**Документация**

**по**

**Групов уеб проект**

Изготвен от:

Ангел Масарлиев, клас 11а, номер 1

Божидар Иванов, клас 11а, номер3

Васил Димов, клас 11а,номер 4

**Идея на проекта**

Проектът представлява уеб страница, за извозване с камиони. Потребителите трябва да се регистрират или да влезнат в системата, за да достъпят пълния сайт. Регистрацията и входът се осъществяват чрез форми за въвеждане на имейл и парола. Сайтът предлага главно меню в горната част, което включва връзки към различни раздели на сайта като "Home", "About Us", "Our Routes", "Our Trucks", "Our Services" и "Contact Us". Това предоставя лесен достъп до информацията. Възможно е определени секции да са достъпни само за регистрирани потребители. Например, за да видите "Our Routes" (маршрутите на камионите), може да се изисква предварителна регистрация и вход в системата.

**Използвани езици и технологии**

Проектът използва различни езици и технологии за създаване на уеб приложение за регистрация и вход на потребители.

Първият пример за използван език е HTML. Той се ползва за създаване на структурата на уеб страниците. В проекта HTML служи за дефиниране на различни елементи като форми за вход и регистрация, бутони, текстови полета и други.

Следващият език, използван в проекта, е CSS (Cascading Style Sheets). Той е от полза за стилизиране и оформление на уеб страниците. В проекта чрез CSS се прилагат стилове за цветове, шрифтове, разположение на елементите и общо визуално представяне на интерфейса.

JavaScript е третият използван език в проекта. Той добавя интерактивност към уеб страниците и управлява събития като кликвания на бутони. В проекта JavaScript е нужен за валидация на форми, превключване между форми за вход и регистрация, както и връзка със системата за аутентикация на Firebase.

Firebase е технология, използвана за управление на аутентикацията на потребителите. С помощта на Firebase потребителите могат да се регистрират и да влизат в системата чрез имейл и парола. Това осигурява сигурно и бързо управление на потребителските идентичности.

Допълнително, проектът използва Google Fonts за зареждане на специфични шрифтове към уеб страниците. Това позволява лесно и ефективно стилизиране на текстовите елементи в приложението.

Използването на CDN (Content Delivery Network) позволява зареждането на външни библиотеки и ресурси директно от мрежата, като например Firebase SDK. Това осигурява бързо и надеждно зареждане на външни зависимости.

Тези езици и технологии работят заедно, за да създадат функционалността на уеб приложението за регистрация и вход на потребители, като предоставят удобство, интерактивност и сигурност за потребителите.

**Описание на кода**

Проекта е разделен на пет части състоящи се от 3 папки и два html файла. Първата папка озаглавена css съдържа два css файла: style.css и styleSait.css. Файлът style.css съдържа стилове, които се използват за стилизиране на формите за вход и регистрация в уеб приложението. Важните части от този CSS файл включват:

1. Нулиране и общи стилове:

{

margin: 0;

padding: 0;

box-sizing: border-box;

font-family: „Poppins“, sans-serif;

}

Тези стилове нулират маргините и падинга на всички елементи и задават box-sizing на border-box, за да се управлява правилно размерът на елементите. Също така, използва се шрифта "Poppins" за всички елементи на страницата.

1. Фон на страницата:

body {

display: flex;

justify-content: center;

align-items: center;

min-height: 100vh;

background: url(„../images/bg.jpg“) no-repeat;

background-size: cover;

background-position: center;

}

Тези стилове центрират съдържанието на страницата вертикално и хоризонтално. Задават минимална височина (min-height) от 100% на вида на прозореца (vh) и задават фоново изображение (bg.jpg), което се покрива и позиционира по центъра.

1. Стилове за контейнера (wrapper):

.wrapper {

position: relative;

width: 440px;

height: 470px;

margin-bottom: 65px;

background: transparent;

border: 2px solid rgba(255, 255, 255, .5);

border-radius: 20px;

backdrop-filter: blur(20px);

box-shadow: 0 0 30px rgba(0, 0, 0, .5);

display: flex;

justify-content: center;

align-items: center;

overflow: hidden;

transition: height .2s ease;

}

Тези стилове дефинират външния контейнер (wrapper), който съдържа формите за вход и регистрация. Задават се размери, граници, ефекти на изчистване на задното осветление (backdrop-filter) и сянка. Осигурява се центриране на съдържанието в контейнера

Другият файл в папката styleSait.css съдържа стилове за общия дизайн на уеб страниците, включително хедър, навигация, съдържание и футър. Някои от важните части включват:

1. Стилове за хедъра (header):

#header {

padding: 0px 0px 5px;

}

#header .top\_banner {

background: url(../images/header.png) no-repeat left top;

height: 150px;

padding: 50px 35px 0 560px;

border: #a4a5a5 solid 1px;

}

Тези стилове задават външния вид на хедъра, включително фоновото изображение и размерите на банера.

1. Стилове за навигацията

.navigation {

margin: 0px 0px 20px;

border: #adb1b2 solid 1px;

padding: 15px 5px;

background: #292121;

}

Тези стилове дефинират изгледа на навигационното меню, включително границите, падинга и фона.

1. Стилове за съдържанието на страницата:

#page\_content {

padding: 10px;

background: url(../images/body.jpg) repeat;

border: #a4a5a5 solid 1px;

}

Тези стилове определят външния вид на основното съдържание на страницата, включително фоновото изображение и границите.

Това са само някои от важните части от стиловете в двата CSS файла, които са отговорни за стилизирането на различните елементи на уеб страниците в проекта.

Следващата папката на име images съдържа PNG, JPG и GIF изображения, съдържа важни визуални ресурси, които се използват за стилизиране и представяне на уеб страниците в проекта.

В последната папка озаглавена scipts се съдържат три JavaScript файла, които имат различни функционалности и са част от уеб приложението за регистрация и вход на потребители, което използва Firebase за управление на автентикацията.

1. register.js:

- Този файл съдържа логиката за регистрация на нов потребител в системата. Използва се Firebase Authentication за създаване на нов потребителски акаунт. Важните части от кода са подобни на `login.js`:

- Импортиране на необходимите функции от Firebase SDK, като `initializeApp` и `createUserWithEmailAndPassword`.

- Конфигуриране на Firebase със специфичните данни за проекта (`firebaseConfig`).

- Инициализиране на Firebase приложението.

- Получаване на обект за управление на автентикацията (`auth`).

- Добавяне се изчакване за събитието "click" на бутона за регистрация (`btnRegister`). При кликване се взимат потребителското име (е-мейл) и парола от полетата във формата и се изпращат към Firebase функцията `createUserWithEmailAndPassword` за създаване на нов акаунт.

- При успешна регистрация се извежда съобщение за успешно създаден акаунт.

1. login.js:

* Този файл съдържа логиката за вход на потребител в системата. Използва се Firebase Authentication за проверка на потребителските данни (потребителско име и парола). Важните части от кода са:
* Импортиране на необходимите функции от Firebase SDK, като `initializeApp` и `signInWithEmailAndPassword`.
* Конфигуриране на Firebase със специфичните данни за проекта (`firebaseConfig`).
* Инициализиране на Firebase приложението.
* Получаване на обект за управление на автентикацията (`auth`).
* Добавяне на слушател за събитието „click“ на бутона за вход (`btnLogin`). При кликване се взимат потребителското име и парола от полетата във формата и се изпращат към Firebase функцията `signInWithEmailAndPassword` за аутентикация.
* При успешен вход потребителят се пренасочва към друга страница (`sait.html`), а при грешка се извежда съобщение за грешка.

1. script.js:

- Този файл съдържа обща логика за превключване между формите за вход и регистрация на потребителите. Важните части от кода са:

- Избиране на елементи от DOM (Document Object Model), като `.wrapper`, `.login-link` и `.register-link`.

- Добавяне на слушатели за събития "click" на връзките за превключване между формите.

- При кликване на връзката за регистрация се добавя клас "active" към `.wrapper`, който променя стиловете на формата за регистрация.

- При кликване на връзката за вход се премахва класът "active" от `.wrapper`, връщайки формата към стандартния си изглед за вход.

Тези JavaScript файлове работят заедно, за да осигурят функционалността за вход и регистрация на потребители в уеб приложението, като използват Firebase за автентикация и DOM манипулации за превключване между различните форми.

Първия от двата файла index.html представлява основата на уеб страницата за вход и регистрация на потребители. Тук са обясненията за важните части от кода:

1. \*\*Document Type Declaration (`<!DOCTYPE html>`)\*\*:

- Това е декларацията за тип на документа, която указва, че това е HTML5 документ.

2. \*\*HTML Root Element (`<html>`)\*\*:

- Отваря основния HTML елемент, с указание на атрибута `lang=“en“`, който указва, че езикът на документа е английски.

3. \*\*Head Section (`<head>`)\*\*:

- Секцията `<head>` съдържа метаданни и ресурси за уеб страницата, като:

- `<meta charset=“UTF-8“>`: Указва Unicode UTF-8 като кодиране за знаци.

- `<meta name=“viewport“ content=“width=device-width, initial-scale=1.0“>`: Задава настройките за мащабиране и размери на устройства.

- `<title>Log In | Registration</title>`: Определя заглавието на уеб страницата.

- `<link rel=“stylesheet“ href=“css/style.css“>`: Свързва външен CSS файл за стилизиране на страницата.

- Заявки за предварително свързване с Google Fonts, за да се използва шрифтът „Poppins“ в уеб страницата.

- Зарежда два JavaScript файла (`script.js`, `register.js`, `login.js`), които са включени чрез модулна система (`type=“module“`).

4. \*\*Body Section (`<body>`)\*\*:

- Секцията `<body>` съдържа съдържанието на уеб страницата.

- Елементът `<div class=“wrapper“>` съдържа две форми (`login` и `register`), които се показват/скриват в зависимост от действията на потребителя.

- \*\*Login Form\*\* (`<div class=“form-box login“>`):

- Форма за вход с полета за електронна поща (`<input type=“email“ id=“email“>`) и парола (`<input type=“password“ id=“password“>`).

- Контроли за запомняне на потребителя и линк към „Забравена парола“.

- Бутон за изпращане на формата за вход (`<button type=“submit“ class=“btn“ id=“btnLogin“>Login</button>`).

- Линк за прехвърляне към формата за регистрация (`<a href=“#“ class=“register-link“>Register</a>`).

- \*\*Registration Form\*\* (`<div class=“form-box register“>`):

- Форма за регистрация с полета за потребителско име (`<input type=“text“ id=“username“>`), електронна поща (`<input type=“email“ id=“email“>`) и парола (`<input type=“password“ id=“password“>`).

- Контрол за приемане на „Условия и правила“.

- Бутон за изпращане на формата за регистрация (`<button type=“submit“ class=“btn“ id=“btnRegister“>Register</button>`).

- Линк за прехвърляне към формата за вход (`<a href=“#“ class=“login-link“>Login</a>`).

Този HTML файл е основната структура на уеб страницата за вход и регистрация на потребители, като използва външен CSS за стилизация и JavaScript за манипулиране на поведението и взаимодействието с потребителите.

Последна част от кода на проекта е sait.html Този HTML код представлява основната структура на уеб страница за управление на информация свързана с компания за превозни услуги, както и предлагани маршрути и услуги. Давам обяснение за най-важните части от кода:

1. \*\*Document Type Declaration (`<!DOCTYPE html>`)\*\*:

- Това е декларацията за тип на документа, указваща, че това е HTML документ.

2. \*\*HTML Root Element (`<html>`)\*\*:

- Отваря основния HTML елемент, без указан език (`lang` атрибутът липсва).

3. \*\*Head Section (`<head>`)\*\*:

- Секцията `<head>` съдържа метаданни и ресурси за уеб страницата, като:

- `<meta charset=“UTF-8“>`: Указва Unicode UTF-8 като кодиране за знаци.

- `<title>Tracky Trucks</title>`: Заглавието на уеб страницата.

- `<meta name=“description“ content=“A description of your website“>`: Описание на уеб страницата за търсачки.

- `<meta name=“keywords“ content=“keyword1, keyword2, keyword3“>`: Ключови думи за търсачки.

- `<link href=“css/styleSait.css“ rel=“stylesheet“ type=“text/css“>`: Връзка към външен CSS файл за стилизиране на страницата.

4. \*\*Body Section (`<body>`)\*\*:

- Секцията `<body>` съдържа съдържанието на уеб страницата.

- \*\*Wrapper (`<div id=“wrapper“>`)\*\*:

- Опаковащ елемент, който съдържа цялото съдържание на страницата.

- \*\*Header (`<div id=“header“>`)\*\*:

- Секция с хедър на страницата, включваща банер и заглавие.

- \*\*Page Content (`<div id=“page\_content“>`)\*\*:

- Основното съдържание на страницата.

- \*\*Navigation (`<div class=“navigation“>`)\*\*:

- Навигационно меню с връзки към различни раздели на уеб страницата.

- \*\*Left Sidebar (`<div class=“left\_side\_bar“>`)\*\*:

- Лява странична лента с различни секции и менюта.

- \*\*Right Section (`<div class=“right\_section“>`)\*\*:

- Дясна секция с основно съдържание и различни разделители.

- \*\*Footer (`<div id=“footer“>`)\*\*:

- Подвал на страницата с авторско право и информация за дизайнера.

Този HTML код описва типична уеб страница за представяне на фирма за превозни услуги, предлагаща информация за маршрути, услуги и контакти. Той използва външен CSS файл за стилизиране и се стреми към организиране на съдържанието чрез различни секции и менюта.

**Изглед на сайта**

1. Изглед на login формата

Картина, която съдържа екранна снимка, Компютърна игра, Дигитална композиция, Стратегическа видеоигра

Описанието е генерирано автоматично

1. Изглед на registration формата

Картина, която съдържа текст, екранна снимка, виолетов, музика

Описанието е генерирано автоматично

1. Изглед на сайта

Картина, която съдържа текст, екранна снимка, Уебсайт, софтуер

Описанието е генерирано автоматично

**Използвани източници:**

[**https://firebase.google.com/docs**](https://firebase.google.com/docs)

[**https://www.w3schools.com/css/**](https://www.w3schools.com/css/)

[**https://www.w3schools.com/html/**](https://www.w3schools.com/html/)

[**https://www.w3schools.com/html/**](https://www.w3schools.com/html/)