1		DataArt	
KTC	8	1	1		KTC	
SaintScript	12	1	1		SaintScript	
Citrus	17	1	1		Citrus	

Рис. 3.2.1.2. Сторінка Booths

Наступним пунктом меню є розділ Webinars. У ньому адміністратор може вносити дані стосовно майбутнього вебінару (рис. 3.2.1.3).

**Webinars** + Add

Date: 30.11.2020 Time: 18:00

Info

Apply

### Рис. 3.2.1.3. Сторінка Webinars

Перейшовши на цю сторінку ми бачимо такі пункти меню:

- Add – кнопка додавання нового вебінару
- Date – Datetime елемент для встановлення дати проведення вебінару
- Time – поле для вводу точного часу проведення вебінару
- Info – поле куди заноситься скрипт самого вебінару

Наступним пунктом меню є пункт меню Companies (рис. 3.2.1.4).

ID	Name	Location	Logo	Date	Status	Actions
12	SoftServe	Rivne		2020-11-10 13:10:59	<input checked="" type="checkbox"/>	
13	KTC	Rivne		2020-11-10 14:05:56	<input checked="" type="checkbox"/>	
14	SoftGroup	Rivne		2020-11-10 14:09:24	<input checked="" type="checkbox"/>	
15	Renome Smart	Rivne		2020-11-10 15:59:45	<input checked="" type="checkbox"/>	
16	DataArt	Lviv		2020-11-10 16:43:10	<input checked="" type="checkbox"/>	
17	SaintScript	Rivne		2020-11-10 16:44:17	<input checked="" type="checkbox"/>	
18	Honeycomb	Rivne		2020-11-10 16:46:33	<input checked="" type="checkbox"/>	
19	Beetroot Academy	Rivne		2020-11-10 16:48:16	<input checked="" type="checkbox"/>	
20	IT Step	Rivne		2020-11-10 16:49:49	<input checked="" type="checkbox"/>	
21	Selecto	Rivne		2020-11-10 16:51:38	<input checked="" type="checkbox"/>	

Рис. 3.2.1.4. Сторінка Companies

Перейшовши на цю сторінку ми бачимо кнопку меню Add new company, яка додає нову компанію, кнопку створення xlsx зі списком компаній, та такі поля в таблиці:

- ID – ідентифікатор компанії в базі даних
- Name – назва компанії
- Location – місцезнаходження компанії
- Logo – логотип компанії
- Date – дата створення даної компанії
- Status – чи активована компанія
- Actions – набір кнопок, такі як:
  - редагування компанії,

- перехід на саму компанію в холі,
- видалення компанії,
- створення вакансії для компанії,

Наступним пунктом меню є пункт Users. Перше, що можемо бачити на цій сторінці це кнопки Add user, Search, Import Users, Export Users (рис. 3.2.1.5).

ID	Name	Email	Creation date	Phone	Gender	University	Field of Study	GPA	Resume	Status	Action
47	vodafone@mail.com	vodafone@mail.com	2020-11-10 20:59:38							Active	✕
46	lifecell@mail.com	lifecell@mail.com	2020-11-10 20:58:20							Active	✕
45	pumb@mail.com	pumb@mail.com	2020-11-10 20:56:51							Active	✕
44	oshchadbank@mail.com	oshchadbank@mail.com	2020-11-10 20:54:44							Active	✕
43	mono@mail.com	mono@mail.com	2020-11-10 20:50:29							Active	✕
42	mindy@mail.com	mindy@mail.com	2020-11-10 20:47:57							Active	✕
41	zagrava@mail.com	zagrava@mail.com	2020-11-10 20:45:41							Active	✕
40	lohika@mail.com	lohika@mail.com	2020-11-10 20:44:22							Active	✕
39	privatbank@mail.com	privatbank@mail.com	2020-11-10 20:32:09							Active	✕
38	smokoko@mail.com	smokoko@mail.com	2020-11-10 20:29:46							Active	✕

Рис. 3.2.1.5. Сторінка Users

Перша кнопка відповідає за додавання нового користувача, а друга за їх пошук. При натисканні на кнопку Search ми бачимо як перед нами з'являються ще поля, за якими можна здійснювати пошук:

- Name – ім'я
- Email – електронна пошта
- University – назва університету. в якому навчається користувач
- Field of study – напрямок навчання користувача
- Status – статус користувача (активний/неактивний)

Фільтрація здійснюється за кожним з цих полів. Після заповнення поля пошуку треба натиснути клавішу Search. При натисканні на клавішу Import Users ми можемо імпортувати користувачів, при натисненні Export Users ми їх можемо експортувати. Передача даних відбувається файлами xlsx.

Наступне, що бачить користувач, зайшовши в панель адміністратора це таблиця зі списком користувачів, яка складається з таких полів:

- ID – ідентифікатор користувача в базі даних
- Name – ім'я користувача
- Email – електронна пошта
- Creation Date – дата реєстрації користувача
- Phone – номер телефона
- Gender – стать користувача
- University – університет, в якому навчається користувач
- Field of study – напрямок навчання
- GPA – середній бал
- Resume – резюме
- Status – чи активований користувач (Так/Ні)
- набір кнопок, який складається з:
  - видалення,
  - редагування інформації про користувача

Наступним пунктом меню є пункт Settings (рис. 3.2.1.6.).

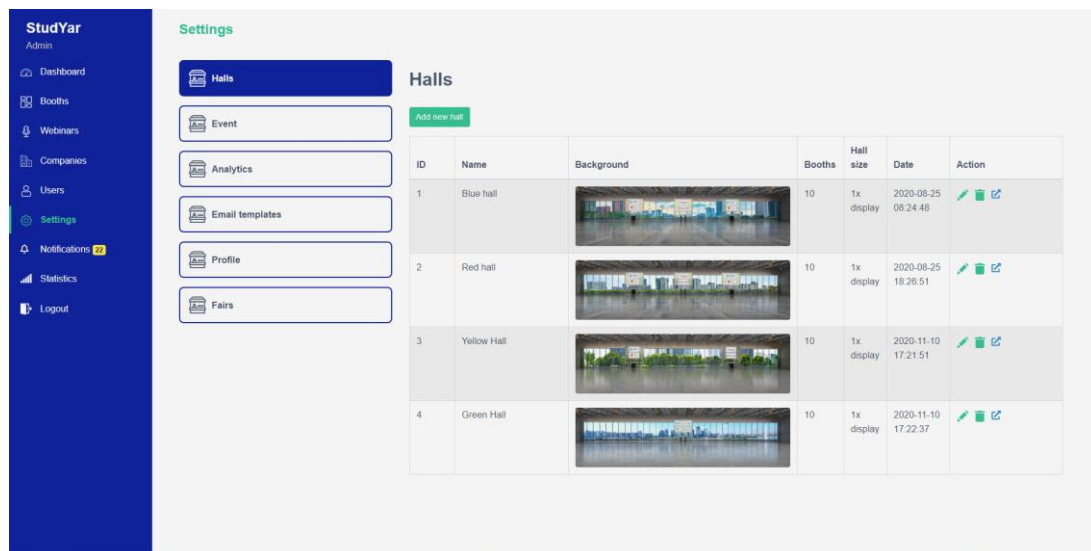


Рис. 3.2.1.6. Сторінка Settings

Першою вкладкою, на яку потрапляє користувач є вкладка Halls. На цій сторінці знаходиться кнопка ”Додати новий хол”. Натиснувши на неї ми заповнимо інформацію та виберемо потрібні параметри. Ці параметри та

інформація будуть показані на сторінці холів в таблиці. В ній знаходиться така інформація про холи, які були додані для ярмарки:

- ID – ідентифікатор хола в базі даних
- Name – назва ярмарки
- Background – зовнішній вигляд холу
- Booths – число стендів в цьому холі
- Hall size – розмір холу на екрані(1x display/2x display)
- Набір кнопок, який складається з:
  - видалення,
  - редагування інформації про хол,
  - перехід на хол.

Другим пунктом меню на сторінці Settings є пункт меню Event. В ньому адміністратор може вказати дату проведення заходу, для того щоб кожен користувач у своїй адміністративній панелі міг бачити початок проведення заходу за допомогою, так званого, зворотного відліку.

Третім пунктом меню на сторінці Settings є пункт меню Analytics. У ньому ми можемо додати скрипт, за допомогою якого наша платформа буде приєднана до аналітики Google Analytics.

Четвертим пунктом меню Settings є пункт меню Email Templates. Зайшовши в на сторінку ми можемо ввести правильні шаблони тексту, використовуючи такі змінні як:

- {{name}}
- {{company}}
- {{email}}
- {{password}}

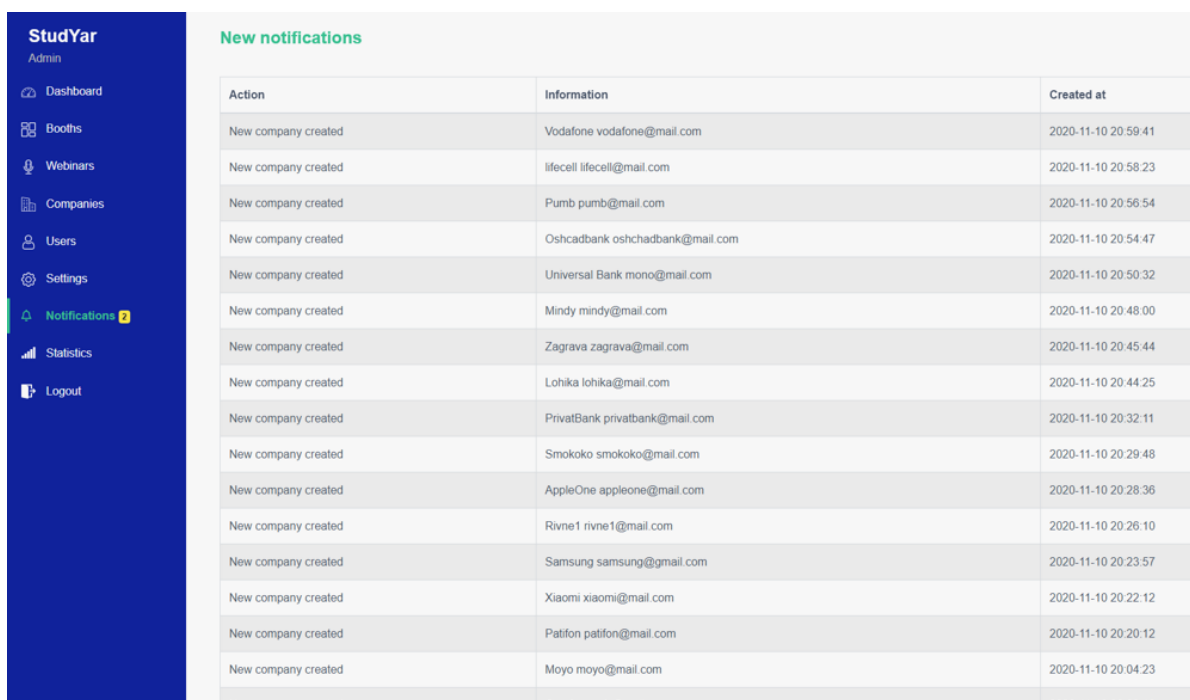
У налаштуваннях доступні такі шаблони відправки листів:

- User creation template – шаблон, який отримує користувач на пошту після створення свого акаунту
- Company creation template – шаблон, який отримує менеджер компанії на пошту після створення свого акаунту

- Candidate application – шаблон, який отримує менеджер компанії на пошту після отримання листа поданого студентом на доступну вакансію
- Forgot password template – шаблон, який отримує користувач на пошту після натискання клавiші Forgot password.

П'ятим пунктом меню Settings є пункт меню Profile. У ньому користувач може редагувати інформацію про себе, введену раніше при реєстрації.

Наступним пунктом меню адміністраторської панелі є пункт меню Notifications(рис. 3.2.1.7).



Action	Information	Created at
New company created	Vodafone vodafone@mail.com	2020-11-10 20:59:41
New company created	lifecell lifecell@mail.com	2020-11-10 20:58:23
New company created	Pumb pumb@mail.com	2020-11-10 20:56:54
New company created	Oshcadbank oshcadbank@mail.com	2020-11-10 20:54:47
New company created	Universal Bank mono@mail.com	2020-11-10 20:50:32
New company created	Mindy mindy@mail.com	2020-11-10 20:48:00
New company created	Zagrava zagrava@mail.com	2020-11-10 20:45:44
New company created	Lohika lohika@mail.com	2020-11-10 20:44:25
New company created	PrivatBank privatbank@mail.com	2020-11-10 20:32:11
New company created	Smokoko smokoko@mail.com	2020-11-10 20:29:48
New company created	AppleOne appleone@mail.com	2020-11-10 20:28:36
New company created	Rivne1 rivne1@mail.com	2020-11-10 20:26:10
New company created	Samsung samsung@gmail.com	2020-11-10 20:23:57
New company created	Xiaomi xiaomi@mail.com	2020-11-10 20:22:12
New company created	Patifon patifon@mail.com	2020-11-10 20:20:12
New company created	Moyo moyo@mail.com	2020-11-10 20:04:23
New company created	Conf.comf@mail.com	2020-11-10 20:03:43

Рис. 3.2.1.7. Сторінка Notifications

У ньому адміністратор може побачити інформацію про останні події, які зберігаються в таблиці з такими полями:

- Action
- Information
- Created at

Останнім пунктом меню є пункт Statistics (рис. 3.2.1.8).

Перейшовши на цю сторінку, адміністратор може бачити віджети, продубльовані для зручності зі сторінки Dashboard а саме:



- Кількість компаній
- Кількість активованих компаній
- Кількість неактивованих компаній
- Кількість учасників ярмарки
- Кількість стендів
- Кількість наявних вакансій на ярмарку
- Кількість користувачів онлайн

Під віджетами ми можемо бачити 3 елементи Tab під назвами: Event, Companies, Users. Першим табом, який бачить адміністратор є Companies. Вибравши потрібну наявну компанію в елементі Dropdown та дату в елементі DateTime та натиснувши кнопку Get отримаємо статистику по цій компанії.

Перейшовши на елемент Tab Event ми можемо бачити статистику по всьому заходу, а зайшовши на Tab Users та вибравши потрібну дату ми можемо бачити статистику за конкретну дату по користувачах (скрін)

Розглянувши панель адміністратора, перейдемо до опису адміністраторської панелі менеджера компанії.

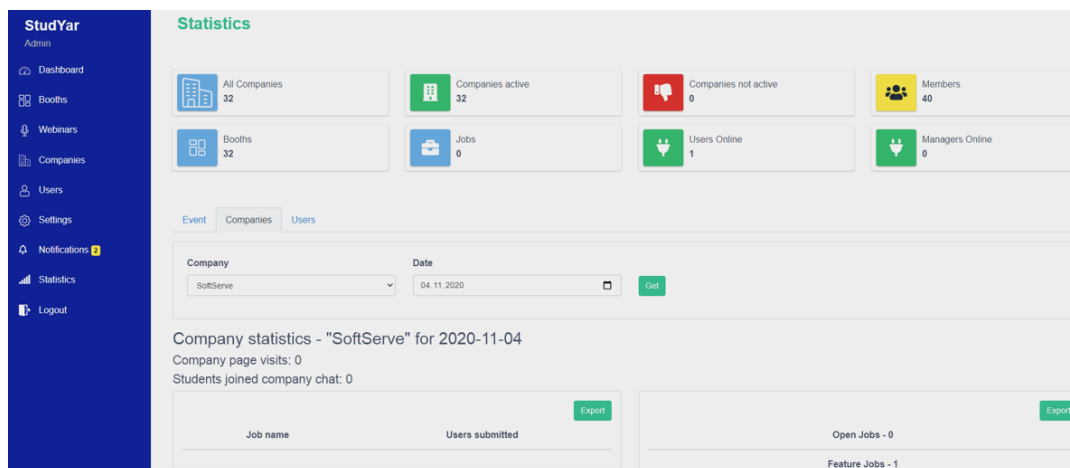


Рис. 3.2.1.8. Сторінка Statistics

### 3.2.2. Адміністраторська панель менеджера компанії

Перше, що бачить менеджер компанії, перейшовши в свою адміністраторську панель це пункт Dashboard, в якій є чотири віджети статистики про його компанію, а саме:

- Візити за сьогодні
- Кількість відкритих вакансій
- Всього візитів
- Кількість користувачів, які подали свою кандидатуру

Також під віджетами можемо бачити кнопку Company booth, натиснувши на яку одразу можемо перейти на сторінку стенду.

Другим пунктом адміністраторської панелі менеджера компанії є пункт меню Profile. В ньому менеджер може редагувати свою особисту інформація, таку як:

- First name
- Last Name
- Email
- Password
- Phone
- Gender

Пункт меню Booth. У цьому пункті меню менеджер може редагувати інформацію про стенд його компанії. Перейшовши на цей пункт менеджер може бачити 4 таби, кожен з яких описує крок заповнення інформації про стенд:

- 1 step. Basic information. У цьому кроці менеджер заповнює таку інформацію як: назва стенду, опис стенду, хол, в якому буде знаходитися стенд(рис. ).
- 2 step. Booth graphic. У цьому кроці менеджер обирає зовнішній вигляд стенда. Він може скористатися дизайнами стендів, які вже наявні в системі, або ж завантажити свій власний.

- 3 step. Jobs. Тут менеджер може вибрати тип подачі вакансії – актуальні вакансії чи вакансії, які плануються на майбутнє.
- 4 step. Finalize. Підтвердження.

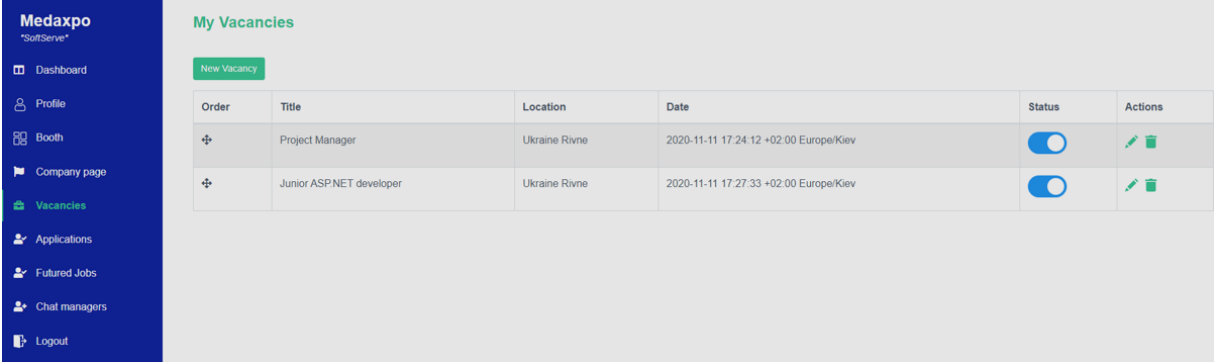
Пункт меню Company page. У цьому пункті меню можна заповнювати та редагувати інформацію про компанію, яка буде показана при візиті на стенд компанії. Тут доступні такі поля для заповнення інформацією про компанію:

- Назва компанії
- Місцезнаходження компанії
- Логотип компанії
- Зовнішній вигляд стенда
- Співвідношення статей в компанії
- Сайт компанії
- Опис компанії (перша частина)
- Фото для опису (перша частина)
- Опис компанії (друга частина)
- Фото для опису (друга частина)
- Посилання на Youtube канал
- Індустрія компанії
- Рік заснування
- Дохід компанії

Пункт меню Vacancies (рис. 3.2.2.1 ). У цьому пункті меню менеджер компанії може створювати нові вакансії за допомогою кнопки “New vacancy” або ж переглядати вже створені вакансії в таблиці з такими полями:

- Order(змiна порядку виведення вакансії)
- Title
- Location
- Date
- Status(відкрита/закрита)

- Редагування та видалення вакансії



Order	Title	Location	Date	Status	Actions
+	Project Manager	Ukraine Rivne	2020-11-11 17:24:12 +02:00 Europe/Kiev	<input checked="" type="checkbox"/>	<a href="#">Edit</a> <a href="#">Delete</a>
+	Junior ASP.NET developer	Ukraine Rivne	2020-11-11 17:27:33 +02:00 Europe/Kiev	<input checked="" type="checkbox"/>	<a href="#">Edit</a> <a href="#">Delete</a>

Рис. 3.2.2.1. Сторінка Vacancies

Пункт меню Applications. У цьому пункті меню менеджер компанії, може отримати інформацію про те, які користувачі подали заявку на вакансію за допомогою таблиці з такими полями:

- Name
- Jobs
- Apply at

Пункт меню Featured job. Цей пункт меню розрахований на вивід інформації про подання користувачів на вакансію поточної компанії. Таблиця включає такі поля:

- Field of study
- User
- CV

Пункт меню Chat managers. У цьому пункті меню менеджер компанії може додавати так званих менеджерів чату компанії. Стосовно ролей в базі даних, це є звичайний користувач, який має змогу спілкуватись від імені компанії. Для додавання такого менеджера потрібно натиснути “Add manager” та ввести ім’я, електронну пошту та пароль для цього користувача.


### 3.2.3. Адміністраторська панель користувача

Перше що бачить користувач, зайшовши на свою адміністраторську панель це профіль. У ньому можна редагувати всю інформацію, а саме:

- First Name
- Last Name
- Email
- Password
- Phone
- Gender
- Name of University
- Field of study
- GPA
- CV

Другим пунктом меню адміністраторської частини користувача є пункт Favourites. У цьому пункті меню ми можемо бачити вакансії, які користувач додав до категорії “Обрані ” або Swag bag.

Останнім пунктом меню є пункт My applications. Тут користувач може переглянути, вакансії на які він відправив свою кандидатуру.



Title	Location	Category	Time	Date apply
Project Manager	Ukraine Rivne	IT - industry	Full Time	2020-11-25 13:51:47 +02:00 Europe/Kiev

Рис. 3.2.3.1. Сторінка My Applications

### 3.3 Огляд ярмарки

Виконавши вхід в систему до початку ярмарки, система направляє користувача на адміністративну панель за замовчуванням. Натиснувши на

логотип компанії, перейдемо на головну сторінку. Перше, що бачить користувач у цьому випадку, це так званий парк(рис. 3.3.1).

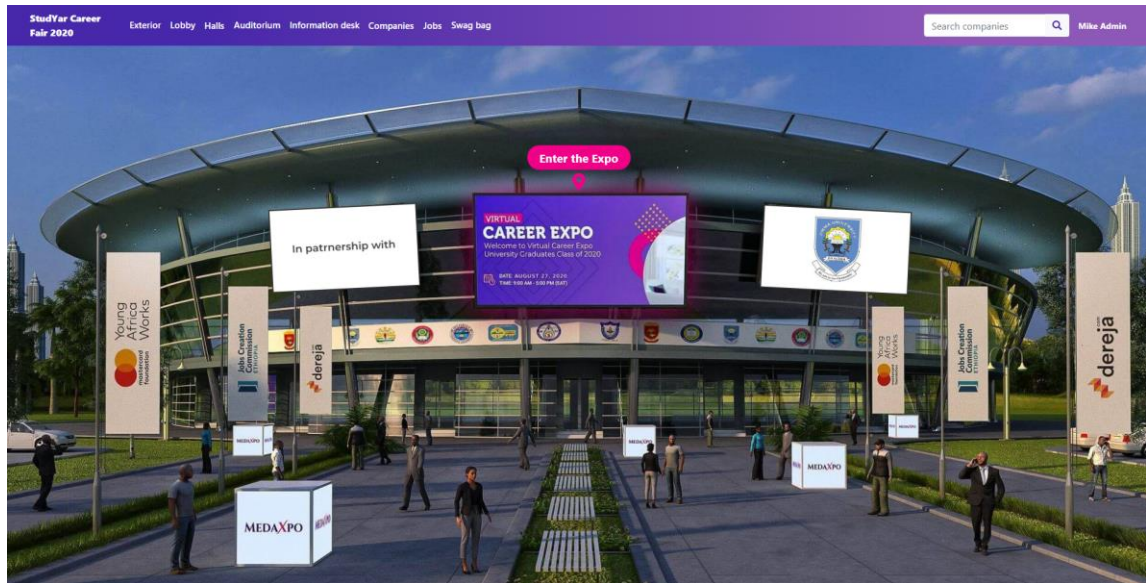


Рис. 3.3.1. Сторінка Park

Також у верхній частині сторінки може бачити такі посилання:

- Exterior
- Lobby
- Halls
- Auditorium
- Information desk
- Companies
- Jobs
- Swag bag

Розглянемо кожну сторінку окремо. Перше, що бачить користувач, це сторінка Exterior. На цій сторінці бачимо місце проведення ярмарку. Натиснувши на табло з написом "Enter the Expo" на будівлі ярмарку, перейдемо в так зване lobby (з англ. коридор).

Пункт меню Lobby. На цій сторінці перед нами є чотири посилання, кожне з яких відповідає за перенаправлення на один з чотирьох холів (рис.3.3.2 ).

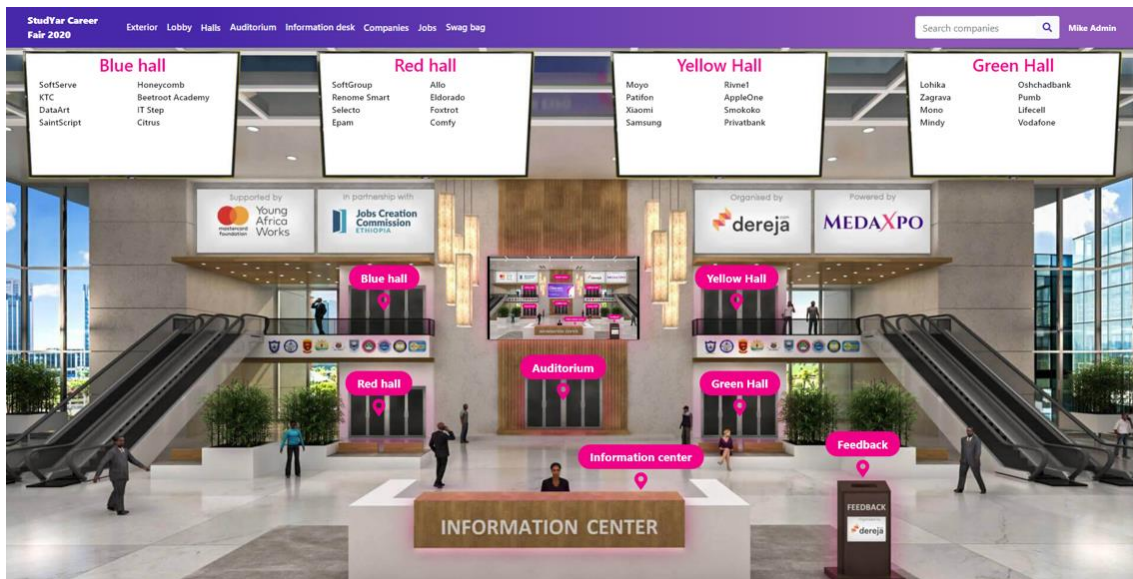


Рис. 3.3.2. Сторінка Lobby

Також ми можемо бачити стенд з кнопкою Feedback, перейшовши на який користувач може залишити в системі відгук. На цій сторінці наявний також стенд Information desk, перейшовши на який, побачимо інформацію про ярмарок. Також посередині бачимо посилання на сторінку Auditorium, яку розглянемо згодом.

Пункт меню Halls. Натиснувши на це посилання в елементі dropdown бачимо перелік всіх холів. Натиснемо на перший та розглянемо його. Перейшовши на дану сторінку ми бачимо великий простір, заповнений стендами з компаніями(рис. 3.3.3 ).

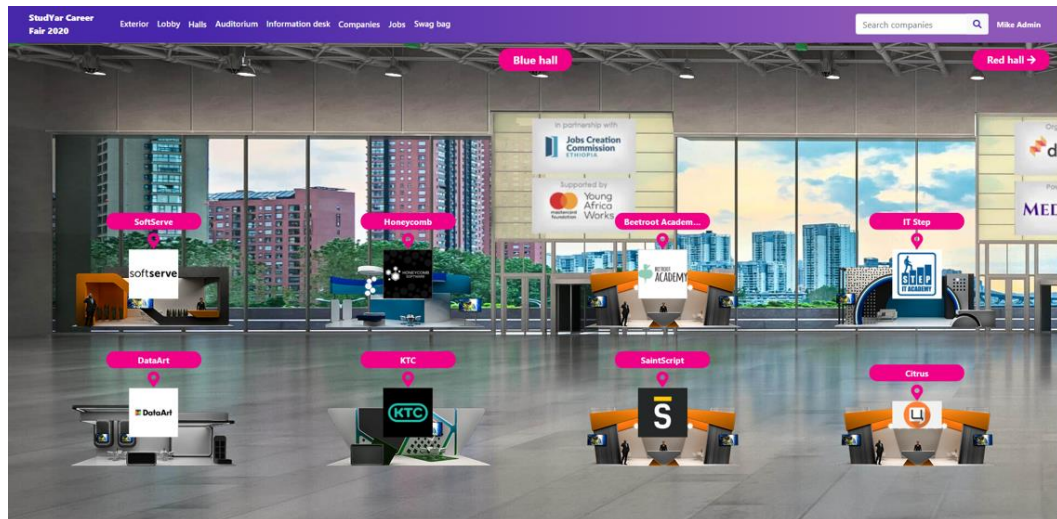


Рис. 3.3.3. Сторінка Halls

Перейдемо на один з них. На дизайні стенду бачимо так звані елементи pin, натиснувши на які, система буде перенаправляти нас на вміст компанії:

- Visit company's site
- Open a chat
- Current job openings
- Read about the company

Розглянемо функціонал чату з компанією за сценарієм користувача з найменшою роллю - юзером.

Зайшовши в чат з компанією додамо цю компанію собі в чат. Тепер ми можемо писати в груповий чат з іншими користувачами менеджера цієї компанії. Для того щоб поспілкуватись з менеджером компанії наодинці або ж провести відеодзвінок, натиснемо на іконку менеджера та натиснемо на кнопку Request direct chat (рис. 3.3.4).



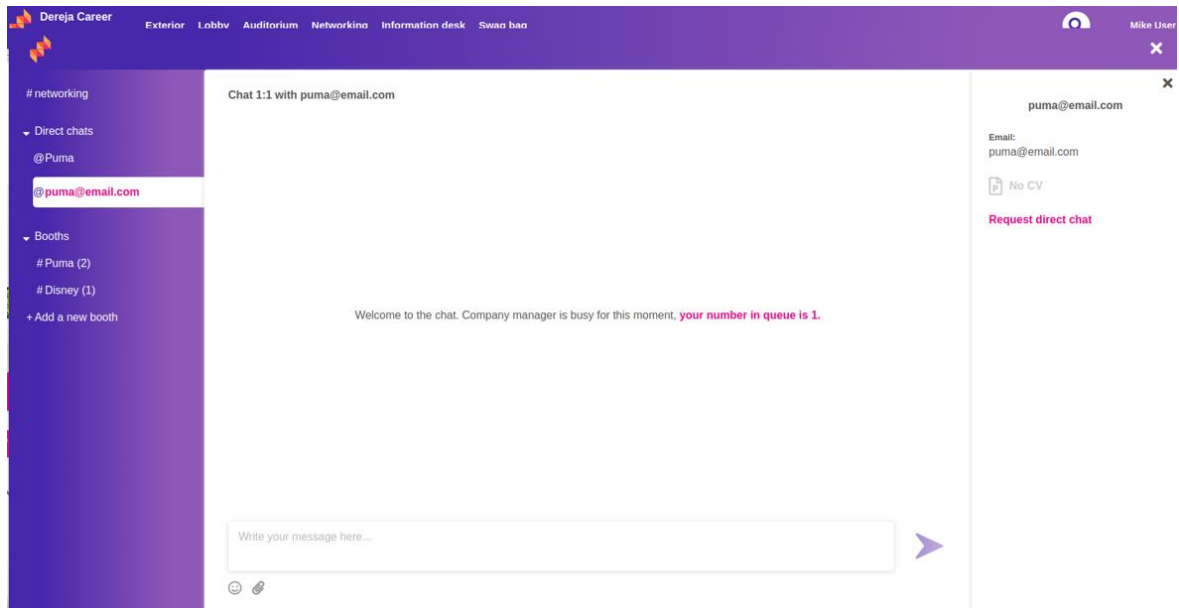


Рис. 3.3.4. Сторінка чату користувача після запиту менеджеру компанії

Після цього менеджер компанії може надати право писати та використовувати відеодзвінок, ти самим оберігаючи себе від можливого спаму.

Пункт меню Auditorium. На цій сторінці відбуваються вебінари для студентів та відвідувачів компанії (рис. 3.3.5 ).

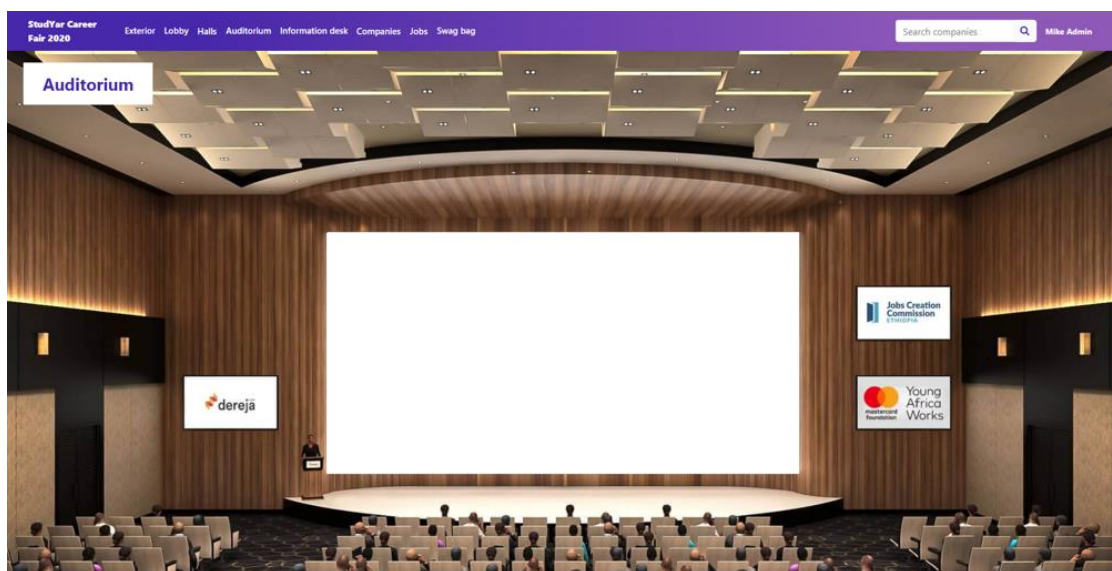


Рис. 3.3.5. Сторінка Auditorium

Знизу ми можемо бачити розклад вебінарів, який розмістив адміністратор за допомогою адміністративної панелі.

Пункт меню Information desk (рис. 3.3.6).

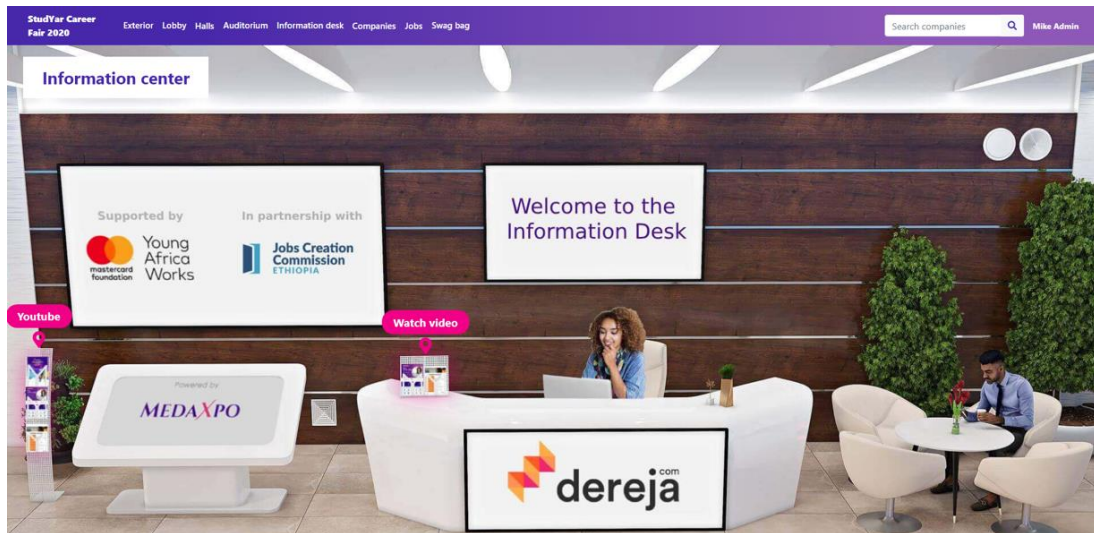


Рис. 3.3.6. Сторінка Information desk

На цій сторінці користувач може побачити багато корисних посилань, такі як YouTube, які описують ярмарку.

Пункт меню Companies. У цьому розділі наявний повний список компаній. При натисканні на назву однієї з компаній користувача перенаправить на стенд цієї компанії.

Пункт меню Jobs. У цьому розділі наявний повний список всіх доступних вакансій. При натисканні на назву однієї з вакансій користувача перенаправить на вакансію, яка була опублікована конкретною компанією.

Пункт меню Swag bag. Розділ в якому користувач може побачити вакансії, які він помітив як “обране” (рис. 3.3.7).

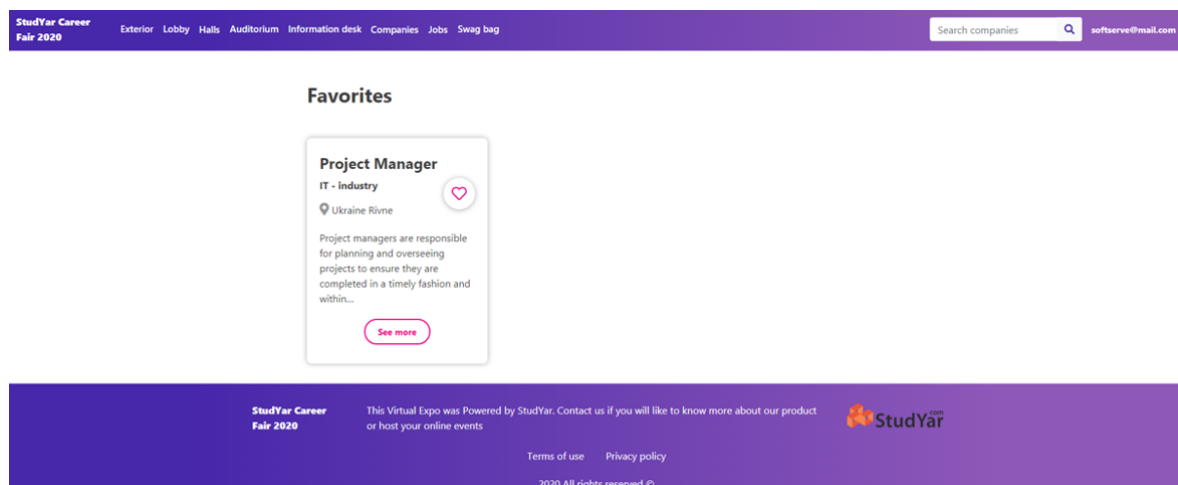


Рис. 3.3.7. Сторінка Swag Bag

## ВИСНОВКИ

Потреба у створенні платформ для проведення віртуальних ярмарок була завжди, але ситуація з карантинними заходами у світі стала товчком для написання магістерської роботи за темою «Використання інформаційних технологій для створення віртуальних ярмарок». Це пов'язано з постійною потребою в спілкуванні та комунікації роботодавців та молодих спеціалістів.

Недоліки в досліджених аналогах:

- застарілий інтерфейс користувача та адміністратора;
- відсутність можливості проведення вебінарів;
- платне використання;

У процесі поетапного виконання роботи було:

- узгоджено вимоги та підібрано інструментарій для реалізації програмного продукту;
- спроектовано та розроблено базу даних;
- здійснено UX-проекування та розроблено вдосконалений UI-дизайн;
- розроблено клієнтську частину та адміністративну панель платформи.
- реалізована можливість комунікації з використанням чатів, відеодзвінків та вебінарів.

Розроблена платформа спростить організацію заходів, на кшталт так звані “дні кар’єри” або “ярмарки кар’єри” , та спростить пошук кваліфікованих кадрів роботодавцями. У свою чергу студенти отримають актуальну інформацію про те, в якому напрямку потрібно розвиватись задля досягнення поставлених цілей.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Что должен знать junior Front-end разработчик в 2020 году [Электронный ресурс] – Режим доступа: <https://dan-it.com.ua/chto-nuzhno-znat-junior-front-end-razrabotchiku-v-2020-godu/>.
2. What is HTML [Электронный ресурс] – Режим доступа: <https://www.educba.com/what-is-html/>.
3. Advantages and disadvantages of CSS [Электронный ресурс] – Режим доступа: <https://www.motocms.com/blog/en/advantages-and-disadvantages-of-css/>.
4. Javascript tutorial [Электронный ресурс] – Режим доступа: <https://www.tutorialspoint.com/javascript/index.htm>.
5. Framework [Электронный ресурс] – Режим доступа: <https://techterms.com/definition/framework>.
6. Laravel - Application Structure [Электронный ресурс] – Режим доступа: [https://www.tutorialspoint.com/laravel/laravel\\_application\\_structure.htm](https://www.tutorialspoint.com/laravel/laravel_application_structure.htm)
7. What is framework [Электронный ресурс] – Режим доступа: <https://hackr.io/blog/what-is-frameworks>.
8. What is Bootstrap and How to Use it in Web Development? [Электронный ресурс] – Режим доступа: <https://f5-studio.com/articles/what-is-bootstrap-and-how-to-use-it-in-web-development/>.
9. Creating A Bootstrap Admin Template With Charts [Электронный ресурс] – Режим доступа: <https://blog.zingchart.com/creating-a-bootstrap-admin-template-with-charts/>
10. Bootstrap Grid System [Электронный ресурс] – Режим доступа: [https://www.w3schools.com/bootstrap/bootstrap\\_grid\\_system.asp](https://www.w3schools.com/bootstrap/bootstrap_grid_system.asp)
11. Bootstrap 3 grid system explained and Grid System Comparison: Bootstrap 3 vs. Foundation 5 [Электронный ресурс] – Режим доступа: <http://mydailyyy.blogspot.com/2016/03/bootstrap-3-grid-system-explained-and.html>

12. Learn Bootstrap: Main Concepts and Principles Explained [Электронный ресурс] – Режим доступа: <https://www.bitdegree.org/learn/learn-bootstrap>.
13. An overview of Vue.js [Электронный ресурс] – Режим доступа: <https://flaviocopes.com/vue-introduction/#why-developers-love-vue>.
14. The HTML DOM (Document Object Model) [Электронный ресурс] – Режим доступа: [https://www.w3schools.com/js/js\\_htmlDOM.asp](https://www.w3schools.com/js/js_htmlDOM.asp)
15. The Good and the Bad of Vue.js Framework Programming [Электронный ресурс] – Режим доступа: <https://www.altexsoft.com/blog/engineering/pros-and-cons-of-vue-js/>.
16. What does a back-end developer do? [Электронный ресурс] – Режим доступа: <https://www.careerexplorer.com/careers/back-end-developer/>.
17. Laravel overview [Электронный ресурс] – Режим доступа: <https://w3points.com/laravel-overview/>.
18. Laravel - overview [Электронный ресурс] – Режим доступа: <https://atsoftware.com.vn/blog/laravel-overview/>.
19. Этапы UX-проектирования: четыре шага для создания прототипа интерфейса [Электронный ресурс] – Режим доступа: <https://livetyping.com/ru/blog/proektirovanie-interfeisov-v-live-typing>
20. Wireframe sketch [Электронный ресурс] – Режим доступа: <https://no.pinterest.com/pin/14144186299713208/>
21. Prototype [Электронный ресурс] – Режим доступа: <https://no.pinterest.com/pin/14144186299713208/>
22. Figma для недизайнеров: горячие клавиши и фишки для совместной работы [Электронный ресурс] – Режим доступа: <https://www.ratio.digital/journal/figma-for-undesigners/>
23. Feature of Brackets Text Editor [Электронный ресурс] – Режим доступа: <https://filehulk.com/download-bracket-text-editor/>
24. Eloquent: Начало работы [Электронный ресурс] – Режим доступа: <https://laravel.su/docs/5.4/eloquent>