

ЧАСТНА ПРОФЕСИОНАЛНА ГИМНАЗИЯ ПО ДИГИТАЛНИ НАУКИ  
“СОФТУНУ СВЕТЛИНА“

ДИПЛОМНА РАБОТА

ЗА ПРИДОБИВАНЕ НА КВАЛИФИКАЦИЯ

НА ТЕМА:

React Application (SoftUni Chat)

ИЗГОТВИЛ

БОЯН ЯНАКИЕВ ЯНАКИЕВ

28.02.2023

НАУЧЕН РЪКОВОДИТЕЛ

НИКОЛАЙ САПУНОВ

## Съдържание

<b>1. Увод .....</b>	<b>3</b>
<b>2. Изложение.....</b>	<b>6</b>
<b>2.1. Цел на дипломния проект .....</b>	<b>6</b>
<b>2.2. Използвани технологии .....</b>	<b>7</b>
2.2.1. Уеб приложение .....	7
2.2.2. React .....	8
2.2.3. React Components .....	8
2.2.4. Virtual Dom .....	9
2.2.5. JSX .....	10
2.2.6. Еднопосочен поток от данни .....	11
2.2.7. Reconciliation .....	11
2.2.8. React Router.....	12
2.2.9. HTML.....	13
2.2.10. CSS .....	14
2.2.11. JavaScript .....	15
2.2.12. Мениджър на пакетите .....	17
2.2.13. Използвани пакети в приложението .....	18
2.2.14. Firebase.....	21
2.2.15. ChatEngine .....	24
<b>2.3. Задачи, които трябва да се решат за постигането на целта.....</b>	<b>25</b>
<b>2.4. Употреба на приложението .....</b>	<b>25</b>
<b>3. Заключение .....</b>	<b>27</b>
<b>4. Източници .....</b>	<b>28</b>

## 1. Увод

В днешния бързо развиващ се свят комуникацията играе жизненоважна роля за свързването на хората и насърчаването на сътрудничеството. С нарастването на дистанционната работа и онлайн обучението необходимостта от ефективни и удобни за потребителя

комуникационни платформи стана по-критична от всякога. SoftuniChat се очертава като проект, насочен към създаване на безпроблемно комуникационно изживяване за потребителите в затворена общност, като училището. Използвайки модерни технологии като React, JavaScript, Firebase и ChatEngine API, SoftuniChat си поставя за цел да трансформира начина, по който учениците и учителите взаимодействат и комуникират един с друг.

Ефективната комуникация е в основата на производителността и успеха във всяка организация или общност. SoftuniChat има за цел да преодолее затрудняването на комуникацията между хората в дадената организация, позволявайки им да се свързват, да си сътрудничат и да споделят идеи без усилие. Независимо дали става въпрос за студенти, които обсъждат учебни материали, съотборници, работещи по проект, или хора, участващи в смислени разговори, SoftuniChat предоставя мощна платформа за комуникация в реално време. Приложението е платформа за чат в реално време, който може да бъде използван във всяка сфера било то в като място за комуникация между учители и ученици, между лекари и пациенти или между най-обикновени приятели. Чрез комбиниране на надеждни технологии с интуитивен потребителски интерфейс, SoftuniChat цели да революционизира начина, по който хората общуват и да подобри цялостното им преживяване с лесно, симплистично и удобно за нови потребители преживяване.

В случая на този проект платформата бива използвана като училищен чат. Направен е за лесно разбиране от възрастни и млади.

SoftuniChat е изграден върху React Framework, известна със своята мащабируемост, гъвкавост и базирана на компоненти архитектура. React позволява създаването на динамични и интерактивни потребителски интерфейси, което го прави идеалният избор за модерна комуникационна платформа. С React SoftuniChat осигурява гладка навигация, актуализации в реално време и интуитивно потребителско изживяване. За да осигури сигурна и надеждна система за регистриране, SoftuniChat интегрира Firebase Authentication. С Firebase потребителите могат да се удостоверяват, като използват своите акаунти в Google или Facebook, осигурявайки безпроблемно изживяване при включване. Firebase също предлага стабилни функции за сигурност, защита на потребителските данни и осигуряване на надеждна среда за комуникация. SoftuniChat използва API на ChatEngine, за да улесни възможностите за съобщения в реално време. ChatEngine опростява процеса на изграждане на функционалност за чат, позволявайки на потребителите да изпращат и получават съобщения незабавно. Чрез използването на мощните функции на ChatEngine, SoftuniChat създава динамично и ангажиращо комуникационно изживяване, насърчавайки ефективното сътрудничество и бързата комуникация в реално време.

## 2. Изложение

### 2.1. Цел на дипломния проект

Основната цел на SoftuniChat е да създаде безпроблемна и удобна за потребителя комуникационна платформа, която свързва хората и улеснява ефективната работа между тях. Проектът има за цел да отговори на нарастващата нужда от ефективни инструменти за комуникация в различни контексти, като онлайн обучение, дистанционна работа и взаимодействия в общността. Чрез използване на напреднали технологии като React, JavaScript, Firebase Authentication и ChatEngine API, SoftuniChat се стреми да постигне следните цели:

1. Насърчи комуникацията между дадените потребители
2. Да подобри преживяването на потребителите
3. Да промотира работа в екип по проекти и т.н.
4. Да подсили информацията на потребителите
5. Да насърчи извършването на комуникацията върху приложения които седят вътре в дадената организация

В обобщение, целта на SoftuniChat е да революционизира начина, по който хората комуникират и да го задържи в кръга на дадената организация(училище, офис и т.н.), като предостави безпроблемна, сигурна и удобна за потребителя платформа, която насърчава сътрудничеството, свързаността и продуктивността.

## 2.2. Използвани технологии

### 2.2.1. Уеб приложение

Уеб приложение, известно още като уеб апликация, е софтуерно приложение, което работи на уеб браузъри. Той е предназначен за достъп през интернет с помощта на уеб браузър като клиент. За разлика от традиционните настолни приложения, които се инсталират на устройството на потребителя, уеб приложенията се намират на отдалечени сървъри и се осъществява достъп чрез мрежова връзка.

Уеб приложенията могат да предложат широка гама от функционалности и услуги, като онлайн пазаруване, платформи за социални медии, имейл клиенти, инструменти за управление на проекти, онлайн банкиране и др. Те са създадени с помощта на уеб технологии като HTML (Hypertext Markup Language), CSS (Cascading Style Sheets) и JavaScript. Тези технологии позволяват на разработчиците да създават динамични и интерактивни потребителски интерфейси, да обработват и съхраняват данни и да комуникират със сървъри за извличане или съхраняване на информация.

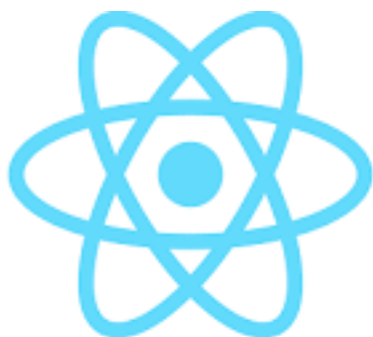
Основното предимство на уеб приложенията е тяхната платформена независимост. Потребителите имат достъп до тях от различни устройства и операционни системи, стига да имат съвместим уеб браузър и интернет връзка. Освен това, уеб приложенията могат лесно да се актуализират и

поддържат, тъй като актуализациите се прилагат от страната на сървъра, не изисквайки никакви действия от крайните потребители.

Като цяло уеб приложенията стават все по-популярни поради тяхната достъпност, съвместимост между платформи и способността да предоставят мощни и интерактивни услуги по интернет.

### 2.2.2. React

React е популярна JavaScript библиотека за изграждане на потребителски интерфейси. React е разработен от Facebook и сега се поддържа от Facebook и жизнена общност от разработчици. React позволява на разработчиците да създават компоненти на потребителския интерфейс за многократна употреба и ефективно да ги актуализират и визуализират в отговор на промените в данните.



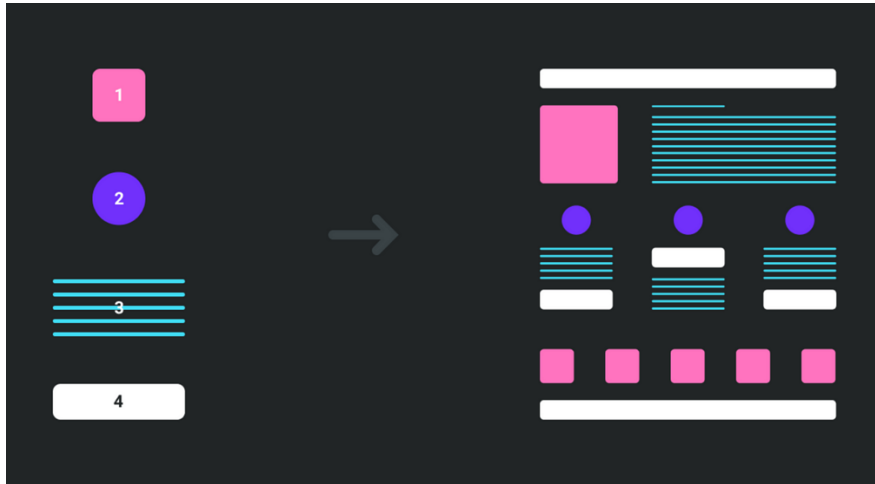
Фигура 1 - React лого

### 2.2.3. React Components

Компоненти: React следва архитектура, базирана на компоненти. Компонентите са самостоятелни модули за многократна употреба, които капсулират собственото си състояние, логика и изобразяване. Компонентите



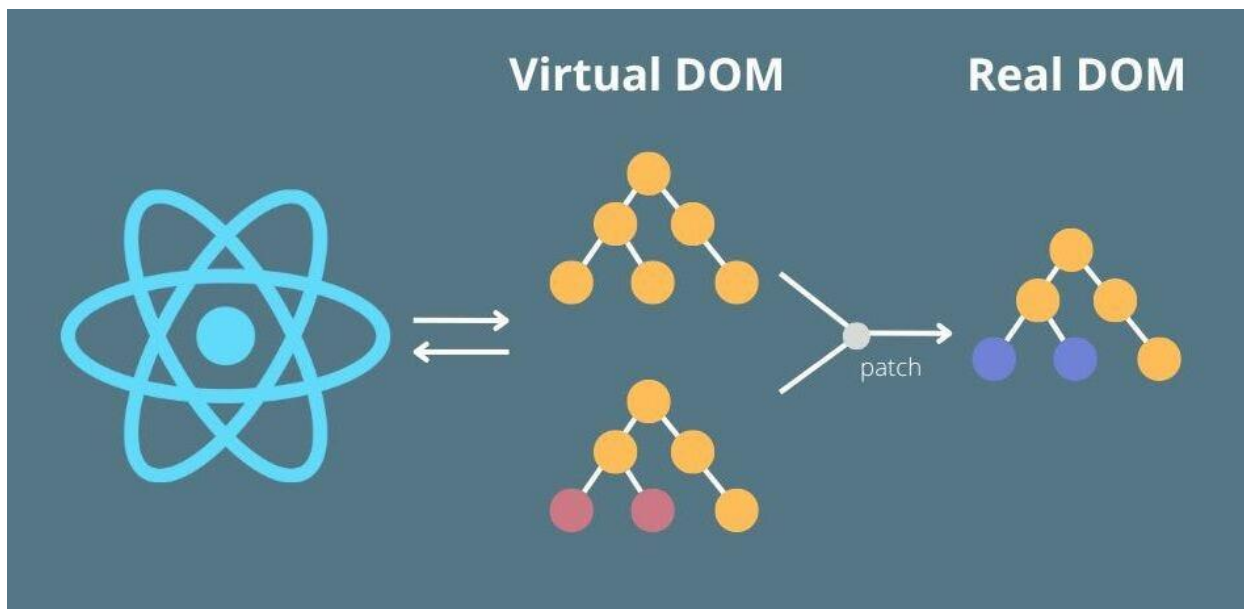
могат да бъдат композирани заедно за изграждане на сложни потребителски интерфейси.



Фигура 2 - React компоненти

#### 2.2.4. Virtual Dom

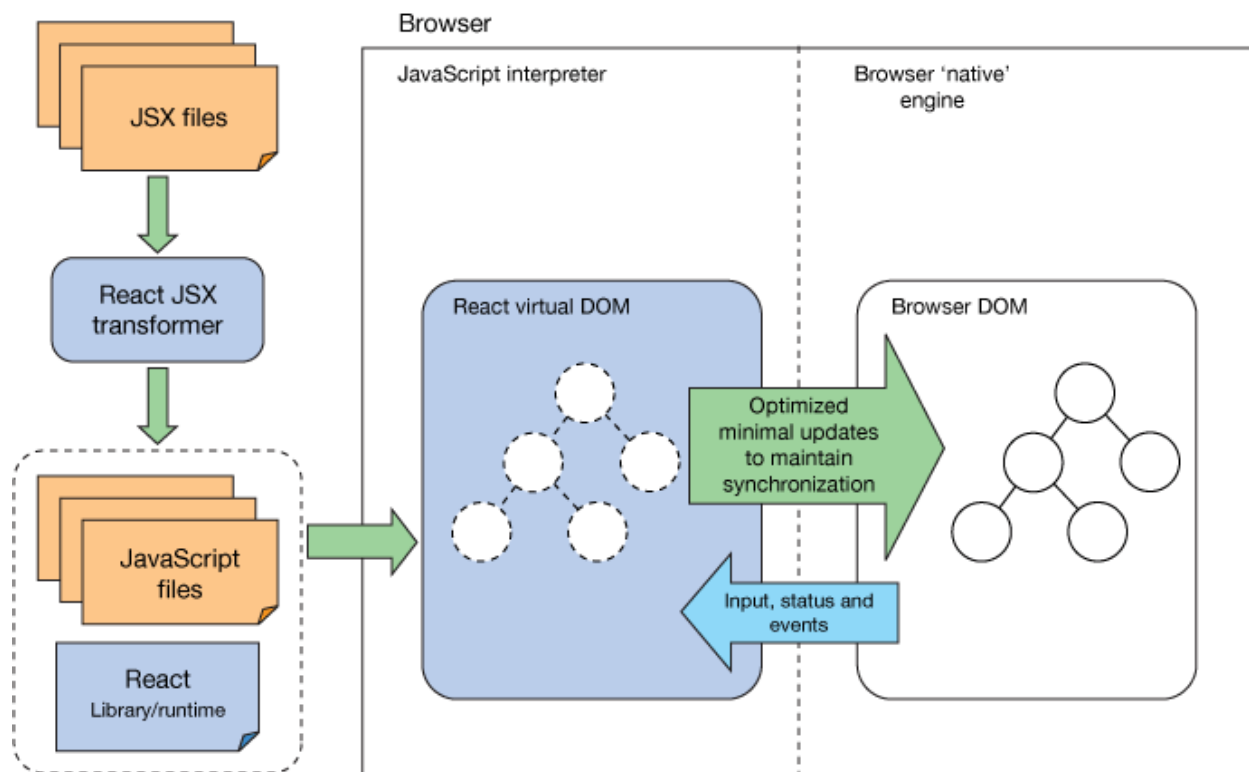
React използва виртуално представяне на действителния HTML DOM, наречен виртуален DOM. Виртуалният DOM е ефективно представяне на потребителския интерфейс в паметта. Когато състоянието на компонент се промени, React сравнява виртуалния DOM с реалния DOM и ефективно актуализира само необходимите части на потребителския интерфейс, намалявайки броя на действителните манипулации на DOM.



Фигура 3 - Виртуален ДОМ

#### 2.2.5. JSX

React използва разширение за синтаксис, наречено JSX. JSX позволява на разработчиците да пишат HTML-подобен код директно в JavaScript. Той съчетава силата на JavaScript и HTML, което улеснява описването на структурата и външния вид на компонентите на потребителския интерфейс.



Фигура 4 - Визуализация на JSX

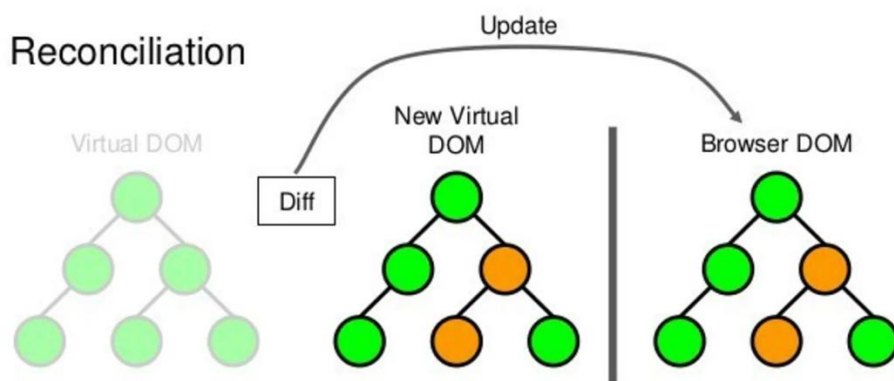
#### 2.2.6. Еднопосочен поток от данни

React следва модел на еднопосочен поток от данни. Данните протичат от родителски компоненти към дъщерни компоненти чрез подпори (свойства). Дъщерните компоненти не могат директно да променят данните, получени чрез подпори, осигурявайки предвидим и лесен за разбиране поток от данни.

#### 2.2.7. Reconciliation

React ефективно актуализира и изобразява компоненти чрез извършване на процес, наречен Reconciliation. Когато състоянието или реквизитите на компонент се променят, React определя кои части от

потребителския интерфейс трябва да бъдат актуализирани и ефективно прилага тези промени към виртуалния DOM.



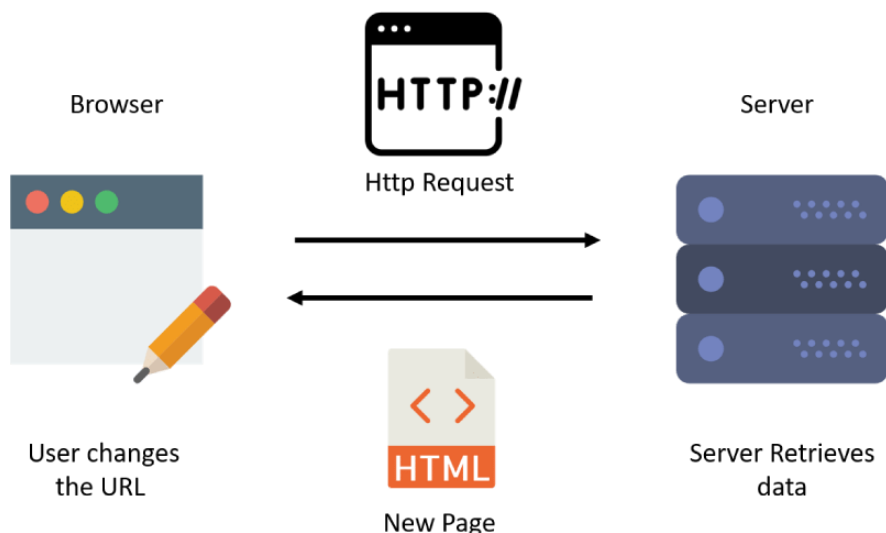
Фигура 5 - Актуализация на компоненти чрез съгласуване

## 2.2.8. React Router

React Router е популярна библиотека за маршрутизиране за React приложения. Той позволява на разработчиците да управляват маршрутизирането и навигацията в своите React приложения. React Router предоставя декларативен начин за дефиниране и управление на различни маршрути и съответните им компоненти, позволявайки безпроблемно и динамично потребителско изживяване.

React Router предоставя стабилно и гъвкаво решение за маршрутизиране на React приложения, което позволява на разработчиците да създават сложни навигационни структури, да обработват динамични URL

адреси и да осигуряват плавни преходи между различни изгледи. Той се използва широко в проекти на React за управление на маршрутизирането на приложения и осигуряване на безпроблемно потребителско изживяване.



Фигура 6 - React Router

## 2.2.9. HTML

HTML (HyperText Markup Language) е най-основният градивен елемент на мрежата. Той определя значението и структурата на уеб съдържанието. Други технологии освен HTML обикновено се използват за описване на външния вид/презентацията (CSS) или функционалността/поведението на уеб страницата (JavaScript).

„Хипертекст“ се отнася до връзки, които свързват уеб страници една с друга, или в рамките на един уеб-сайт, или между уеб-сайтове. Връзките са основен аспект на мрежата. Като качвате съдържание в Интернет и го свързвате със страници, създадени от други хора, вие ставате активен участник в World

Wide Web.

HTML използва "маркиране" за аотиране на текст, изображения и друго съдържание за показване в уеб браузър.

Описанието на документа става чрез специални елементи, наречени HTML елементи или техните маркери, които се състоят от тагове и съответстващите етикети (HTML tags) и ъглови скоби (като например елемента `<html>`). HTML елементите са основната градивна единица на кода, който изграждат уеб страниците. Чрез тях се форматира, графично оформя текста и неговите отделните части в рамките на една уеб страница, като например заглавия, цитати, текстови раздели, хипертекстови препратки и т.н. Най-често HTML елементите са групирани по двойки `<h1>` и `</h1>`.



Фигура 7 – HTML

#### 2.2.10. CSS

CSS (Cascading Style Sheets) е език за описание на стилове (език за стилови файлове, style sheet language) – използва се основно за описание на

онлайн представянето на уеббазиран документ, който написан на език за маркиране. Най-често се използва допълнително към чистия HTML, но се прилага и върху XML уебстраници и документи. Спецификацията на CSS официално се поддържа от W3C.

CSS още в началото на развитието на www започва да се добавя към стандартния HTML с цел да бъдат разделени съдържанието и структурата на уеб страниците отделно от тяхното визуално представяне.

CSS може да се пише не само в отделни файлове но и в style тагове, като в html стринга на програмата.



Фигура 8 - CSS

#### 2.2.11. JavaScript

JavaScript е широко разпространен език за програмиране, който се използва предимно за създаване на динамично и интерактивно уеб съдържание. Това е многофункционален език, който може да се използва

както за преден край, така и за бек-енд уеб разработка, както и за други цели като скриптове от страна на сървъра, разработка на игри и разработка на мобилни приложения.

JavaScript първоначално е създаден, за да добави интерактивност към уеб страниците, позволявайки на разработчиците да манипулират и променят съдържанието на уеб страница от страна на клиента, което означава в уеб браузъра на потребителя. Поддържа се от всички съвременни уеб браузъри, което го прави мощен инструмент за подобряване на потребителското изживяване в мрежата.

Езикът за програмиране е от високо ниво, което означава, че предоставя абстракции, които улесняват разработчиците да пишат код, без да се тревожат за детайлите на ниско ниво.

Той е интерпретиран език, което означава, че не е необходимо да се компилира преди да бъде изпълнен. JavaScript машината на браузъра интерпретира и изпълнява кода директно.

JavaScript е много развит език и с навлизането на технологии като Node.js и различни библиотеки и Framework-ове вече може да се използва и за програмиране от страна на сървъра. Той се превърна в основен език за уеб разработка, като дава възможност на разработчиците да създават интерактивни потребителски изживявания, да създават уеб приложения и да създават широка гама от софтуерни решения.





Фигура 9 - JS лого

## 2.2.12. Мениджър на пакетите

Мениджърът на пакети е софтуерен инструмент, който помага при управлението, инсталирането и премахването на софтуерни пакети или библиотеки в рамките на компютърна система или специфична среда за програмиране. Той опростява процеса на обработка на зависимости, като гарантира, че всички необходими компоненти или модули са налични и правилно интегрирани за гладкото функциониране на приложенията или проектите.

Мениджърите на пакети са особено важни при програмирането и разработката на софтуер, където проектите често разчитат на различни външни библиотеки или модули. Те рационализират процеса на придобиване и управление на тези зависимости, като улесняват разработчиците да се

съсредоточат върху основните си задачи, вместо да боравят ръчно с отделни пакети. SoftuniChat използва NPM като мениджър на пакетите.

NPM (Node Package Manager) е мениджър на пакети за JavaScript и мениджър на пакети по подразбиране за Node.js, среда за изпълнение, която позволява изпълнение на JavaScript код извън уеб браузър. Той служи като централно хранилище за повторно използваем JavaScript код, предоставяйки лесен начин за разработчиците да споделят, откриват и управляват зависимости в своите проекти.

С NPM разработчиците могат лесно да инсталират, актуализират и премахват пакети (известни също като модули или библиотеки), необходими за техните приложения. Тези пакети могат да варират от малки помощни функции до цялостни рамки, предоставящи решения за различни аспекти на разработката на JavaScript.



Фигура 10 - NPM лого

### 2.2.13. Използвани пакети в приложението

Проекта включва сбор от няколко пакета с помощта на които имаме готова функционалност за библиотеките.

```
"dependencies": {  
  "@ant-design/icons": "^4.6.2",  
  "@testing-library/jest-dom": "^5.11.4",  
  "@testing-library/react": "^11.1.0",  
  "@testing-library/user-event": "^12.1.10",  
  "axios": "^0.21.1",  
  "firebase": "^7.20.0",  
  "react": "^16.13.1",  
  "react-chat-engine": "^1.8.10",  
  "react-dom": "^16.13.1",  
  "react-router-dom": "^5.2.0",  
  "react-scripts": "3.4.3",  
  "web-vitals": "^1.0.1"  
},
```

Фигура 11 - лист с пакети(скрийншот от сорс код)

Axios - Axios е популярна JavaScript библиотека, използвана за правене на HTTP заявки от уеб браузър или Node.js среда. Той предоставя прост и интуитивен API за изпращане на асинхронни HTTP заявки за взаимодействие с уеб сървъри и извличане на данни.

Firebase - NPM пакетът Firebase е официалната клиентска библиотека на JavaScript за Firebase, която е цялостна платформа за разработка, предоставена от Google. Той позволява на разработчиците лесно да интегрират услугите на Firebase в своите JavaScript проекти, включително уеб и Node.js приложения.

Пакетът предоставя унифициран API и набор от функции, които позволяват на разработчиците да взаимодействат с различни услуги на Firebase

React – NPM пакетът react е официалният пакет, който предоставя основната функционалност на библиотеката React за изграждане на потребителски интерфейси в приложения на JavaScript. React е популярна JavaScript библиотека, разработена от Facebook и широко използвана за създаване на динамични и интерактивни UI компоненти.

React-chat-engine - React Chat Engine NPM пакетът е библиотека, която предоставя набор от компоненти и помощни програми за изграждане на приложения за чат или интегриране на функционалност за чат в проекти на React. Той предлага абстракционен слой за справяне със сложността на съобщенията в реално време и свързаните с чат функции, позволявайки на разработчиците да се съсредоточат върху изграждането на потребителския интерфейс и персонализирането на чат изживяването. React Chat Engine NPM пакетът използва платформата Chat Engine, която е базирана на облак бекенд услуга за изграждане на приложения за чат.

React-dom - React DOM NPM Пакетът е официалният пакет, който осигурява интеграцията между React и Document Object Model (DOM) на браузъра.

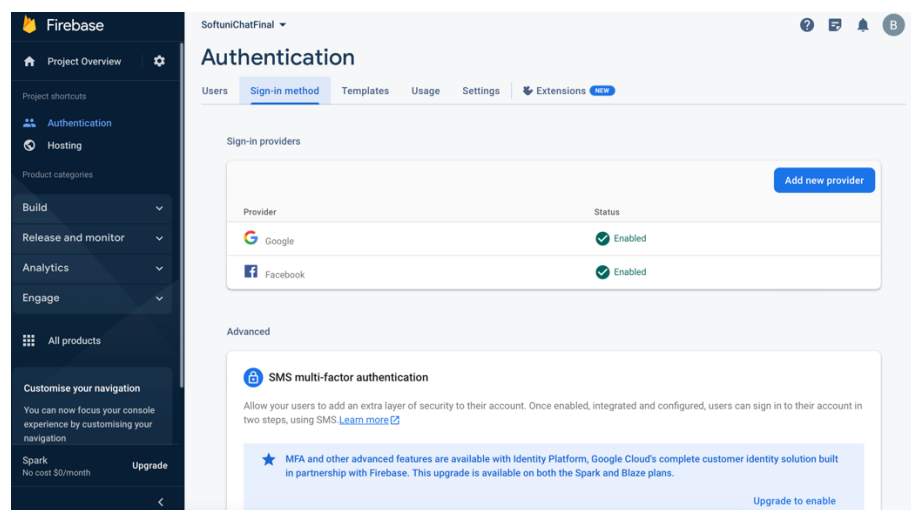
React-router-dom - React Router Dom пакетът е JavaScript библиотека, която предоставя възможности за маршрутизиране за React приложения. Това е популярно разширение на библиотеката на React Router, специално

проектирано за уеб приложения, които използват React за изграждане на потребителски интерфейси.

## 2.2.14. Firebase

Firebase е цялостна платформа за разработка, предоставена от Google, която предлага широка гама от услуги и инструменти за изграждане на уеб и мобилни приложения. Една от ключовите услуги, предоставяни от Firebase, е Firebase Authentication, която предоставя на разработчиците лесен начин за добавяне на удостоверяване и управление на потребителите към техните приложения.

Firebase Authentication поддържа различни методи за удостоверяване, включително имейл/парола, телефонен номер и доставчици на удостоверяване в социалните медии като Google и Facebook които са използвани в проекта.



Фигура 12 - Firebase Auth

Firebase осигурява безпроблемна интеграция с Google Sign-In, което позволява на потребителите да се удостоверяват със своите акаунти в Google.

С Firebase Authentication разработчиците могат да активират Google Sign-In като метод за удостоверяване в своето приложение, като просто конфигурират Firebase проекта и добавят необходимия код от страна на клиента.

Потребителите могат да влизат, като използват своите идентификационни данни за Google, а Firebase обработва процеса на удостоверяване, включително управление на токени и проверка на самоличността на потребителя.

След като бъдат удостоверени, разработчиците имат достъп до информацията на потребителя и да извършват действия като показване на потребителски профили или управление на специфични за потребителя данни. И могат да бъдат насочвани към различни страници специализирани

за дадения потребител чрез AuthContext.js (файл в сорс кода)

```
export const AuthProvider = ({ children }) => {
  const [loading, setLoading] = useState(true);
  const [user, setUser] = useState(null);
  const history = useHistory();

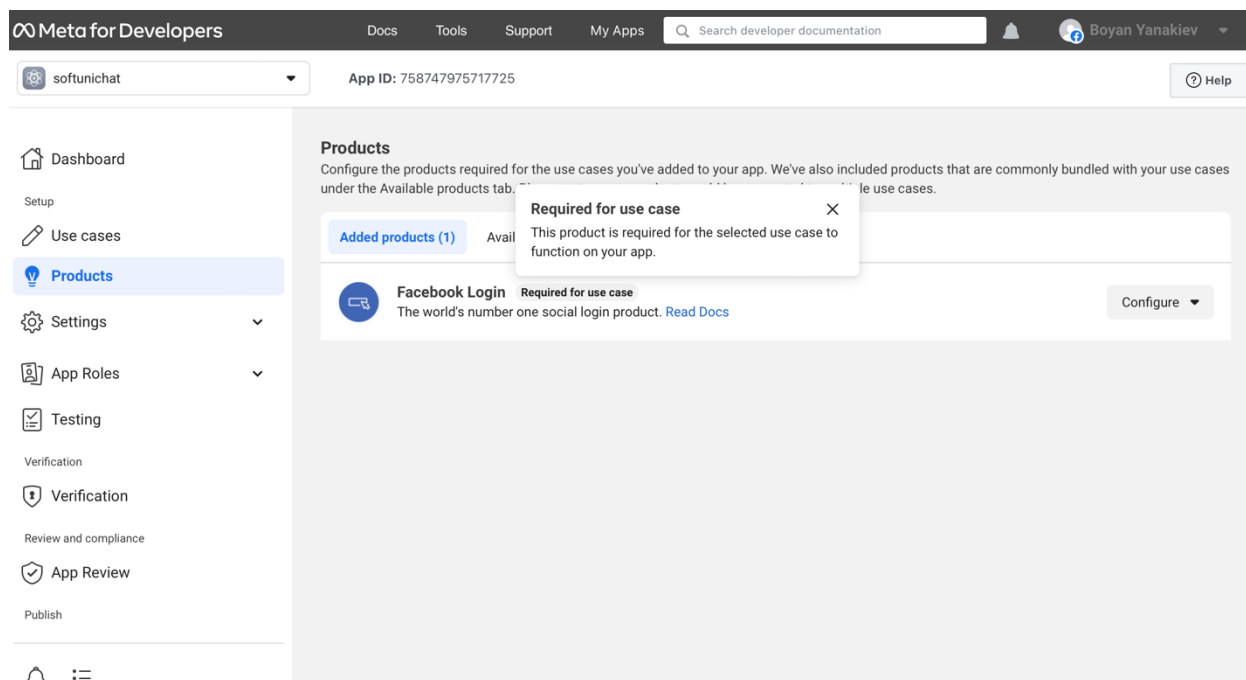
  useEffect(() => {
    auth.onAuthStateChanged((user) => {
      setUser(user);
      console.log("User", user);
      setLoading(false);
      if(user) history.push("/chats")
      else history.push("/"); console.log("No user");
    })
  }, [user, history]);

  const value = { user };

  return(
    <AuthContext.Provider value={value}>
      {!loading && children}
    </AuthContext.Provider>
  );
}
```

Фигура 13 - AuthContext.js

Firebase също поддържа регистрация с Facebook профили, обаче при нея имаме една допълнителна стъпка за интеграцията, която включва регистриране на приложение с Facebook Log-In API.



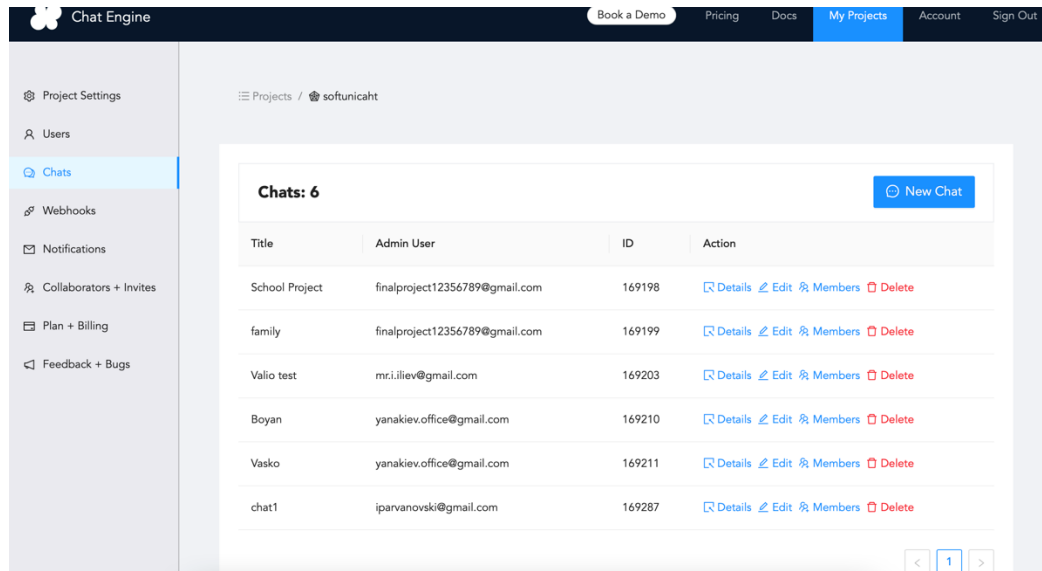
Фигура 14 - Facebook For Developers API

### 2.2.15. ChatEngine

Chat Engine е платформа и набор от инструменти, които предоставят инфраструктурата и функционалността за изграждане на приложения за чат. Той опростява процеса на интегриране на съобщения в реално време и чат функции в уеб и мобилни приложения, позволявайки на разработчиците да се съсредоточат върху създаването на персонализирано чат изживяване за своите потребители, като позволява Real time комуникация, съхраняване на



чатове и на файловете изпращани между потребителите.



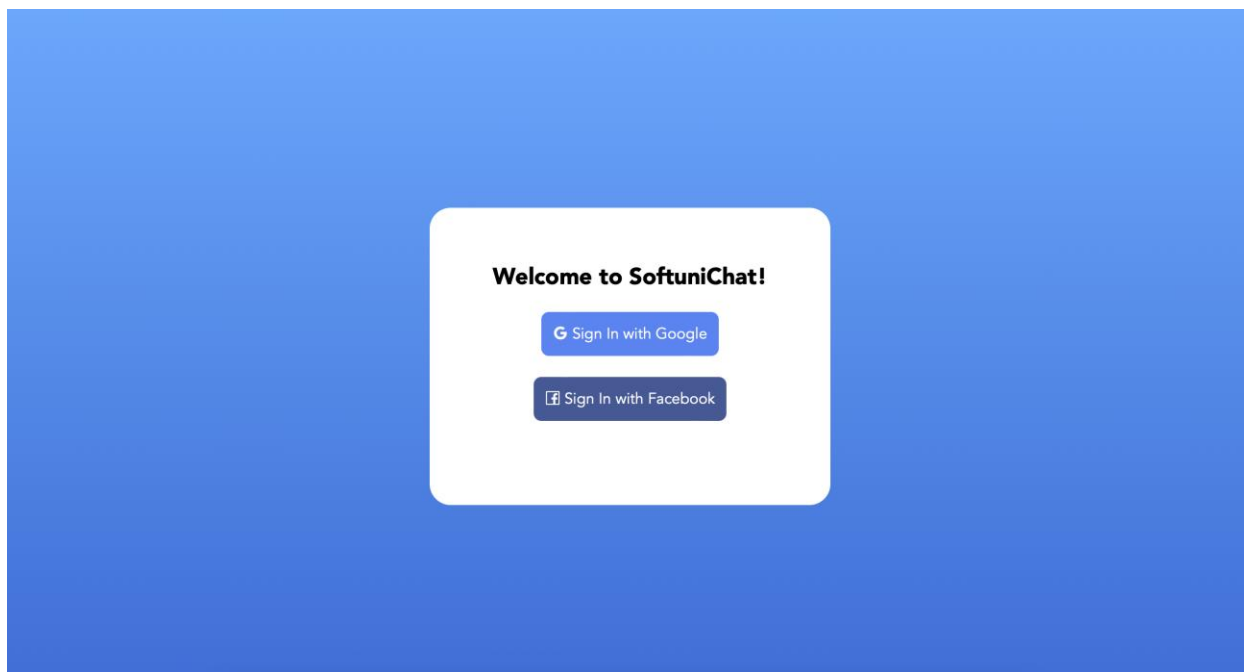
Фигура 15 - Chat Engine Dashboard

## 2.3. Задачи, които трябва да се решат за постигането на целта

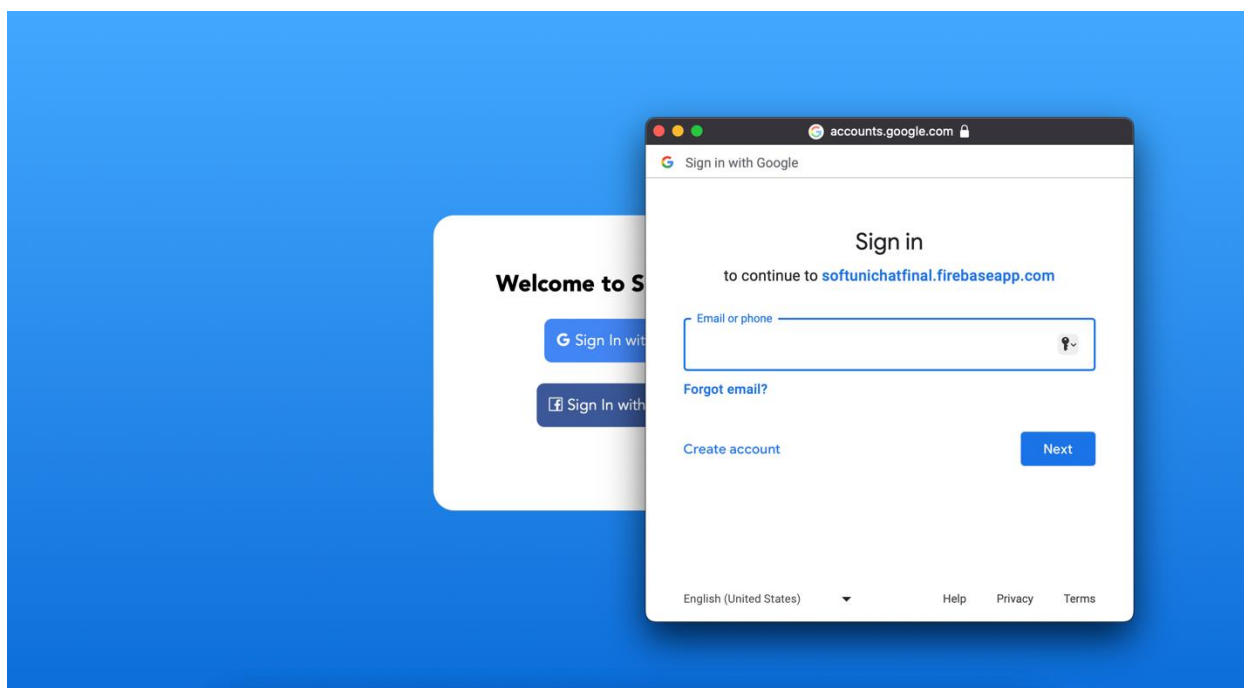
- Намиране на актуална тема и нуждата от подобен софтуер
- Избор на рентабилни и лесни за поддръжка технологии
- Проектиране на приложението
- Избор на сигурни и защитени начини за регистриране на потребителите
- Избор на back end за приложението (лесен начин за съхраняване на данни като чатове снимки и контакти)

## 2.4. Употреба на приложението

### 1. Регистрация

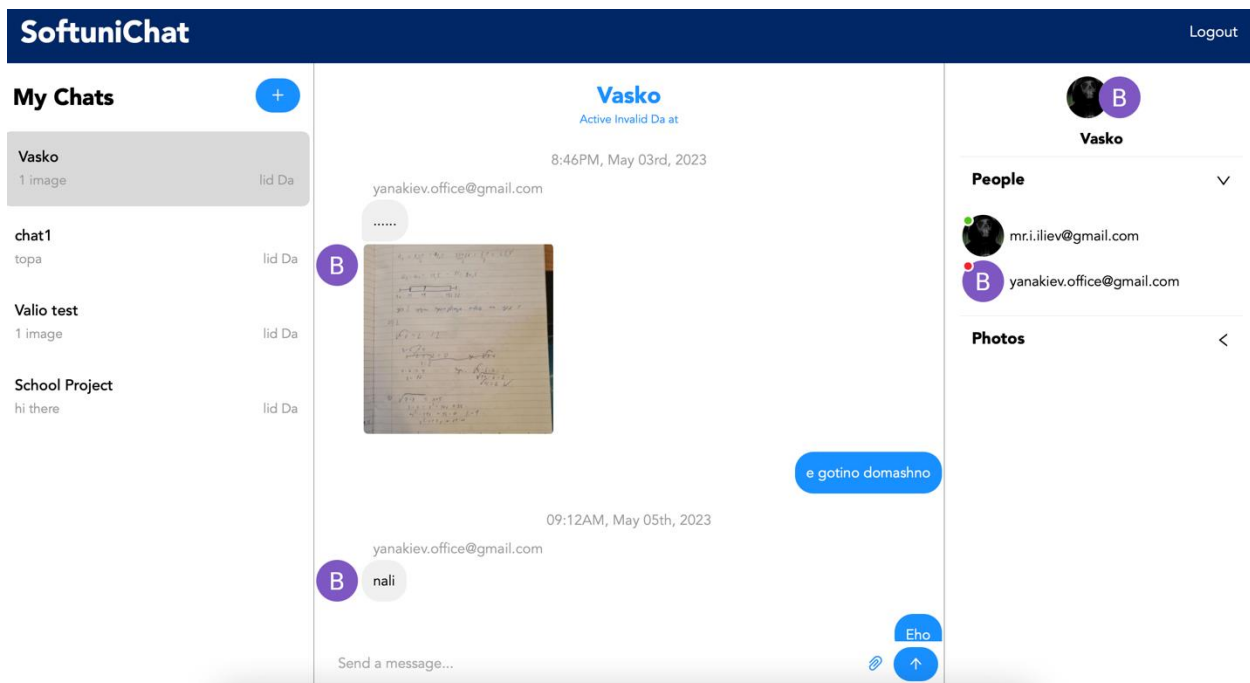


Фигура 16 - Начална страница



Фигура 17 - Регистрация

## 2. Създаване на чат с други потребители



Фигура 18 - Чат Приложение – SoftuniChat

### 3. Заключение

SoftuniChat е приложение за комуникация в училищна среда. Същата тази платформа може да бъде използвана на много различни места, не само в училищни организации. Приложението използва рентабилна, иновативна и модерна технология. То успешно успява да постигне целта си да създаде безпроблемна комуникационна платформа, която свързва хора в един и същи социален кръг, подобрява колаборацията и увеличава ефективността.

Със своя интуитивен потребителски интерфейс, възможности за изпращане на съобщения в реално време, сигурно удостоверяване и персонализирани функции, SoftuniChat трансформира начина, по който хората взаимодействат и се свързват помежду си в дадена организация или среда.

## 4. ИСТОЧНИЦИ

<https://en.wikipedia.org/wiki/JavaScript>

<https://en.wikipedia.org/wiki/HTML>

<https://en.wikipedia.org/wiki/CSS>

<https://react.dev>

[https://en.wikipedia.org/wiki/React\\_\(software\)](https://en.wikipedia.org/wiki/React_(software))

<https://firebase.google.com/docs>