

# **Web Application Development (WAD621S) Assignment Part B**

## **Documentation**



**NAMIBIA UNIVERSITY  
OF SCIENCE AND TECHNOLOGY**

**Practical lecturer: Mrs N. Nashandi**

**Done By: Keith Craig Van Niekerk,  
219002444**

# Table of content

<b>1) Introduction.....</b>	<b>2</b>
<b>1.1) Login details.....</b>	<b>2</b>
<b>2) User Experience.....</b>	<b>3</b>
<b>2.1) Login Screen.....</b>	<b>3</b>
<b>2.2) Inventory.....</b>	<b>4</b>
<b>2.3) Create new profile.....</b>	<b>6</b>
<b>3) Functionality.....</b>	<b>8</b>
<b>3.1) Authentication.....</b>	<b>8</b>
<b>3.2) CURD and Queries.....</b>	<b>11</b>
<b>4) Final review.....</b>	<b>14</b>

## **1) Introduction**

For this assignment we were tasked with creating a website that allows create, read, update and delete operations to take place. All the webpages used in this assignment were designed with responsiveness in mind. It should also have a login page with the ability to authenticate registered users and it should query the back-end at least two times. For the purpose of this assignment I decided to create a car dealer website that keeps track of the dealers inventory. The login screen will allow you to sign in and access the inventory list. If the login is successful they will be directed to the inventory list where they can view cars in inventory, update entries and delete entries of cars already sold, add new cars to the inventory list as well as add new users(More employees).

### **1.1) Login details**

Please test login with the details below.

#### **User 1**

Email: [testing@gmail.com](mailto:testing@gmail.com)

Password: 12345678

#### **User 2**

Email: [student@gmail.com](mailto:student@gmail.com)

Password: student123

**Please feel free to add and test a new user once logged in.**

Please note new users can only be added by existing users, therefore please sign in with the above users to create a new user. This is to prevent unauthorised access.


## 2) User Experience

### 2.1) Login Screen

The login screen was designed to be responsive. It consists of a welcome message and asks the users to please sign in below. It is followed by the company logo and the login form. The login form consists of a email and password. Next is the Login button which captures the data inputted and verifies it with the back-end(firebase). As most login pages have a forgot password button, so does this one. When it is pressed it displays a popup message asking the user to contact an admin followed by an email address.

Welcome to Craig's World

Please sign in below



Email

Enter Email

Password

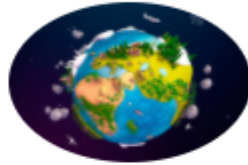
Enter Password

Login

Forgot Password?

## Welcome to Craig's World

Please sign in below



Email

Password

Login

[Forgot Password?](#)

### 2.2) Inventory

The inventory screen consists of a navigation bar with 3 fields Inventory, Add user and log out. The inventory is where the Create, Read, Update and Delete(CRUD) functionality takes place. Here the staff members are able to view the current stock, edit details about the current stock or delete it for if the car is already sold. It also gives the staff member the ability to add new stock to the inventory list. The add user is used to create new profiles so that new staff can also have access to this system. Log out takes the user out of the inventory site and back to the login screen.

## Welcome to Craig's World

Car dealer website

Current inventory

Manufacturer and Model	Chassis Number	Year	Price	
------------------------	----------------	------	-------	--

### Add new stock here

Manufacture and Model

Chassis Number

Year

Price

Price

Submit

[Inventory](#)[Add User](#)[Log Out](#)

## Welcome to Craig's World

Car dealer website

Current inventory

Manufacturer and Model	Chassis Number	Year	Price
------------------------	----------------	------	-------

Add new stock here

Manufacture and Model

Chassis Number

Year

Price


Submit

### 2.3) Create new profile

This page was designed to be responsive. The create new profile page consists of a header informing the user that they are on the page to create a new profile as well as asking them to complete the form below. The form consists of email and password. After completing the form the user can either create the profile or cancel and return to the Inventory site.

### Creating a profile

Please complete the form below to create a profile



---

**Email**


**Password**

[Cancel](#)[Create account](#)



## Creating a profile

Please complete the form below to create a profile



---

**Email**

**Password**

[Cancel](#)[Create account](#)

### 3.) Functionality and implementation

This assignment required us to design a website with the functionality below. It was implemented using html, css and JavaScript. The method used for authentication as well as storing new users was via firebase.

#### 3.1) Authentication

The website should only be accessible by staff at Craig's World. Therefore it will need to have some sort of authentication. Email and password were chosen as the authentication methods.

```

<script type="module">
// Import the functions you need from the SDKs you need
import { initializeApp } from "https://www.gstatic.com/firebasejs/9.2.0/firebase-app.js";
import { getAuth, createUserWithEmailAndPassword } from "https://www.gstatic.com/firebasejs/9.2.0/firebase-auth.js";
// TODO: Add SDKs for Firebase products that you want to use
// https://firebase.google.com/docs/web/setup#available-libraries

// Your web app's Firebase configuration
const firebaseConfig = {
  apiKey: "AIzaSyCM6h3qAxMMh-GZU8ngXDmRRY7luQE48Uo",
  authDomain: "authe-9e76a.firebaseio.com",
  projectId: "authe-9e76a",
  storageBucket: "authe-9e76a.appspot.com",
  messagingSenderId: "274908983184",
  appId: "1:274908983184:web:2f8b62f85cbc5aa3ed4d15"
};

// Initialize Firebase
const app = initializeApp(firebaseConfig);
const auth = getAuth(app);

document.getElementById("signup").addEventListener('click', function(){
  const email = document.getElementById("email").value
  const password = document.getElementById("password").value

  createUserWithEmailAndPassword(auth, email, password)
    .then((userCredential) => {
      // Signed in
      const user = userCredential.user;
      // ...
      console.log("created")
      window.alert("Account successfully created")
    })
    .catch((error) => {
      const errorCode = error.code;
      const errorMessage = error.message;
      // ..
      console.log(errorCode + errorMessage)
    });
})

```

(JavaScript used to create new user via firebase)

```

<script type="module">
  // Import the functions you need from the SDKs you need
  import { initializeApp } from "https://www.gstatic.com/firebasejs/9.2.0/firebase-app.js";
  import { getAuth, signInWithEmailAndPassword } from "https://www.gstatic.com/firebasejs/9.2.0/firebase-auth.js";
  // TODO: Add SDKs for Firebase products that you want to use
  // https://firebase.google.com/docs/web/setup#available-libraries

  // Your web app's Firebase configuration
  const firebaseConfig = {
    apiKey: "AIzaSyCM6h3qAxMMh-GZU8ngXDmRRY7luQE48Uo",
    authDomain: "authe-9e76a.firebaseio.com",
    projectId: "authe-9e76a",
    storageBucket: "authe-9e76a.appspot.com",
    messagingSenderId: "274908983184",
    appId: "1:274908983184:web:2f8b62f85cbc5aa3ed4d15"
  };

  // Initialize Firebase
  const app = initializeApp(firebaseConfig);
  const auth = getAuth(app);

  document.getElementById("loginBut").addEventListener('click', function(){
    const email1 = document.getElementById("emaillogin").value
    const password1 = document.getElementById("passwordlogin").value

    signInWithEmailAndPassword(auth, email1, password1)
      .then((userCredential) => {
        // Signed in
        const user = userCredential.user;
        // ...
        console.log("logged in")
      })
      .catch((error) => {
        const errorCode = error.code;
        const errorMessage = error.message;
        console.log(errorCode + errorMessage)
      });
  });
</script>

```

(JavaScript used to login via firebase)

### Authentication

Users   Sign-in method   Templates   Usage

Add user

Identifier	Providers	Created ↓	Signed in	User UID
craign2801@gmail.com	✉	4 Nov 2021	4 Nov 2021	wrfS9r4SFjWfkyZYbhPXNbTgbV2
student@gmail.com	✉	4 Nov 2021	4 Nov 2021	05TpkMaSd9NjIBkA48s3Aa9ATae2
testing@gmail.com	✉	4 Nov 2021	4 Nov 2021	wax9F3lUasV0U78ycN5x1mR6EoF3

Rows per page: 50   1 - 3 of 3

(Created users stored in firebase)

### **3.2) CRUD and Queries**

The CRUD operations that will take place are in the inventory list. It will allow staff to view the inventory, update the inventory, delete sold items as well as add new ones.

```

var selectedRow = null

function onFormSubmit() {
    if (validate()) {
        var formData = readFormData();
        if (selectedRow == null)
            insertNewRecord(formData);
        else
            updateRecord(formData);
        resetForm();
    }
}

function readFormData() {
    var formData = {};
    formData["manufacturer"] = document.getElementById("manufacturer").value;
    formData["model"] = document.getElementById("model").value;
    formData["year"] = document.getElementById("year").value;
    formData["value"] = document.getElementById("value").value;
    return formData;
}

function insertNewRecord(data) {
    var table = document.getElementById("CarList").getElementsByTagName('tbody')[0];
    var newRow = table.insertRow(table.length);
    cell1 = newRow.insertCell(0);
    cell1.innerHTML = data.manufacturer;
    cell2 = newRow.insertCell(1);
    cell2.innerHTML = data.model;
    cell3 = newRow.insertCell(2);
    cell3.innerHTML = data.year;
    cell4 = newRow.insertCell(3);
    cell4.innerHTML = data.value;
    cell4 = newRow.insertCell(4);
    cell4.innerHTML = `

```

```

function onEdit(td) {
    selectedRow = td.parentElement.parentElement;
    document.getElementById("manufacturer").value = selectedRow.cells[0].innerHTML;
    document.getElementById("model").value = selectedRow.cells[1].innerHTML;
    document.getElementById("year").value = selectedRow.cells[2].innerHTML;
    document.getElementById("value").value = selectedRow.cells[3].innerHTML;
}

function updateRecord(formData) {
    selectedRow.cells[0].innerHTML = formData.manufacturer;
    selectedRow.cells[1].innerHTML = formData.model;
    selectedRow.cells[2].innerHTML = formData.year;
    selectedRow.cells[3].innerHTML = formData.value;
    window.alert("Successfully updated")
}

function onDelete(td) {
    if (confirm('Are you sure to delete this item?')) {
        row = td.parentElement.parentElement;
        document.getElementById("CarList").deleteRow(row.rowIndex);
        resetForm();
        window.alert("Successfully deleted")
    }
}

function validate() {
    isValid = true;
    if (document.getElementById("manufacturer").value == "") {
        isValid = false;
        document.getElementById("CarInventoryLog").classList.remove("hide");
    } else {
        isValid = true;
        if (!document.getElementById("CarInventoryLog").classList.contains("hide"))
            document.getElementById("CarInventoryLog").classList.add("hide");
    }
    return isValid;
}

```

(JavaScript used to implement CRUD operations)

Welcome to Craig's World				
Car dealer website				
Current Inventory				
Manufacturer and Model	Chassis Number	Year	Price	
Volkswagen Polo	SDAFHGYSDF348DFH	2015	170000	<a href="#">Edit</a> <a href="#">Delete</a>
Volkswagen Tiguan	SADG54DRH4DFHAG	2019	620000	<a href="#">Edit</a> <a href="#">Delete</a>
Toyota Aygo	SJDHF83892SJHSADG73745	2020	130000	<a href="#">Edit</a> <a href="#">Delete</a>
Land Rover Range Rover Evoque	KYASFDFYA7823HSDG	2018	860000	<a href="#">Edit</a> <a href="#">Delete</a>

Items added

#### **4) Final Review**

I personally found this assignment to be very challenging, I had to do in depth research to try and solve the questions. I was not able to solve some of the issue and due to this some of the assignment functions do not work properly

**(End of assignment)**