

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Казахский национальный исследовательский технический университет имени
К.И.Сатпаева

Институт автоматизации и информационных технологий

Кафедра "Программная инженерия"

Махсумов Ильяр Мухаметжанулы

Разработка веб-сайта для ТОО «Kadeen»

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к дипломному проекту

Специальность 5В070400 – Вычислительная техника и программное
обеспечение

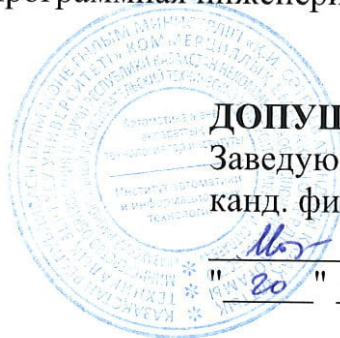
Алматы 2022

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Казахский национальный исследовательский технический университет имени
К.И.Сатпаева

Институт автоматизации и информационных технологий

Кафедра "Программная инженерия"



ДОПУЩЕН К ЗАЩИТЕ

Заведующий кафедрой ПИ

канд. физ-мат. наук, профессор

Молдагулова А. Н. Молдагулова

" 20 " 05 2022 г.

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к дипломному проекту

На тему: "Разработка веб-сайта для ТОО «Kadeen»"

по специальности 5В070400 – Вычислительная техника и программное
обеспечение

Выполнил

Махсумов И.М.

Рецензент

канд. техн. наук, Доктор Ph.D.

Кадырович Д.Д. Кадырович

" 20 " мая 2022 г.

Научный руководитель

Сениор-лектор, Доктор Ph.D.

Черикбаева Л.Ш. Черикбаева Л.Ш.

" 20 " мая 2022 г.

Алматы 2022

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Казахский национальный исследовательский технический университет имени
К.И.Сатпаева

Институт автоматизации и информационных технологий

Кафедра "Программная инженерия"

5B070400 – Вычислительная техника и программное обеспечение



УТВЕРЖДАЮ

Заведующий кафедрой ПИ
канд. физ-мат. наук, профессор

А. Н. Молдагулова
" 20 " 05 2022 г.

ЗАДАНИЕ

на выполнение дипломного проекта

Обучающемуся Махсумов Ильяр Мухаметжанулы

Тема: Разработка веб-сайта для ТОО «Kadeen»

Утверждена приказом проректора по академической работе № 489-П/Ө
от "24" декабря 2021 г.
"25" мая 2022 г.

Срок сдачи законченного проекта

Исходные данные к дипломному проекту: Описание компании, наименование компании, короткое описание требований

Перечень подлежащих разработке в дипломном проекте вопросов:

а) Анализ требований и технологий

б) Разработка сайта

в) Построение базы данных

г) Написание сервера



Перечень графического материала (с точным указанием обязательных чертежей): представлены 21 слайда презентации

Рекомендуемая основная литература: из 20 наименований

ГРАФИК
подготовки дипломного проекта

Наименование разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю и консультантам	Примечание
1. Определение требований от компании Kadeen	10.02.2022	Выполнено
2. Анализ аналогичных систем и разбор требований	15.02.2022	Выполнено
3. Составления плана разработки и определение технологий	25.02.2022	Выполнено
4. Разработка сайта	15.04.2022	Выполнено
5. Написание пояснительной записки к дипломному проекту	30.04.2022	Выполнено

Подписи
консультантов и нормоконтролера на законченный дипломный проект с указанием относящихся к ним разделов проекта

Наименования разделов	Консультанты, И.О.Ф. (уч. степень, звание)	Дата подписания	Подпись
Нормоконтролер	М.Н Жекамбаева Доктор Ph.D., ассоциированный профессор	18.05.22	
Программное обеспечение	Марғұлан Қ. Магистр техн.наук, лектор	18.05.22	

Научный руководитель  Ч. Л. Шариповна

Задание принял к исполнению обучающийся  Махсумов И.М.

Дата "14" 11 2021 г.

АННОТАЦИЯ

Данный дипломный проект предназначен для разработки веб-сайта для не большой компании “Kadeen” с целью расширения компании и предоставления своих услуг онлайн через веб-сайт.

В дипломной работе описывается процесс разработки, проектирование системы. Был проделан путь по анализу аналогичных веб-сайтов и компаний работающих в одной сфере деятельности с компанией Kadeen. Описываются и технологии разработки, в частности фреймворк Angular и почему именно этот фреймворк был выбран для веб-сайта. Про серверную часть, которая написана на Node js сервере Express, так же немного про СУБД MySql.

Конечный продукт веб-сайт “kadeen.kz” в будущем позволяет пользователем получать дополнительные знания в сфере IT и не только. Таким образом компания сможет увеличить свою клиентскую базу и соответственно получить больше прибыли.

АҢДАТПА

Бұл дипломдық жоба компанияны кеңейту және веб-сайт арқылы өз қызметтерін онлайн ұсыну мақсатында "Kadeen" компаниясының веб-сайтын әзірлеуге арналған.

Дипломдық жұмыста веб-сайтты әзірлеу процесі, жүйені жобалау сипатталған. Ұқсас веб-сайттар мен Kadeen компаниясымен бір қызмет саласында жұмыс істейтін компанияларды талдау бойынша жол жасалды. Әзірлеу технологиялары да сипатталған, атап айтқанда Angular фреймворк және неге бұл фреймворк веб-сайт үшін таңдалды. Node Js тын Express жақтауында жазылған сервер және MySQL ДҚБЖ туралы да айтылып өтті.

Соңғы өнім веб-сайт "kadeen.kz" болашақта пайдаланушыға IT саласында ғана емес, қосымша білім алуға мүмкіндік береді. Осылайша, компания өзінің клиенттік базасын көбейтіп, сәйкесінше көбірек пайда таба алады.

ANNOTATION

This diploma project is designed to develop a website for a small company “Kadeen” in order to expand the company and provide its services online through the website.

The thesis describes the process of development, system design. A long way has been done to analyze similar websites and companies working in the same field of activity with Kadeen. The development technologies are also described, in particular the Angular framework and why this particular framework was chosen for the website. About the server part, which is written in NodeJS Express, also a little bit about the MySQL database.

The final product is the “kadeen.kz” in the future allows the user to gain additional knowledge in the field of IT and not only. Thus, the company will be able to increase its customer base and, accordingly, get more profit.

СОДЕРЖАНИЕ

	Введение	9
1	Общее описание	10
1.1	Цель разработки веб-сайта	10
1.2	Анализ аналогичных систем	10
1.3	Определения, термины и сокращения	11
2	Технологический раздел	13
2.1	Angular Framework	13
2.2	Express js	14
2.3	MySQL Database	14
2.4	Продукты JetBrains или почему IDE так важна	14
2.5	Система контроля версий	15
3	Проектная часть	16
3.1	Архитектура проекта	16
3.2	UML Diagrams	16
3.2.1	Use case диаграмма	17
3.2.2	ER- диаграмма	17
3.2.3	Диаграмма активности	18
3.3	UX/UI	19
4	Экспериментальная часть	21
4.1	Главная страница	21
4.2	Курсы	22
4.3	Дорожная карта	24
4.4	Менторы	24
4.5	Создание описаний и динамические компоненты	25
4.6	Структура проекта	25
4.7	Дополнительные пакеты и фреймворки	27
	Заключение	30
	Список использованной литературы	31
	Приложения А (Техническое задание)	32
	Приложения Б (Текст программы)	34

ВВЕДЕНИЕ

Абсолютно все компании в нынешнем XXI веке не представляют ведение своего бизнеса без веб-сайта. Все больше и больше предприятий стремятся создать свой собственный сайт для расширения бизнеса до мировых масштабов. Вы, наверное, спросите *-А такое разве возможно?!*, да вполне возможно. Как школьник, снимая короткие видео в тик-токе смог стать знаменитым на всю страну, так и компания, которая имеет хороший веб-сайт, не может не привлекать внимание среди остальных тысячи компаний оказывающих точно такие же услуги.

ТОО «Kadeen» имеет отличные возможности для развития. Команда, состоящая из профессионалов своего дела которые готовы растить новое поколение айти специалистов, хочет все больше продвигаться вперед.

Веб сайт в первую очередь нацелен на молодых людей до 30 лет, что бы они смогли пройти весь путь по становлению специалистов в области информационных технологий находясь на одном сайте. Это даст им возможность получать постоянную поддержку со стороны сайта и менторов. Студенты возможность учиться онлайн и при желании посещать курсы, которые будут проводиться в оффлайн.

Для компаний важно воспитывать для себя будущих работников, так множество компаний регулярно проводит обучение для студентов и всех желающих. Но насколько это будет затратно для самой компании? Искать из тысячи желающих, отбирать среди них максимум 30 и обучать студентов их на протяжении нескольких месяцев. И не факт, что студенты пойдут работать именно в эту компанию. Kadeen предлагает способ избавиться от лишних затрат и взять на себе роль обучения новых специалистов. Студенты после прохождения наших курсов, могут проявить желание работать в компаниях партнерах. Таким образом Kadeen будет обеспечивать партнеров новыми сотрудниками, а студентам трудоустройство.

1 Общее описание

1.1 Цель разработки веб-сайта

Тема дипломного проекта ясно дает понять цели дипломной работы. Разработка веб-сайта для компании Kadeen. В итоге иметь продукт, через который компания будет предоставлять свои услуги по всему миру. Веб-сайт нацелен на то, чтобы привлекать все больше пользователей, и обеспечивать компанию непрерывным потоком клиентов.

Клиентами, в частности, являются студенты и молодые люди, которые ищут новые знания или же специалисты, которые хотят сменить свою работу и начать карьеру в сфере информационных технологий. Так же сайт доступен и для тех, кто готов учить других! Таким образом мы получаем не только потребителей информации, но и тех, кто создает контент.

Сайт должен привлекать внимание клиента своей простотой и в то же время быть максимально сложной. Так в глазах пользователя он будет выглядеть не навязчивым, спокойным и информативным.

1.2 Анализ аналогичных сайтов

В мире существует множество компаний предоставляющие аналогичные услуги в сфере образования. С многими из них вы возможно сталкивались по несколько раз. Например, популярными являются сайты как Coursera, Udemu, Stepik предоставляют свои курсы онлайн, а Attractor school, Академия Шаг, Decode имеют свои офисы в Алматы и обучают оффлайн.

Давай те разберем почему наш сайт будет лучше и круче чем другие. Начнем с первых трех. У этих сайтом одно единое направление, это – обучить онлайн. Получается пользователи получают знания на сайте и больше не получают никакой помощи для дальнейшего продвижения. Но их плюс в том что пользователь получает некоторые основные знания, с помощью которым они с малой вероятностью получают работу. В этом и заключается минут полного онлайн обучение, я считаю, что студенту необходимо получать Soft skills не посредственное в команде вместе с другими студентами. Разбирая вторую тройку, акцентируем внимание непосредственно на их сайты. Академия Шаг имеет большую популярность, но при этом на их сайте детализация курса разбросана и найти эти самые курсы оказалось немного сложно даже для меня, уверенного пользователя ПК. Так же отсутствуют отзывы от студентов, все мы хотим знать больше о курсе и как он проводиться. Пользователю будет трудно покупать курс, после которого он не знает куда идти дальше, это, еще не говоря о том, что курсы слишком дорогие их прохождение может продлиться на несколько месяцев.

Attractor school – большинство замечаний что было у Академии Шаг можно отнести и к данному сайту. Порадовало то, что есть фотографии студентов. Но главная проблема так остается, это нету информации про сами курсы.

Decode. Данный сайт можно назвать основным нашим конкурентом, на сайте нету замечаний что было у предыдущих двух. Есть хорошая детализация про курсы, про средние заработные платы, про ход курса. Но отсутствуют отзывы студентов, которые прошли обучение. Так же обучение доступна только оффлайн.

Что в итоге нам потребуется сделать для сайта что бы быть лучше, чем другие. Начнем, пожалуй, с самих курсов, как мы читали выше:

- курсы должны быть четко описаны, пользователь должен знать куда идти после прохождении курса;
- дать возможность учиться онлайн;
- отзывы студентов очень важны;
- все необходимая информация должна быть на видном месте.

1.3 Определения, термины и сокращения

В таблице 1 сформулированы все термины и сокращения, которые использовались для разработки веб-сайта, а так же специфические термины связанные с используемыми технологиями.

Таблица 1 – сокращения, термины и их описание

Сокращение или термин	Описание
HTML	Hypertext markup language
CSS	Cascading style sheets
Typescript	Язык программирования
Модуль	Часть приложения с определенной логикой
Компонент	Самая маленькая единица в Angular app
Git	Система контроля версий
Gitlab	Удаленная система контроля версий
Фреймворк	Технология разработки
HTTP	HyperText Transfer Protocol
UX/UI	User Experience/ User interface
ER	Entity Relationship
AOT	Ahead-of-time
PWA	Progressive web application
БД	База данных

Продолжения таблицы 1

API	Application programming interface
IDE	Integrated development environment
SSL	Secure sockets layer
NPM	Менеджер пакетов для nodeJS
DRY	Принцип программирования, не повторяйся

2 Технологический раздел

В дипломной работе были использованы современные стек разработки веб сайтов Angular, ExpressJS и база данных MySQL. Так же активно использовался установщик NPM.

2.1 Angular Framework

Angular [1] один из трех самых популярных Javascript фреймворков отличающийся своей высокой производительностью и расширяемостью. Angular позволяет разделять код что в будущем при работе в больших системах не было конфликтов между компонентами. Так же данный фреймворк является кроссплатформенным что позволяет его использование в мобильных телефонах через PWA.

PWA – это технология позволяющая превратить веб сайт в приложение. Приложение запускается в обертке браузера и даже появляется возможность использовать сайт в офлайн режиме [2].

Данный фреймворк был выбран для дипломного проекта из-за его быстродействий и высокой производительности.

Архитектура приложений в Angular довольно-таки проста. Frontend приложения строиться на основе компонентов и модулей в Angular [3] (рисунок 2.1).

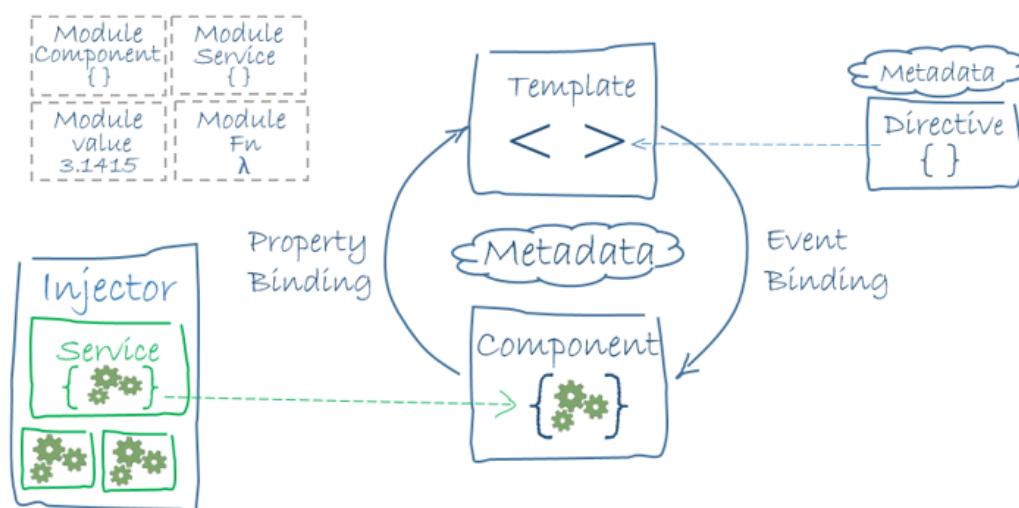


Рисунок-2.1 – Архитектура Angular приложений

Еще один плюс данного фреймворка это то что использует язык программирование Typescript. Typescript является суперсетом Javascript что позволяет нам в typescript – е использовать все функции которые есть в javascript. Так у языка есть собственные особенности как типизация, интерфейсы и т. д.

Фреймворк хорош тем, что компонентная архитектура позволяет дробить сайт на части. Это помогает не запутаться в коде и пере использовать компоненты что преследует принципы DRY. Модульность позволяет выносить в отдельную часть функционал сайта, например у вас большой сайт, где 5000 пользователей и несколько отделов. Для бухгалтеров отдельный модуль и для поставщиков отдельный модуль.

2.2 Express JS сервер

Express JS это быстрый, гибкий и минималистический веб фреймворк [4] Используется язык программирование Javascript. Выбрал данный фреймворк, потому что он очень прост в использовании и вот еще несколько его плюсов:

- Разработка сайта происходит намного быстрее по сравнению с другими фреймворками;
- Легко кастомизируется и настраивается;
- Позволяет нам создать REST Api;
- Легко установить связь с базами данных как MongoDB, MySQL и тд.

Но главной причиной почему я выбрал именно эту технологию является то, что он по структуре и языке программирования похож на Angular, что в дальнейшем упростит мне написание кода.

2.3 MySQL Database

MySQL – реляционная система управления базами данных разработанная корпорацией Oracle и использующая язык запросов SQL [5]. СУБД является одной самых популярных и может обеспечивать высокую производительность. Так же оно является бесплатным. MySQL отлично подходит для веб страниц.

2.4 Продукты JetBrains или почему IDE так важна

Все мы, когда пишем код, сталкиваемся теми или иными неудобствами. Во-первых, редактор может не удовлетворят наши личные предпочтения, во-вторых, он может быть ограниченным по функционалу что делает процесс разработки намного дольше, в-третьих, это количество и качество встроенных инструментов, вам может стать жутко неудобно, когда вам приходится использовать другие инструменты или вовсе делать вручную вещи которые могла бы сделать за вас ваша IDE.

WebStorm – интегрированная среда разработки на Javascript и связанных

веб технологий [6]. IDE позволяет нам использовать встроенный терминал для запуска приложения и имеет встроенную систему контроля версий.

DataGrip – инструмент созданный специально для SQL разработчиков [7]. Хорош тем что можем подключаться к различным базам данных без установление дополнительных плагинов.

В процессе были использованы и другие ПО для работы со специфической частью приложения, как Postman для проверки запросов API и GitHub Desktop для управление версиями

2.5 Система контроля версий

Для отслеживания разработки и хранения версий сайта используется система контроля версий Git. А удаленный сервер на Gitlab залит. Это удобно тем, что к проекту можно посмотреть с любых устройств, а также что бы делать откат если что-то пошло не так. Для проекта я веду гит с самого начало (рисунок 2.2) и в нескольких ветках. Основной на данный момент является ветка develop.

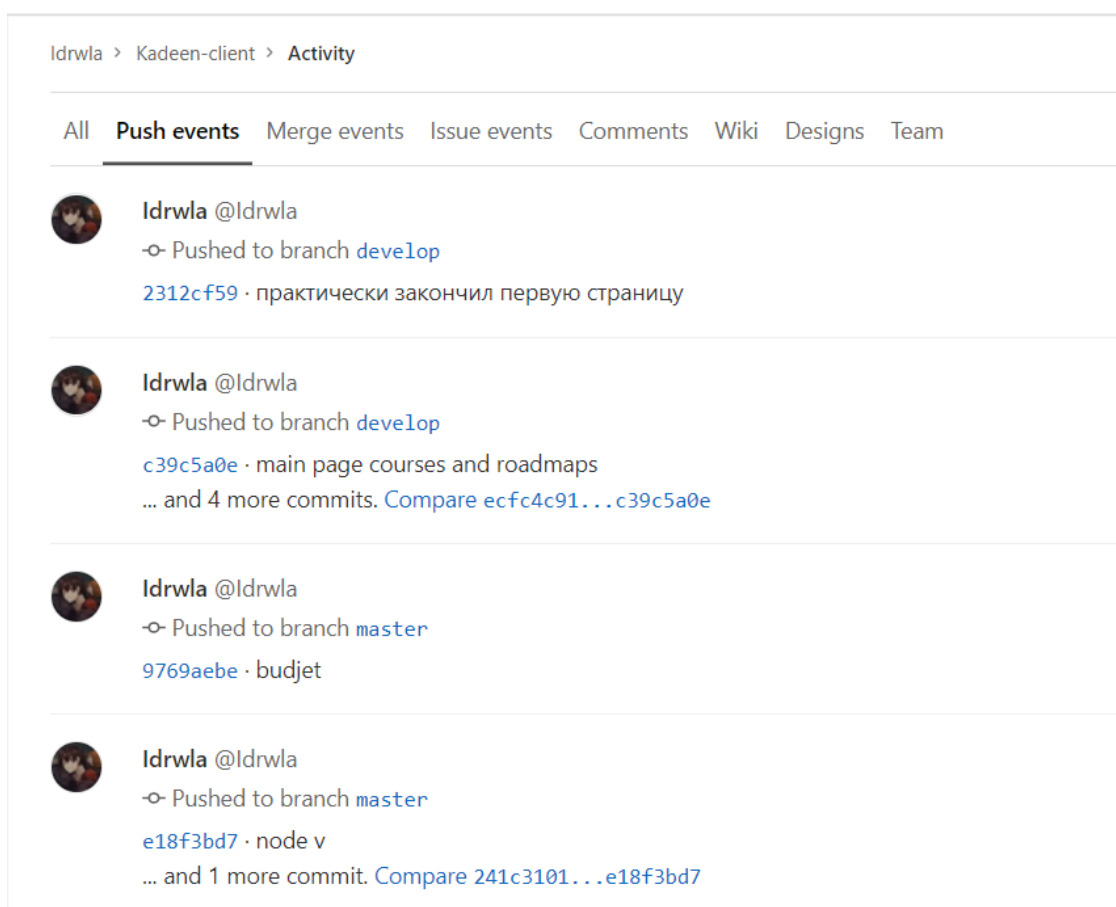


Рисунок-2.2 – Git events

3 Проектная часть

3.1 Архитектура проекта

В этом разделе мы более детально разберем архитектуру всего приложения, из каких частей оно состоит и все ли так просто. Вы, наверное, заметили, что тут присутствует nginx, он будет служить нам как балансировщик нагрузки, HTTP-кэша и прокси сервер [8].

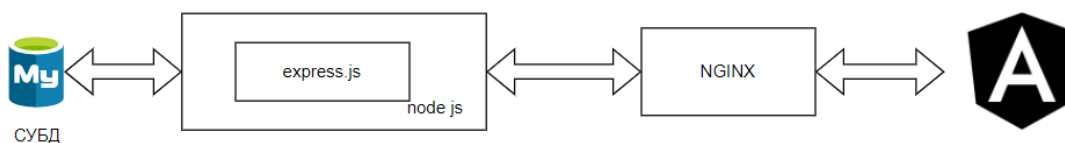


Рисунок-3.1 – Архитектура проекта

Архитектура может показаться довольно таки странной, но она построена с учетом того, чтобы иметь возможность внедрения сторонних сервисов. Например, компания планирует в будущем подключить сервис для приема платежей, все что потребуется это использовать один из текущих сервисов для работы с системой платежей. Так же стоит заметить, что между сервером Node и Angular есть Nginx [9], все верно это прокси сервер что позволит нам передавать пакеты предварительно зашифровав их.

Почему мне понадобился сервер Nginx, вот несколько ответов:

- Защищенность что позволит получить сертификат SSL;
- На сайте присутствует много статического контента (фото, тексты и тд);
- Nginx помогает распределять нагрузку на сервер;
- Делает сайт намного быстрее.

3.2 UML diagrams

UML – способ визуализации программы с помощью коллекции диаграмм [10]. В настоящее UML диаграммы состоят из 13 видов диаграмм. В дипломном проекты мы не будем рассматривать их всех, а только те которые будут относиться к проекту.

3.2.1 Use Case диаграмма

Диаграмма отражает отношение между актерами и прецедентами. Актерами являются наши клиенты, сотрудники, а прецеденты это основной функционал.



Рисунок-3.2 – Use Case диаграмма сайта

1.2.2 ER Диаграмма

Entity Relationship – диаграмма, показывающая отношение между сущностями в базе данных [11]. Диаграмма состоит из двух типов объектов – сущностей и отношений. Есть три вида отношений в базах данных

- Один к одному;
- Один ко многим;

– Многие ко многим.

В ER диаграмме сайта показываються основные сущности базе данных. На рисунке 3.2 показывается ER диаграмма сайта.

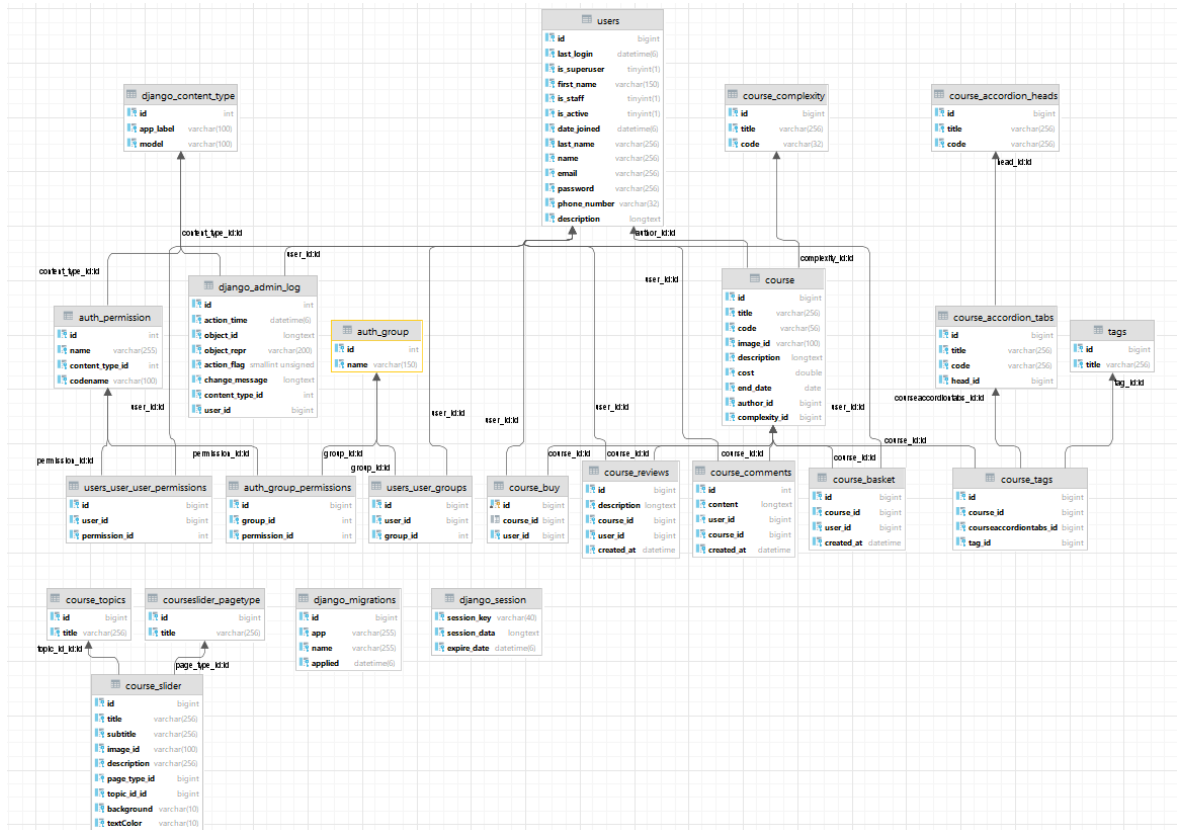


Рисунок-3.3 – ER Диаграмма сайта

В больших системах ER диаграмма становится не читаемой.

3.2.3 Диаграмма активности

Диаграмма активности показывает примерное поведение объекта в системе [12]. Для сайта обычно диаграммы бывают не такими сложными по сравнению с большими коммерческими системами.

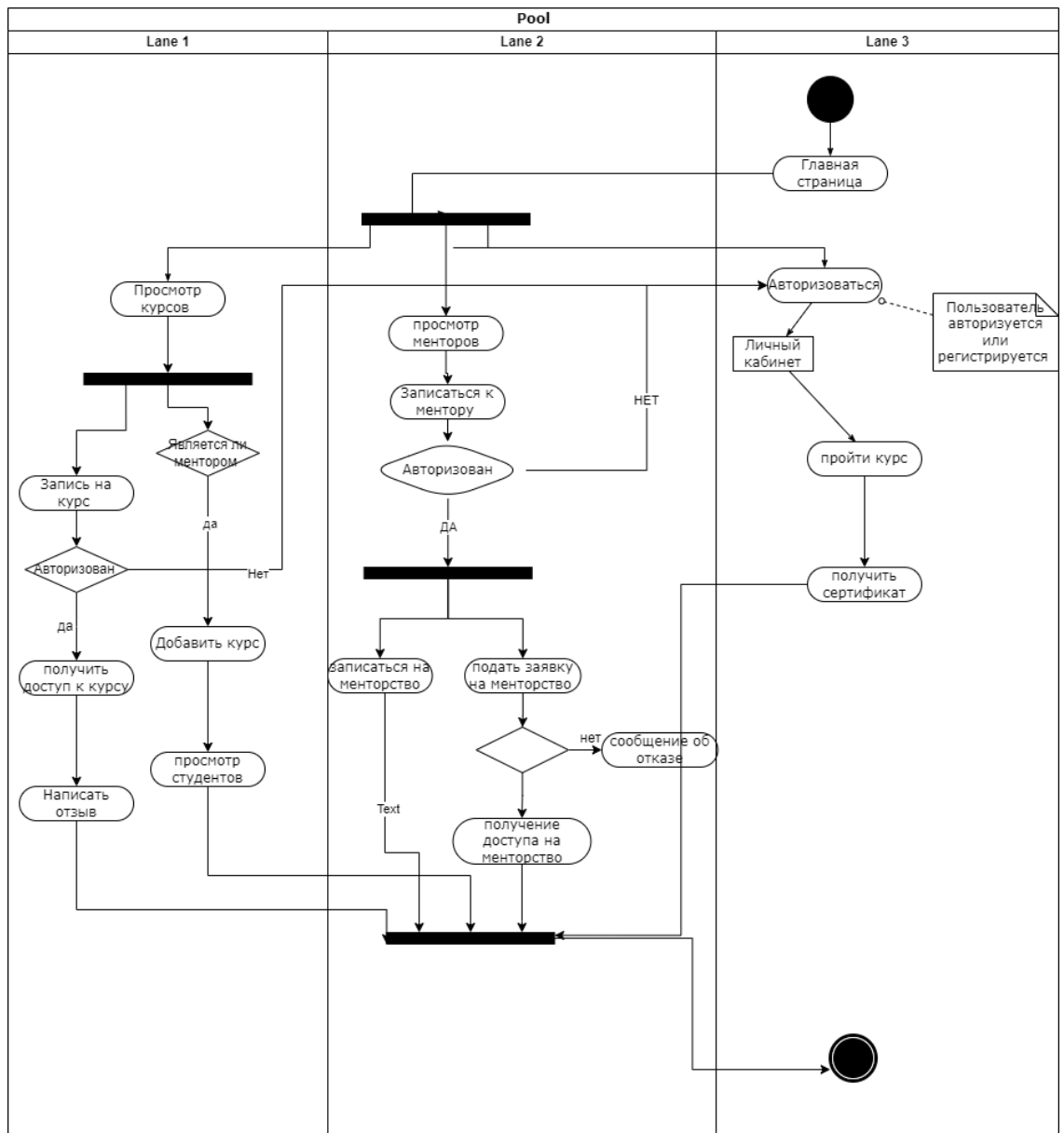


Рисунок-3.4 – Диаграмма активности сайта

3.3 UX/UI

UX/UI это основная составляющая дизайна сайта. Она является очень популярной среди дизайнеров и даже есть отдельная специальность UI/UX–дизайнера

UX – (User Experience) [13] дословно переводя опыт пользователя. Говорит нам о том, что какой опыт получит пользователь от нашей системы. Так же пользователь уже приходит к нас с опытом и важно, чтобы он смог свободно находиться на сайте.

UI – (User Interface) [14] пользовательский интерфейс. Включает в себя всю красоту сайта.

Важно не изобретать что-то новое, а придерживаться общепринятых стандартов пользовательского интерфейса.

У UX/UI есть свои предпочтения для тех или иных задач. Например, форма для заполнения (рисунок 3.5). Вы можете подумать, что это всего лишь пару полей которых нужно просто показать пользователю. Но не тут то было, нужно учитывать насколько эти поля удобно расположены, сочетаются ли выбранные цвета и тд .

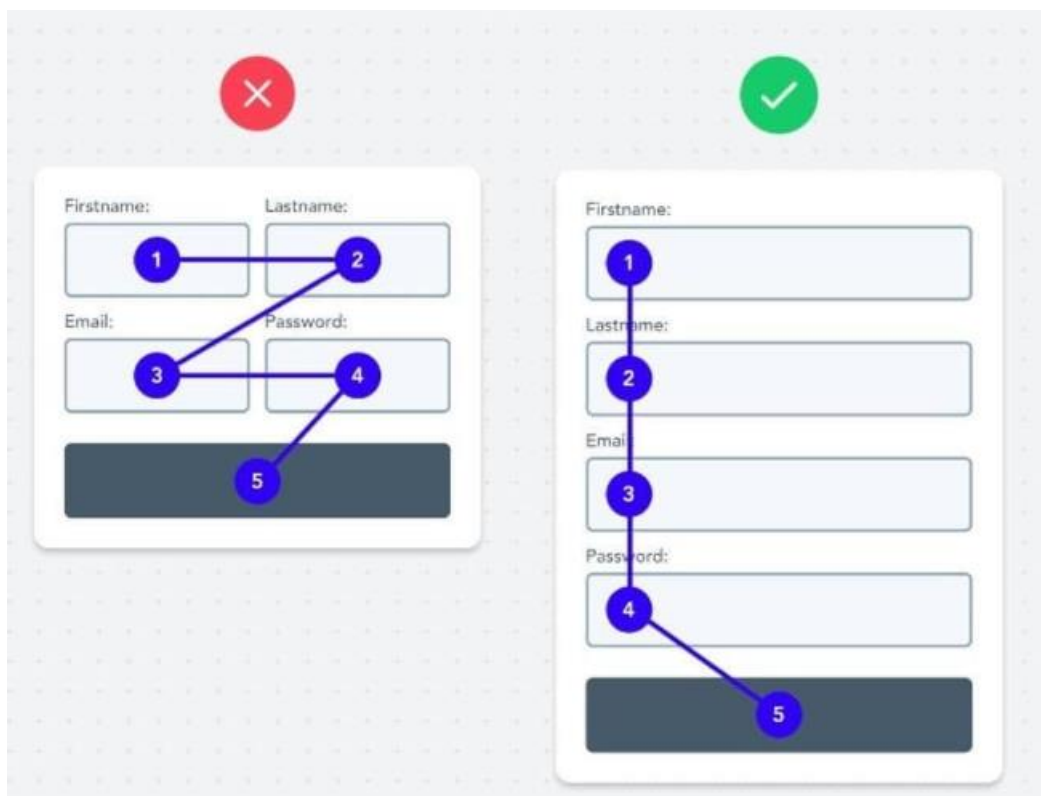


Рисунок-3.5 – Неправильно и правильно составленные формы

Цветовая палитра. Для сайтов рекомендуется использовать не больше пяти цветов. Для сайта я подобрал 4 основных цвета (рисунок 3.6)

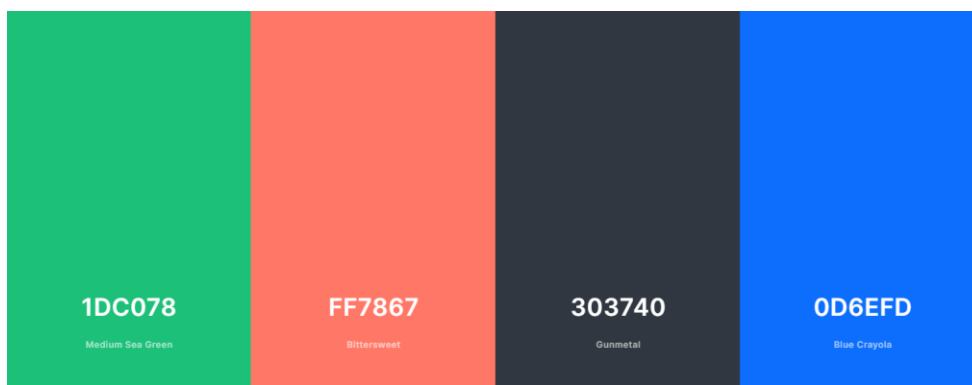


Рисунок-3.6 – цветовая палитра сайта

4 Экспериментальная часть

В данном разделе я расскажу вам о том, как велась разработка веб-сайта и про текущую структуру сайта.

Сайт состоит из нескольких страниц, каждая из них может содержать в себе под страницы.

Страницы сайта:

- 1) Главная
- 2) Курсы
- 3) Дорожная карта (Roadmaps)
- 4) Менторы
- 5) Сообщество

Сообщество — это место вопросов и ответов и особой логики ща собой не несет.

4.1 Главная страница

В этой странице показывается предварительная информация про курсы которая находится в публичном доступе

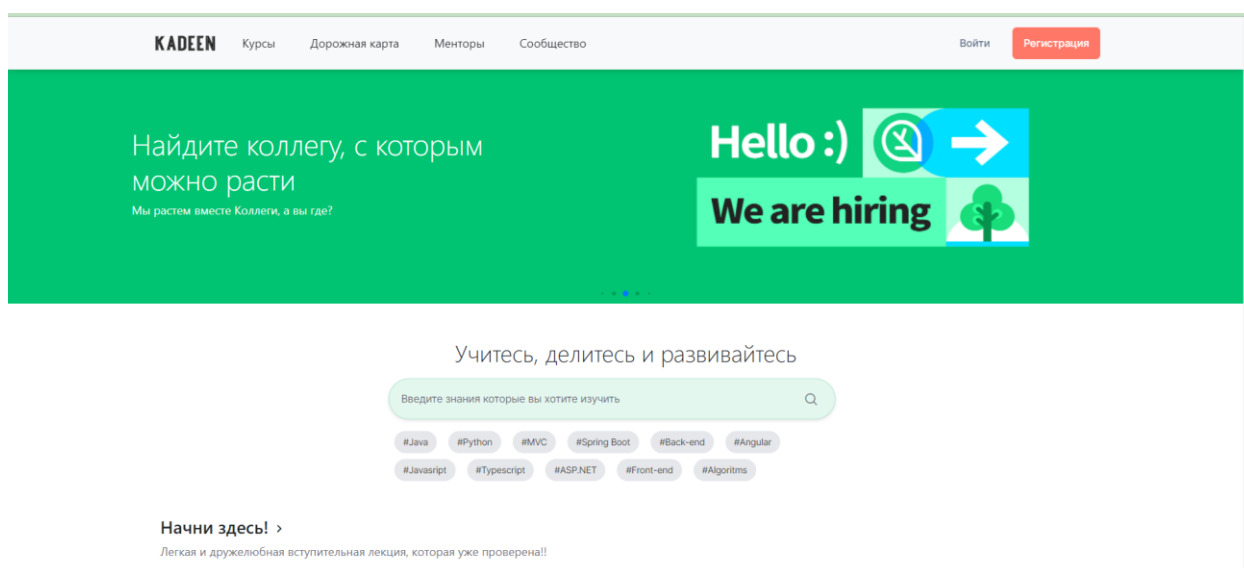


Рисунок-4.1– Главная страница

На главной странице доступно очень много информации для пользователей. Пользователь может тут искать по сайту, видит новые и популярные направления для обучения. Так же есть система тегов. Теги указываются в курсах, roadmaps, в опросах и блогах. В поисковой строке можно указать как теги, так и название курса. Поиск ведется по названиям курсов и дорожных карт.

4.2 Курсы

В странице курсы пользователь просматривает курсы, то есть можно сказать, что это листовая страница курсов (рисунок 4.2).

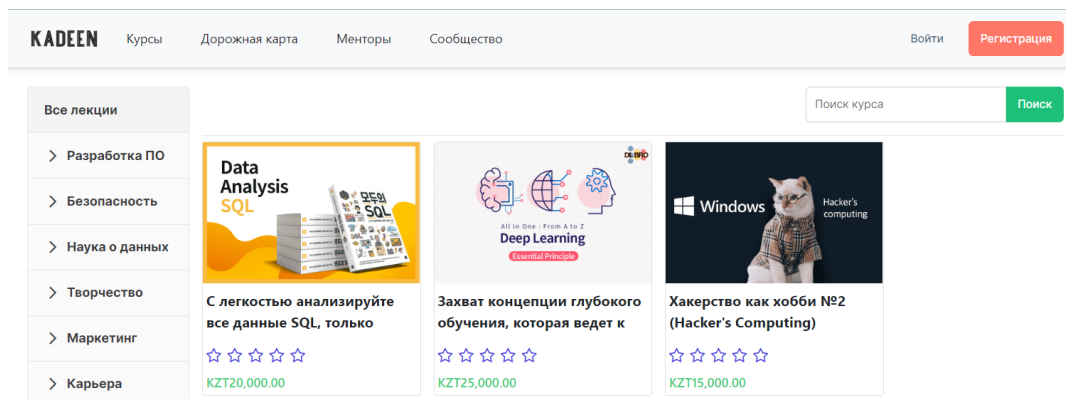


Рисунок-4.2 – листовая страница курсов

С левой стороны сайта есть панель для фильтрации курсов и сферам разработки и с верху стоит поисковая строка. Сейчас это не заметно, так как у нас всего три курса, но на сайте пагинация курсов. Пагинация позволяет нам передавать большие данные кусками что в дальнейшем уменьшит передаваемый трафик и повысит производительность сайта.

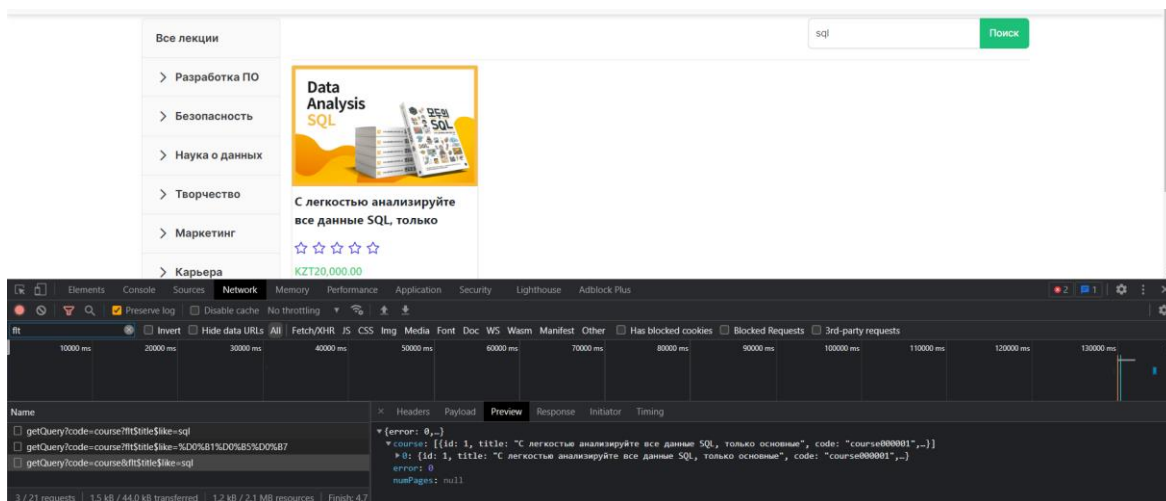


Рисунок-4.3 – Пример поиска по курсам

В детальной странице (рисунок 4.4) можно будет подробно рассмотреть информацию про курс и при желании записаться на него.



Рисунок-4.4 – Детальная страница курса

Помимо просмотра курса так же есть сообщество, где вы сможете задавать вопросы, а также написать отзыв о курсе

А сами курсы, которые можно просматривать онлайн, точнее их содержание во вкладке Уроки (рисунок 4.5)

Введение в лекцию	учебный план	Урок 16	сообщество
▶ Введение в однострочные функции			05:28
▶ Однострочные функции - функции символьного типа, функции числового типа			15:06
▶ Однострочные функции - функции типа даты, функции преобразования			14:48
▶ Однорядные функции - обычные функции			13:01
▶ Введение в групповые функции			09:15
▶ Групповые функции — предложение GROUP BY, предложение HAVING		Предварительный просмотр	11:51
▶ (Анализ) Подготовка сравнительного анализа и интерпретация результатов		Предварительный просмотр	12:36

Рисунок-4.5 – Содержание курса

Каждый урок — это набор коротких видео лекций. Для воспроизведения используется библиотека VideoJS.

Предварительный просмотр доступен всем пользователям системы, в том числе и не зарегистрированным. Что делает курсы более прозрачными и дает пользователю оценить для себя качество обучения.

4.3 Дорожная карта (Roadmaps)

Дорожная карта — это объединения нескольких курсов. Например, чтобы стать разработчиком на стеке MEAN вам потребуется изучить несколько курсов. Курсы в roadmap MEAN могут входить

- 1) Основы HTML/CSS
- 2) Javascript
- 3) Angular
- 4) Сети базовые протоколы передачи данных
- 5) nodejs and angular
- 6) Mongodb
- 7) MEAN stack

Таким образом пользователь получает подробный путь, по которому ему стоит двигаться дальше. На этом мы, наверное, определились с тем, что такое Дорожная карта, это коллекция курсов. Если идти по цене тогда при покупке дорожной карты пользователь получает все курсы, которые внутри карты.

4.4 Менторы

Менторы— одна из основных частей. Эта часть была добавлена с целью того, что пользователи смогли бы делиться с другими участниками со своими знаниями, и наставлять их. Логика работы как у фрилансного центра. Пользователь добавляет к себе описание профиля и то нам чем он может вам наставлять. Студентам стоит обратиться на менторство в том случае если трудно дается материал курса.

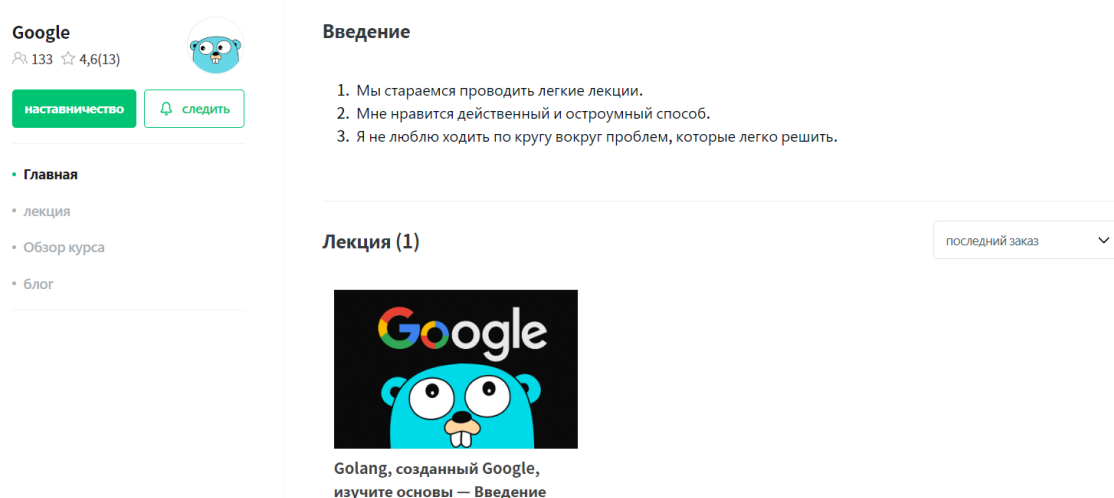


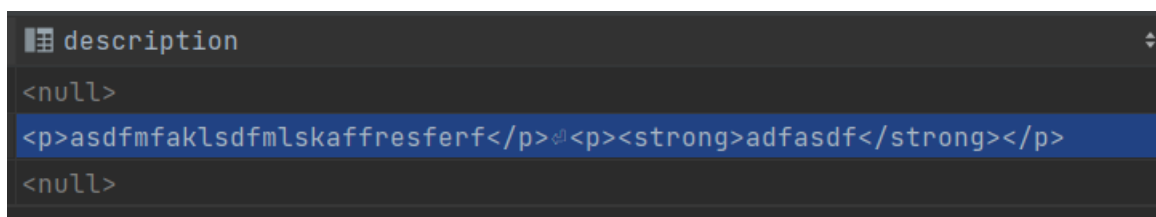
Рисунок-4.6 – Страница ментора

Менторы в основном будут заниматься один на один, а формат урока определяются ментором так же, как и стоимость, время и максимальное количество учеников.

4.5 Создание описаний и динамические компоненты

На сайте присутствуют части, которые при обычном построении могли бы затруднить разработку и сделать ее долгой. Например, описание для курсов. У каждого курса уникальное описание и менторы, которые добавляют их не хотят, что бы оно было как обычное текстовое поле. Что бы дать им кастомизировать я используется AOT подход для рендеринга компонентов. Это генерации коды для Angular в реальном времени. Стоял вопрос как же дать ментору редактировать HTML код. Поискав в интернете, я наткнулся на Tiny Editor [17] который как раз таки решает нашу проблему.

В базе данных описание пользователей и курсов храниться с HTML тегами (рисунок 4.7).

A screenshot of a code editor window titled 'description'. The editor shows HTML code with a blue selection highlighting a paragraph tag. The code is:

```
<null>
<p>asdfmfaklsdfmlskaffresferf</p> <p><strong>adfasdf</strong></p>
<null>
```

Рисунок-4.7 – Пример хранения описания

4.6 Структура проекта

Проект состоит из двух частей это клиентская и серверная часть. У обоих использованных технологий есть рекомендуемые структуры для построения приложения. Это делается в первую очередь для понятности и соблюдения чистоты приложения. Структура клиентской части (рисунок 4.8) показывает файловую директорию и как они разделены. В основной папке находятся компоненты, сервисы, модели, которые используются глобально.

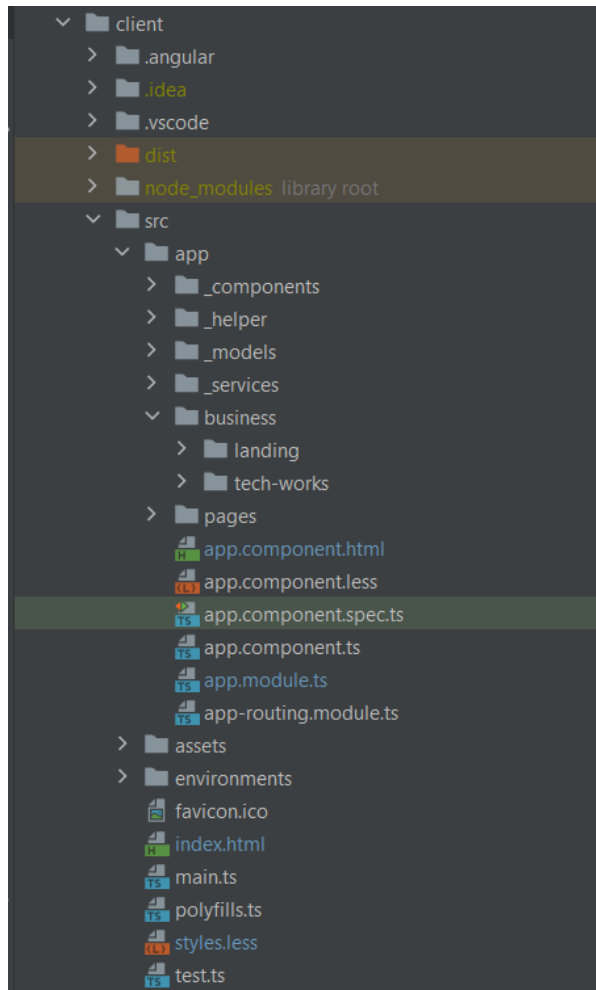


Рисунок-4.8 – Структура клиентской части

Серверная часть (рисунок 4.9) не большая. В главной папке показывается, что является основным для приложения и архитектура очень близка к Flat архитектуре.

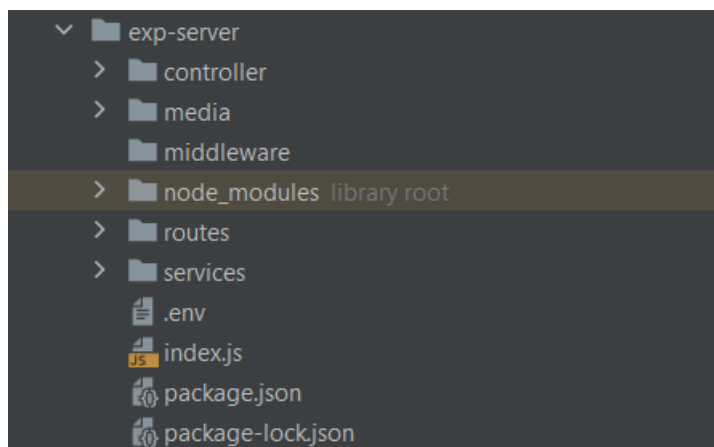


Рисунок-4.9 – Структура серверной части

4.7 Дополнительные пакеты и фреймворки

Для создания компонентов или же layout для сайта вручную ушло бы много времени и не факт, что оно будет мне отлично служить. По этой причине в проекте используется дополнительные фреймворки для построения компонентов и сохранять единый стиль на сайте.

Bootstrap – имеет огромную коллекцию классов и готовых шаблонов. Он очень помогает, когда у разработчика мало знаний в CSS. С его помощью компоненты писать становится гораздо легче. Например что бы окрасить кнопку (рисунок 4.10) и придать ему эффекты пришлось бы самому писать классы и выносить в глобальные стили, но с bootstrap это делается гораздо быстрее.

```
<button pButton label="Больше отзывов"
  icon="pi pi-angle-right" iconPos="right" (click)="openComments()"
  class="p-button-text
p-button-outlined
p-button-secondary
border-green-300
transition-colors transition-duration-300
mr-2 ml-2
hover:bg-green-300"></button>
```

Рисунок-4.10 Пример использования классов из Bootstrap



Рисунок-4.11 – Результат рисунка 4.10

При наведении на кнопку появляется эффект изменения заднего фона.

PrimeNG – тоже одна полезная для Angular приложений библиотека. Имеет в себе много готовых компонентов, которых можно еще и кастомизировать. Так же есть классы которые тоже можно использовать для верстки других компонентов.

```
<aside>
  <p-accordion>
    <p-accordionTab class="main-accordion"
      header="Все лекции">
    </p-accordionTab>
    <ng-container *ngFor="let ac of accordionHeadItems">
      <p-accordionTab header="{{ac.title}}">
        <p-listbox [options]="ac.items || []" optionLabel="title" optionValue="code" (onClick)="filterByTag($event)"></p-listbox>
      </p-accordionTab>
    </ng-container>
  </p-accordion>
</aside>
```

Рисунок-4.12 – пример использования компонентов из PrimeNG

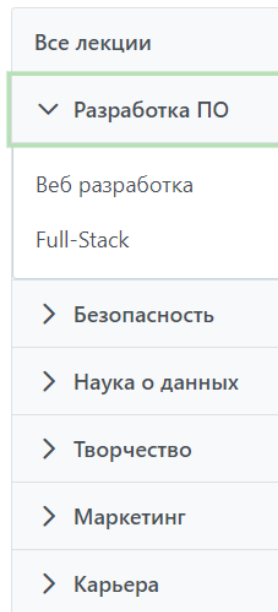


Рисунок-4.13 результат рисунка 4.12

SwiperJS – библиотека для слайдеров. Она очень широко используется на главной странице. С помощью нее я можно сказать сэкономил очень много времени. Сделать качественный слайдер за короткое время для меня невозможное.

```
<swiper [pagination]="{
  dynamicBullets: true
}"
[slidesPerView]="num_visible>5? 5: num_visible"
[autoplay]="{
  delay: 3500,
  disableOnInteraction: false
}">
<ng-template swiperSlide *ngFor="let item of items">
  <div class="pt-3 pb-3 pr-2 pl-2 mb-3 hover:bg-green-50 cursor-pointer" (click)="goDetail(item.id)">
    <div class="w-100 flex flex-wrap justify-content-center">
      <img [src]="item.image_id" height="150px" width="230px">
    </div>
    <div class="font-semibold">
      Title {{item.title}}
    </div>
    <span *ngIf="showAuthor" class="text-500">{{item.author_name}}</span>
    <p-rating *ngIf="showRating" [cancel]="false"></p-rating>
    <span *ngIf="showCost" class="text-green-400">{{item.cost | currency: 'KZT'}} &#8376;</span>
  </div>
</ng-template>
</swiper>
```

Рисунок-4.14 – Пример использования SwiperJS

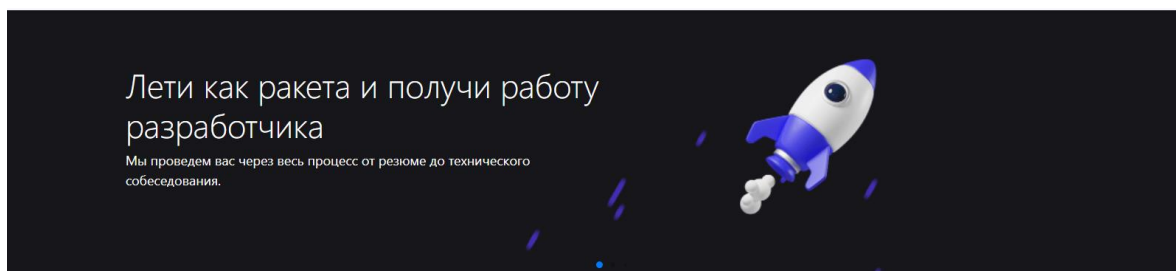


Рисунок-4.16 Результат рисунка 4.14

Данная картинка и является результатом использования SwiperJS. Тут на самом деле несколько картин и они меняются периодически.

ЗАКЛЮЧЕНИЕ

Результатом данной работы является полноценный сайт для компании Kadeen. Вовремя проектирование и разработки было много сложностей, в особенности с определением требований и ожиданий заказчика (компании). Были устроены несколько встреч и проведены брифы для формулировки четкого понимания сайта. В итоге пришли к выводу что сайт будет не только для одной компании, а получит возможность расширяться за счет других пользователей.

Если говорить непосредственно про разработку сайта для меня это первый опыт работы с данным стеком технологий, не считая Angular. Были эксперименты и с другими языками и Backend серверами точнее с Django. Он оказался не совсем тем, что нужно для сайта и перешли на Express js.

Функционал сайта оказался больше, чем ожидалось в изначальном видении. Но и есть некоторые нюансы, которые не выполнимы в рамках дипломной работы, в частности это система платежей, которую нельзя подключить для тестирования.

В итоге мы получаем образовательную платформу, где каждый может повысить свои навыки в программировании и не только. Ожидается что специалисты с других сфер тоже будут выкладывать свои курсы и тем самым на сайте появятся новые разделы.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

- 1 Документация Angular [Электронный ресурс] – Режим доступа: <https://angular.io/>, свободный
- 2 Добавляем PWA в Angular приложение [Электронный ресурс] – Режим доступа: <https://habr.com/ru/post/554316/>, свободный
- 3 Концепты построение приложений на Angular [Электронный ресурс] – Режим доступа: <https://angular.io/guide/architecture>, свободный
- 4 Официальный сайт [Электронный ресурс] – Режим доступа: <https://expressjs.com/ru/>, свободный
- 5 Все что нужно знать про MySQL [Электронный ресурс] – Режим доступа: <https://www.talend.com/resources/what-is-mysql/>, свободный
- 6 Официальная страница WebStorm [Электронный ресурс] – Режим доступа: <https://www.jetbrains.com/ru-ru/webstorm/>, свободный
- 7 Официальная страница DataGrip [Электронный ресурс] – Режим доступа: <https://www.jetbrains.com/ru-ru/datagrip/>, свободный
- 8 Node.js против nginx [Электронный ресурс] – Режим доступа: <https://ru.education-wiki.com/8500472-node.js-vs-nginx>, свободный
- 9 NGINX [Электронный ресурс] – Режим доступа: <https://nginx.org/ru/>,
- 10 UML Diagram [Электронный ресурс] – Режим доступа: <https://www.smartdraw.com/uml-diagram/>, свободный
- 11 ER Diagram [Электронный ресурс] – Режим доступа: <https://www.smartdraw.com/entity-relationship-diagram/>, свободный
- 12 Яков Ф, Моисеев А. Angular и Typescript сайт строение для профессионалов, 2018
- 13 Херрон Д. NodeJS web Development, 2020
- 14 Сегодня А., Что такое UX/UI? [Электронный ресурс] – Режим доступа: <https://habr.com/ru/post/321312/>, свободный
- 15 Cross origin resource sharing [Электронный ресурс] – Режим доступа: <https://developer.mozilla.org/ru/docs/Web/HTTP/CORS>, свободный
- 16 Официальная страница Tinymce [Электронный ресурс] – Режим доступа: <https://www.tiny.cloud/>, свободный
- 17 Godbolt M. Frontend Architecture for Design Systems: A Modern Blueprint for Scalable and Sustainable Websites // М: «O'Reilly Media», 2016
- 18 Черный Б Профессиональный TypeScript. Разработка масштабируемых JavaScript приложений // М: «O'Reilly Media», 2020
- 19 Божинов В. RESTful Web API Design with Node.js // М: «Packt Publishing», 2016
- 20 М. Кантелон, М. Хартер, Т. Головайчук, Н. Райлих, Node.js в действии, 2014

Приложения А

(обязательное)

Техническое задание

А.1.5 Техническое задание на разработку веб-сайта для ТОО «Kadeen»

Настоящее техническое задание распространяется на разработку веб-сайта для ТОО «Kadeen» предназначенная для поднятия потока клиентов для предприятия, а также автоматизировать бизнес-процессы внутри компании. У молодой компании пока не имеется аналогичный сайт, разработка данного сайта даст возможность представить услуги компании на первых страницах поисковиков.

А.1.5.1 Основание для разработки

Система разрабатывается на основании устного распоряжения Заместителя Председателя Правления АО «Институт цифровой техники и технологий» по разработке программных продуктов.

А.1.5.2 Назначение

Разработка сайта предназначена для автоматизации учебных и бизнес-процессов внутри компании.

Формат ведения курсов может быть как оффлайн, так и онлайн, в зависимости от создателя курса. Все учебные материалы можно просматривать на сайте преподавателем либо же учеником. Сайт расширяет возможности обычных сайтов по обучению тем, что можно будет нанять ментором для индивидуального обучения.

Компания получает доступ к данным, которые она может в дальнейшем анализировать и составлять планы по улучшению сайта.

А.1.5.3 Требования к функциональным характеристикам

В системе будет 4 основных ролей пользователя, для каждого из них будет

Продолжения приложения А

реализовать отдельный функционал.

Основные роли:

- студенты;
- преподаватели;
- гости;

Функциональные возможности студентов:

- регистрация/авторизация;
- читать ленту;
- запись на курс;
- оплата обучений (функционал не входит в 70% работы, Google pay возможна не полная реализация);
- комментировать курс которые он прошел либо которые проходит на момент;
- просматривать материалы по урокам;

Функциональные возможности преподавателя:

- вести курс;
- проверять посещаемость учеников;
- выкладывать материал по курсу;
- выкладывать задания (Google forms интеграция, не включено в 70%).

Для гостей она будет выглядеть как Landing page что даст потенциальному клиенту подробную информацию о компании, про предоставляемые услуги, а также есть доступ просматривать актуальные курсы. А запись на курс можно будет провести уже зарегистрировавшись.

А.1.5.4 Требования к надежности

Все роли в системе должны работать только в своей ответственной среде. На клиентской части приложения должно действовать Guard что будет предотвращать не желанный доступ к области ответственности другой роли. На серверной части так же действует проверка доступа ролей на операции CRUD. Все важные данные пользователей должно быть надежно защищено при передачи по сети используя протокол HTTPS и методы POST. Защищать систему от неправильно заполненных данных пользователем.

Приложение Б (обязательное)

Текст программы

Коды сайта Quiry API

```
const express = require('express');
const db = require('../services/db')

const router = express.Router();
module.exports = router;

router.route('/').get(getQuery);

function getQuery(req, res) {
  parseFilters(req)
  const code = req.query.code
  if (!code) {
    res.send({
      error: 1,
      errText: 'Code is required parameter!'
    })
    return;
  }
  const limit = +req.query.limit || 20
  const page = +req.query.page

  if (limit && page) {
    getQueryPagination(code, limit, page, req, res);
  } else {
    getQueryAll(code, req, res);
  }
}

function getQueryPagination(code, limit, page, req, res) {
  const skip = (page - 1) * limit;
  const _limit = skip + ',' + limit; // Here we compute the LIMIT parameter
  for MySQL query
  db.query(`SELECT count(*) as numRows FROM ${code} where 1=1 and
  ${parseFilters(req)}`, (err, rows, fields) => {
    if (err) {
      console.log("Get Query Pagination error: ", err);
      res.send({error: 1});
    } else {
      const numRows = rows[0].numRows;
      const numPages = Math.ceil(numRows / limit);
      db.query(`SELECT * FROM ${code} where 1=1 ${parseFilters(req)} LIMIT
      ${_limit}`, function (err, rows, fields) {
        if (err) {
          console.log("Get Query Pagination error: ", err);
          res.send({error: 1});
        } else {
          res.send({error: 0, rows, numPages});
        }
      });
    }
  });
}
```

Продолжения приложения Б

```
function getQueryAll(code, req, res) {
  db.query(`SELECT * FROM ${code} where 1=1 ${parseFilters(req)}`, function
(err, rows, fields) {
  if (err) {
    console.log("Get Query All error: ", err);
    res.send({error: 1});
  } else {
    res.send({error: 0, [code]: rows, numPages: null});
  }
});
}

//{ code: 'users_user', 'flt$email$like$': 'il' }
function parseFilters(req) {
  let condition = ''
  Object.keys(req.query)?.filter(x => x.startsWith('flt'))
  ?.forEach(x => {
    const [_ , column, filter, ...other] = x.split('$');
    switch (filter) {
      case 'like':
        condition += `and ${column} like '%${req.query[x]}%'`
        break;
      case 'eq':
        condition += `and ${column} = '${req.query[x]}'`
        break;
      case 'in':
        // скобки расставить в запросе
        condition += `and ${column} in ${req.query[x]}`
        break;
    }
  })

  return condition !== '' ? ` ${condition}` : 'and true';
}
```

Код авторизации пользователя

```
const express = require('express');
const db = require('../services/db')
const userCtrl = require('../controller/auth.controller')
const bcrypt = require('bcrypt')

const router = express.Router();
module.exports = router;

router.post('/login', (req, res) => {
  const email = req.body.email;
  const password = req.body.password;

  db.query('select password from users where email = ?', [email], ((err,
result) => {
    if (result && result.length !== 0 && !err) {
      bcrypt.compare(password, result[0].password, (err, bcrypt_res) => {
        if (bcrypt_res && !err) {
          const token = userCtrl({
            email
```

Продолжения приложения Б

```
    })

    bcrypt_res.send({

        error: 0,
        data: {
            token
        }
    })
} else {
    res.send({
        error: 1,
        errText: 'incorrect password'
    })
}
})
} else {
    res.send({
        error: 1,
        errText: 'User not found'
    })
}
}

})))

});
```

Получение детали по записям

```
const express = require('express');
const db = require('../services/db')

const router = express.Router();
module.exports = router;
router.get('/', getDetail)

function getDetail(req, res) {
    const code = req.query.code;
    const id = req.query.id;
    if (!code && !id) {
        res.send({
            error: 1,
            errText: 'Code is required parameter'
        })
        return;
    }
    db.query(`SELECT * FROM ${code} where id = ${id}`, function (err, rows,
fields) {
        if (err) {
            console.log("Get Query All error: ", err);
            res.send({error: 1});
        } else {
            res.send({error: 0, [code]: rows, numPages: null});
        }
    });
}

}
```

Продолжения приложения Б

Сервер

```
const express = require('express')
const path = require('path')
const http = require('http')
const compression = require('compression')
const bodyparser = require('body-parser')
const cors = require('cors')

const router = require('./routes/index.route')
require('dotenv').config()
const app = express()
const appname = 'client'
// Compress static assets to enhance performance.
// Decrease the download size of your app through gzip compression:
app.use(compression())
app.use(cors())
app.use(bodyparser.json())
app.use(express.static(path.join(__dirname, '../client/dist', appname)))
app.use('/media', express.static(path.join(__dirname, '/media')))

// Point static path to dist

app.get('/all', (req, res) => {

  res.send({
    add: 'asdf'
  })
})
app.use('/api', router)
// Catch all routes and return the index file
// app.get('*', (req, res) => {
//   res.sendFile(path.join(__dirname, '../client/dist', appname,
// 'index.html'))
// })

// Get port from environment and store in Express.
const port = process.env.PORT || '3000'
app.set('port', port)
// Create HTTP server.
const server = http.createServer(app)
// Listen on provided port, on all network interfaces.
server.listen(port, () => {
  console.log(`Angular app \`${appname}\` running on port ${port}`)
  console.log(__dirname)
})
```

Пакеты установленные на сервер

```
{
  "name": "exp-server",
  "version": "1.0.0",
  "description": "Kadeen express server",
  "main": "index.js",
  "scripts": {
    "test": "test"
  },
}
```


Продолжения приложения Б

```
"author": "Idrwla",

"license": "ISC",
"dependencies": {
  "bcrypt": "^5.0.1",
  "body-parser": "^1.20.0",
  "compression": "^1.7.4",
  "cors": "^2.8.5",
  "dotenv": "^16.0.0",
  "express": "^4.18.1",
  "jsonwebtoken": "^8.5.1",
  "mysql2": "^2.3.3",

  "sequelize": "^6.19.0"
}
}
```

Пакеты установленные на клиент

```
{
  "name": "client",
  "version": "0.0.0",
  "engines": {
    "node": "14.17.5"
  },
  "scripts": {
    "ng": "ng",
    "start": "node ../exp-server/index.js",
    "postinstall": "ng build --configuration production --aot",
    "build": "ng build",
    "watch": "ng build --watch --configuration development",
    "test": "ng test",
    "fast": "ng build --source-map=false --watch=true"
  },
  "private": true,
  "dependencies": {
    "@angular-devkit/build-angular": "~13.2.6",
    "@angular/animations": "~13.2.0",
    "@angular/cdk": "^13.3.2",
    "@angular/cli": "~13.2.6",
    "@angular/common": "~13.2.0",
    "@angular/compiler": "~13.2.0",
    "@angular/compiler-cli": "~13.2.0",
    "@angular/core": "~13.2.0",
    "@angular/forms": "~13.2.0",
    "@angular/platform-browser": "~13.2.0",
    "@angular/platform-browser-dynamic": "~13.2.0",
    "@angular/router": "~13.2.0",
    "@fortawesome/angular-fontawesome": "^0.10.1",
    "@fortawesome/fontawesome-svg-core": "~1.2.36",
    "@fortawesome/free-solid-svg-icons": "^5.15.4",
    "angularx-social-login": "^4.1.0",
    "bootstrap": "^5.1.3",
    "compression": "^1.7.4",
    "express": "^4.17.3",
    "path": "^0.12.7",
    "primeflex": "^3.1.3",
    "primeicons": "^5.0.0",
    "primeng": "^13.2.1",
    "remixicon": "^2.5.0",
```

Продолжения приложения Б

```
"rxjs": "~7.5.0",
"swiper": "^8.1.3",
"tslib": "^2.3.0",
"typescript": "~4.5.2",
"zone.js": "~0.11.4"
},
"devDependencies": {
"@angular-devkit/build-angular": "~13.2.6",
"@angular/cli": "~13.2.6",
"@angular/compiler-cli": "~13.2.0",
"@types/jasmine": "~3.10.0",
"@types/node": "^12.11.1",
"jasmine-core": "~4.0.0",
"karma": "~6.3.0",
"karma-chrome-launcher": "~3.1.0",
"karma-coverage": "~2.1.0",

"karma-jasmine": "~4.0.0",
"karma-jasmine-html-reporter": "~1.7.0",
"typescript": "~4.5.2"
}
}
```

Сервис для получения данных с сервера

```
import {Injectable} from '@angular/core';
import {HttpClient, HttpHeaders} from "@angular/common/http";
import {environment} from "../../environments/environment";
import {Observable} from "rxjs";

@Injectable({
  providedIn: 'root'
})
export class ApiService {

  private contentHeader = new HttpHeaders({
    'Content-Type': 'application/json',
    'Access-Control-Allow-Origin': '*',
  });

  private contentHeaderText = new HttpHeaders({
    'Content-Type': 'html/text',
    'Access-Control-Allow-Origin': '*',
  });

  private contentHeaderFile = new HttpHeaders({
    'Content-Type': 'multipart/form-data'
  });

  constructor(private http: HttpClient) {
  }

  getQuerySelect(code: string): Observable<any> {
    return this.http.get(`${environment.apiUrl}${code}`, {headers:
this.contentHeader})
  }

  getDetail(code: string, id: string | number): Observable<any> {
    return this.http.get(`/api/getDetail`, {
      headers: this.contentHeader, params: {
```

Продолжения приложения Б

```
        code: code, id: id
    }
    })
}

getQuery(code: string, params?: {}): Observable<any> {
    return this.http.get(`/api/getQuery`, {

        params: {code: code, ...params}
    })
}
}
```

Сервис по работе с аутентификацией

```
import {EventEmitter, Injectable} from '@angular/core';
import {HttpClient, HttpHeaders} from "@angular/common/http";
import {environment} from "../../environments/environment";
import {Router} from "@angular/router";

import {MessageService} from "primeng/api";
import {DataResponse} from "../_models/common";
import {User} from "../_models/user";

@Injectable({
    providedIn: 'root'
})
export class AuthService {

    private user: User | null = null;
    loggedIn = false;
    loggedInEmitter = new EventEmitter();
    private headers = new HttpHeaders({
        'Content-Type': 'application/json',
        'Access-Control-Allow-Origin': '*'
    });
    private _httpOptions = {headers: this.headers, withCredentials: true}

    constructor(private http: HttpClient,
                 private router: Router,
                 private toast: MessageService) {
        if (!this.user) {
            this.getMe();
        }
    }

    singIn(userData: any): void {
        this.http.post<DataResponse>(`${environment.apiUrl}api/singIn/`, userData,
        this._httpOptions)
            .subscribe(r => {
                console.log(r)
                if (r) {
                    this.loggedInEmitter.next(true)
                    this.user = r.data as User
                    localStorage.setItem('kadeen-jwt', r.jwt);
                    this.loggedIn = true;
                }
            });
    }
}
```

Продолжения приложения Б

```
        this.router.navigate(['/']).then()
    }

    }, error => {
        console.log('Error', error)
        this.toast.add({severity: 'error', summary: 'Ошибка авторизации'})
    })
}

singUp(userData: any): void {
    this.http.post<DataResponse>(`${environment.apiUrl}api/singUp/`, userData,
this._httpOptions)
        .subscribe(r => {
            console.error(r)
            if (r) {
                this.loggedInEmitter.next(true)
                this.user = r.data as User;
                localStorage.setItem('kadeen-jwt', r.jwt);
                this.loggedIn = true;
                this.router.navigate(['/']).then()
            }
        }, error => {
            this.toast.add({severity: 'error', summary: 'Ошибка регистрации'})
        });
}

getMe(): void {
    const body = {
        data: {
            jwt: localStorage.getItem('kadeen-jwt')
        }
    }
    this.http.post<DataResponse>(`/api/user/`, body, this._httpOptions)
        .subscribe(r => {
            if (r['status']) {
                this.user = r.data as User
            } else {
                this.user = null
            }
            this.loggedInEmitter.emit(r.status)
        })
}

getUserEmail(): string {
    return this.user?.email || '';
}
}
```

Пример использования Swiper js

```
<swiper [pagination]="{
    dynamicBullets: true
}"
```

Продолжения приложения Б

```
[autoplay]="{
delay: 3500,
  disableOnInteraction: false
}">

<ng-template swiperSlide *ngFor="let item of images">

  <div class="scene">
    <div class="scene_content" [ngStyle]="{backgroundColor: item.background}">
      <div class="container">
        <div class="cover_right">
          <img class="d-block" [src]="item.image_id" [alt]="item.title">
        </div>
        <div class="cover_left" [ngStyle]="{color: item.textColor}">
          <h1 class="fw-light">{{item.title}}</h1>
          <span>{{item.subtitle}}</span>
        </div>
      </div>
    </div>
  </div>

</ng-template>

</swiper>
```

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Казахский национальный исследовательский технический университет имени
К.И.Сатпаева

Институт автоматизации и информационных технологий

Махсумов Ильяр Мухаметжанулы
5B070400 – Вычислительная техника и программное обеспечение

ОТЗЫВ НАУЧНОГО РУКОВОДИТЕЛЯ

к дипломному проекту

Тема: «Разработка веб-сайта для ТОО Kadeen»

Студенту Махсумову И. М. предстояло проделать путь разработки веб сайта для компании с нуля. Были проведены несколько встреч с представителем компании для прояснения требований и формулировки четкого описание сайта для дальнейшей разработки веб сайта и написании дипломного проекта.

В наше время интернет-площадка является одной из самых прибыльных и быстро развивающейся. Большинство компаний имеют или планируют сделать свой сайт. В пределах дипломного проекта студентом был создан прототип веб сайта для компании, которая до этого не имела веб сайта.

Обучающимся был использован современный стек технологий для достижение поставленных целей и студент уверенно использовал выбранные технологии.

Если говорить о сайте, то она имеет хорошую идею и в будущем сможет стать конкурентоспособным на уровне с теми сайтами, с которыми было сравнение.

Пояснительная записка выполнена на высоком уровне, соответствующим требованиям, предъявляемым к данным видам работ. Считаю, что Махсумов Ильяр достоин присвоение академической степени бакалавра техники и технологии по специальности 5B070400 - вычислительная техника и программное обеспечение.

Научный руководитель: Сениор-лектор, Доктор Ph.D.

 _____ Черикбаева Л. Ш.

" 20 " мар 2022 г

РЕЦЕНЗИЯ

На дипломный проект студента 4 курса Казахского национального исследовательского технического университета им.К.И.Сатпаева специальности 5В070400 «Вычислительная техника и программное обеспечение» Махсумов Ильяр Мухаметжанулы
На тему: «Разработка веб сайта для ТОО Kadeen»

Структура дипломной работы включает в себя: введение, технологический раздел, проектная часть, экспериментальная часть, список используемых литературы и приложений и заключение. Так же были использованы таблицы и фотографии.

Во введении описывается цели разработки веб-сайта и возможности для развития. Способы избавления от лишних трудозатрат и взятия на себе роль обучения новых специалистов.

В первой главе определяется цели разработки веб-сайта, его актуальность и анализ аналогичных сайтов. Также все термины и сокращения, которые использовались для разработки веб-сайта, а также специфические термины связанные с используемыми технологиями

Во второй главе затронули технологический раздел, инструменты, которые были использованы в разработке веб сайта. Плюсы использования готовых фреймворков и где они применялись в проекте.

В третьей главе рассказано про архитектуру проекта, которая является неотъемлемой частью всех приложений. Также приведены UML диаграммы, в частности, Use case, ER и диаграмма активности. Была не большая часть про дизайн и стилистику сайта что для современных сайтов является одной из ключевых принципов.

В четвертой главе рассказывается про саму разработку сайта и что в итоге получили. Каждый раздел был описан и приведены примеры использования некоторых дополнительных технологий, которые использовали. Показаны страницы с сайта как все сейчас выглядит.

Замечание по проекту

В целом работа представлена завершенной и может быть оценена на «отлично», а при успешной защите Махсумов И. достоин присвоение академической степени бакалавра техники и технологии по специальности 5В070400 - вычислительная техника и программное обеспечение.

Рецензент:

Ст. преподаватель, доктор Ph.D
Казахского Национального
Университета имени Аль-Фараби




Даркенбаев Д.К.

Протокол

о проверке на наличие неавторизованных заимствований (плагиата)

Автор: Махсумов Ильяр Мухаметжанулы

Соавтор (если имеется):

Тип работы: Дипломная работа

Название работы: Разработка веб-сайта для ТОО «Kadeen»

Научный руководитель: Жибек Алибиева

Коэффициент Подобия 1: 0.8

Коэффициент Подобия 2: 0

Микропробелы: 16

Знаки из здругих алфавитов: 0

Интервалы: 0

Белые Знаки: 0

После проверки Отчета Подобия было сделано следующее заключение:

- Заимствования, выявленные в работе, является законным и не является плагиатом. Уровень подобия не превышает допустимого предела. Таким образом работа независима и принимается.
- Заимствование не является плагиатом, но превышено пороговое значение уровня подобия. Таким образом работа возвращается на доработку.
- Выявлены заимствования и плагиат или преднамеренные текстовые искажения (манипуляции), как предполагаемые попытки укрытия плагиата, которые делают работу противоречащей требованиям приложения 5 приказа 595 МОН РК, закону об авторских и смежных правах РК, а также кодексу этики и процедурам. Таким образом работа не принимается.
- Обоснование:

Дата

19.05.2022

Мв-

Заведующий кафедрой