**ПРОЕКТ**

**по Разработка на софтуер**

на тема:

**„Групови уеб проекти”**

на

Атанас Чолаков, клас 11а, № 2,

специалност ,,Приложно програмиране“

Ученик: .......................................

/Атанас Чолаков/

Пловдив, 2024 г.

СЪДЪРЖАНИЕ

1. Увод……………………………………………………………..2
2. Специфики на разработката…………………………………..2
3. Реализация на логическата функционалност…..……………4
4. Заключение………………………………………………….....8
5. Литература……………………………………………………..9
6. Увод.

Проектът представлява уеб приложение – автокъща, с Login система. За направата му са използвани езиците за програмиране HTML, CSS и JavaScript, библиотеката bootstrap за адаптивността на страниците в приложението и FireBase за база от данни. Основната цел на приложението е да предостави Login система за потребителите, като в същото време предоставя начална страница с обяви за автомобили, за да осигури интересна и пълноценна начална точка за потребителите. Потребителите имат възможност да създадат нов профил или да влезнат със съществуващ такъв. Логин формата е централна част от приложението, като осигурява сигурен достъп до персонализирани функционалности. Началната страница е насочена към предоставянето на обяви за различни автомобили. Това осигурява първична информация и интерес за потребителите.

1. Специфики на разработката.

Проектът е фокусиран върху създаването на Login система, интегрирана в уеб приложение за автокъща. Разработката се базира на използването на различни технологии, насочени към сигурността, управлението на потребителски профили и предоставянето на начална страница с обяви за автомобили.

За съхранение на потребителските данни и управление на автентикацията е използвана базата от данни Firebase, която осигурява бърз и сигурен достъп до информацията. Това включва създаване на нови потребителски профили, управление на сесии и проверка на валидността на данните за вход.

HTML, CSS и JavaScript са използвани за създаване на потребителски интерфейс, който е интуитивен за потребителите и ефективно взаимодейства с логин системата. Той е представен на фигура 1 и фигура 2. Bootstrap е внедрен за бързо стилизиране и адаптиране на дизайна към различни устройства.

Картина, която съдържа текст, екранна снимка, дизайн

Описанието е генерирано автоматично

*Фигура 1 – Страница за регистрация на потребител*

Картина, която съдържа текст, екранна снимка, дизайн

Описанието е генерирано автоматично

*Фигура 2 – Страница за влизане в профил*

Основната страница на приложението предоставя обяви за автомобили. Това създава интересна и пълноценна начална точка за потребителите, които влизат в системата. На фигура 3 е представена началната страница след влизане в профил.

Използваните технологии са избрани и мотивирани от техните възможности за бързо разработване, сигурност и удобство за потребителите.

Картина, която съдържа превозно средство, Сухопътно превозно средство, текст, колело

Описанието е генерирано автоматично

*Фигура 3 – Начална страница на уеб приложението*

1. Реализация на логическата функционалност.

JavaScript код за регистрация на потребителите:

 import { initializeApp } from "https://www.gstatic.com/firebasejs/10.11.1/firebase-app.js";

 import { getAuth, createUserWithEmailAndPassword } from "https://www.gstatic.com/firebasejs/10.11.1/firebase-auth.js";

 const firebaseConfig = {

   apiKey: "AIzaSyD3k9JiV7uyP1nnfG7uVy9ELbdudRNET\_U",

   authDomain: "login-7ecc5.firebaseapp.com",

   projectId: "login-7ecc5",

   storageBucket: "login-7ecc5.appspot.com",

   messagingSenderId: "1039991397912",

   appId: "1:1039991397912:web:efc5b6ae4c869b6039ed6d"

 };

 // Initialize Firebase

 const app = initializeApp(firebaseConfig);

 const auth = getAuth(app);

 const submit = document.getElementById('submit');

 submit.addEventListener("click", function(event) {

     event.preventDefault();

    const email = document.getElementById('email').value;

    const password = document.getElementById('password').value;

    createUserWithEmailAndPassword(auth, email, password)

    .then((userCredential) => {

      // Signed up

      const user = userCredential.user;

      alert("Creating Account...")

      window.location.href = "grand.html";

    })

    .catch((error) => {

      const errorCode = error.code;

      const errorMessage = error.message;

      alert(errorMessage)

    });

 })

Кодът представлява имплементацията на функционалностите за регистрация на потребители чрез Firebase Authentication.

Първоначално се импортират необходимите функции от Firebase SDK, като initializeApp и getAuth от модула firebase-app.js, и createUserWithEmailAndPassword от модула firebase-auth.js.

След това се дефинира конфигурацията на Firebase за уеб приложението, която включва ключ, домейн, идентификатор на проекта и други настройки, необходими за връзка със сървърите на Firebase.

След като конфигурацията е налице, се инициализира Firebase приложението с функцията initializeApp, като се подава конфигурацията като аргумент. След това се взима auth обекта, който представлява аутентикационния модул на Firebase.

След това кодът добавя „слушател на събитие“ (EventListener) за бутона с идентификатор "submit". При кликване на бутона, се изпълнява функция, която предотвратява стандартното поведение на формата с event.preventDefault().

След това се взимат стойностите на полетата за имейл и паролата от HTML формата.

С функцията createUserWithEmailAndPassword се създава нов потребител в системата, като се предоставя имейлът и паролата. Ако операцията е успешна, се връща обект с данни за потребителя и се извежда съобщение за успешно създаване на акаунт. Последва пренасочване към началната страницата – "grand.html".

В случай на грешка при регистрацията, се хваща грешката и се извежда съобщение за нея. На фигура 4 е представена примерно извеждане на съобщение с грешка.

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

Описанието е генерирано автоматично

*Фигура 4 – Примерна грешка при регистрация на потребител*

JavaScript код за влизане в профил:

 import { initializeApp } from "https://www.gstatic.com/firebasejs/10.11.1/firebase-app.js";

 import { getAuth, signInWithEmailAndPassword } from "https://www.gstatic.com/firebasejs/10.11.1/firebase-auth.js";

 const firebaseConfig = {

   apiKey: "AIzaSyD3k9JiV7uyP1nnfG7uVy9ELbdudRNET\_U",

   authDomain: "login-7ecc5.firebaseapp.com",

   projectId: "login-7ecc5",

   storageBucket: "login-7ecc5.appspot.com",

   messagingSenderId: "1039991397912",

   appId: "1:1039991397912:web:efc5b6ae4c869b6039ed6d"

 };

 // Initialize Firebase

 const app = initializeApp(firebaseConfig);

 const auth = getAuth(app);

 const submit = document.getElementById('submit');

 submit.addEventListener("click", function(event) {

     event.preventDefault();

    const email = document.getElementById('email').value;

    const password = document.getElementById('password').value;

    signInWithEmailAndPassword(auth, email, password)

    .then((userCredential) => {

      // Signed up

      const user = userCredential.user;

      alert("Logging In...")

      window.location.href = "grand.html";

    })

    .catch((error) => {

      const errorCode = error.code;

      const errorMessage = error.message;

      alert(errorMessage)

      // ..

    });

 })

Кодът за влизане в профил е подобен на този за регистрация, като разликите се отнасят до използваните функции и техните аргументи.

За разлика от функцията createUserWithEmailAndPassword, която се използва за регистрация на нов потребител, тук се използва функцията signInWithEmailAndPassword, която се използва за влизане в системата с вече съществуващ потребителски акаунт.

По същия начин, след успешно влизане в системата, се изписва съобщение за успешно влизане, при грешка се изписва нейният вид.

Останалите части от кода, като инициализацията на Firebase, взимането на стойности от HTML формата и обработката на възможни грешки, са аналогични на тези в кода за регистрация.

Примерно поле за въвеждане на данни в HTML файловете:

<div data-mdb-input-init class="form-outline form-white mb-4">

                      <input type="email" placeholder="Enter email..." id="email" class="form-control form-control-lg" />

                      <label class="form-label mt-2" for="email"><h5>Email</h5></label>

                    </div>

Кодът представлява поле за въвеждане на информация, в случая за имейл адрес. Той включва входно поле с валидация за формат на email адреса и лейбъл, който описва полето. Стилизиран е чрез Bootstrap класовете за постигане на единно и привлекателно визуално представяне. Другите полета за въвеждане на информация са направени по аналогичен начин.

1. Заключение.

Проектът представлява уеб приложение за автокъща, което включва система за създаване и влизане в акаунти и начална страница с обяви за автомобили. Използвани са HTML, CSS, JavaScript, Bootstrap и Firebase за съхранение на данни и управление на потребителските профили.

Логин системата е осигурена чрез Firebase Authentication, като потребителите имат възможност за създаване на нови профили и влизане със съществуващи такива. Началната страница предоставя обяви за автомобили и създава интересна и пълноценна начална точка за потребителите.

Използваните технологии и инструменти са избрани поради тяхната ефективност, удобство за потребителите и възможност за бързо разработване на приложението. В резултат се получава функционално и лесно достъпно уеб приложение за автокъща, което предоставя възможност за управление на потребителски профили и преглед на обяви за автомобили.

1. Литература.

W3Schools (<https://www.w3schools.com/>)

MDN Web Docs (<https://developer.mozilla.org/en-US/docs/Web/JavaScript>)

Firebase Documentation (<https://firebase.google.com/docs>)

Firebase YouTube Channel (<https://www.youtube.com/user/Firebase>)