The Report

Group 2- Tom, Blessing, Sean, Sam

May 2025

---

1 Introduction

1.1 Contributions

2 Design

We have decided to use a black and red theme for our IBAY website. By using a dark theme, it allows IBAY to stand out from other websites. Our website is still mainly light coloured but the dark branding should pop out at potential customers. Below is our favicon to help users know which tab is for the IBAY website.

Figure 1: Favicon

Our logo also fits this colour scheme. It is a very simplistic logo which makes our website more memorable for customers.

Figure 2: Logo

3 Login Page

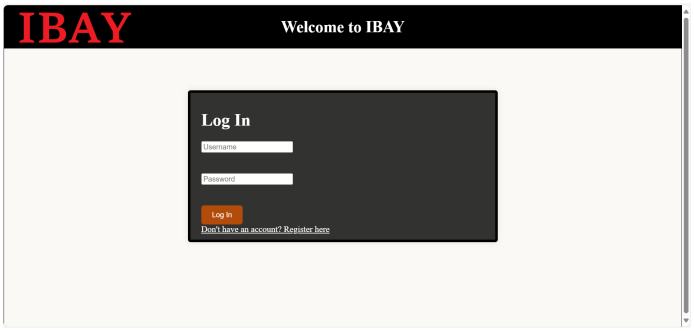

Figure 3: Login Page

On the login page, we have the logo placed in the top left with a 'Welcome to IBAY' message in the middle. In the centre of the page, we have a 'Log In' form for the user to fill out and a hyperlink at the bottom of that to bring new customers to our register page to create a new account. The login page is the first page which the user will encounter. Each component of this page is styled by using the 'loginPage.css' file. Here is an example of some CSS styling used for the login page.

```
/*Styling for header*/
.header {
    position: absolute;
    top: -20px;
    width: 100%;
    height: 80px;
    background-color: #000;
    color: #fff;
    text-align: center;
    line-height: 80px;
    font-size: 30px;
    font-weight: bold;
    border-bottom: 5px solid grey;
}
```

Figure 4: CSS example code

Once the user has entered a username and password, these are sent to the 'login.php' file using the post method. This allows the 'login.php' file to compare this information with that in the 'iBayMembers' database. In order to do this, we have added SQL to return the rows which have the specified username and password. This allows us to check if the username and password are correct and that they match. There is a risk here of SQL injection but we have taken measures to prevent it. By using prepared statements, SQL injection should not be possible. Prepared statements use templated SQL statements with parameters that have no value; these are labelled with '?'. The database then performs and stores the statement but doesn't actually execute it. Once the values of the parameters are confirmed, the database then executes the systems with the

parameters taking the values which are now decided. This is useful against SQL injection as the templated statement does not come from the user.

4 Register Page

Figure 5: Register Page

This page is accessed by clicking the link in the Login Page which takes us to the 'signUpPage.html' file. From here, the user can return to the Login Page by clicking the link near the top of the Sign Up box. In this box, there is a form where the user can enter all the information needed to add the new user to the 'iBayMembers' database. There are 2 textboxes for password (the second being for confirmation.) The user is also required to enter their: firstname, surname, email address, address, postcode, phone number, date of birth and gender. We also have buttons at the bottom to allow the user to reset their responses and one for them to sign up once all fields have been filled out. Here is an example of how we made, textboxes, radio buttons and our sign up/ reset buttons.

```
<!-- HTML for textbox for date of birth -->
<label for="dob">Date of Birth:</label>
<input type="date" id="dob" name="dob" value="1990-01-01" required><br><br>

<!-- HTML for radio buttons for gender -->
<label for="gender">Gender:</label>
<input type="radio" name="gender" value="female">Female
<input type="radio" name="gender" value="male" checked>Male
<input type="radio" name="gender" value="other">Other
<br><br>

<!-- HTML for submit and reset buttons -->
<input type="submit" class="btn" value="Sign Up" id="signUpBtn" name ="signUp-Submit" >
<input type="reset" class="btn" value="Reset" id="resetBtn">
```

Figure 6: Buttons example

The HTML file links to the JavaScript file 'signUP.js' where the user's responses are validated against many constraints. In particular, the password is limited so that it must have at least: 8 characters, 1 number, 1 uppercase letter, 1 lowercase letter and one special character. There is one exception to this which is the password 'test' purely for the assessment of our project.

```
//Event listener for password
password.addEventListener("input", function () {
  var passwordVal = document.getElementById("password").value;
  //If password is "test", skip password validation
  if (passwordVal == "test") {
    password.setCustomValidity('');
  } else {
    //Check if password is at least 8 characters
    if (passwordVal.length < 8) {
      password.setCustomValidity("Password must be at least 8 characters");
    //Check if password contains at least one number
    } else if (passwordVal.search(/[0-9]/) < 0) {
      password.setCustomValidity("Password must contain at least one number");
    //Check if password contains at least one uppercase letter
    } else if (passwordVal.search(/[A-Z]/) < 0) {
      password.setCustomValidity("Password must contain at least one uppercase letter");
    //Check if password contains at least one lowercase letter
    } else if (passwordVal.search(/[a-z]/) < 0) {
      password.setCustomValidity("Password must contain at least one lowercase letter");
    //Check if password contains at least one special character
    } else if (passwordVal.search(/[^a-zA-Z0-9]/) < 0) {
      password.setCustomValidity("Password must contain at least one special character");
    } else {
      password.setCustomValidity('');
    }
  }
  password.reportValidity();
})
```

Figure 7: Password Constraints

5 Buyer's Page

6 Seller's Page

This page is accessed by clicking the 'sell' button on the buyers page. As pictured above, our Seller's page has the header clearly letting the User know that they are on the Seller Page. Underneath that, is a box containing a link to the item upload page for sellers and the box also contains the items which the current user is selling. The items the user has for sale have the following information: Item Title, Item Price, Item Category, Item Description, Duration and there are also buttons which allow the user to either edit or delete the item. The HTML file 'ViewItemsPage.html' defines the table where the items the user has for sale are listed. This file then calls the 'seller.js' file which checks the items and their information against suitable constraints such as having the end date to be after the start date (code pictured below) and having the price in an acceptable regex format.

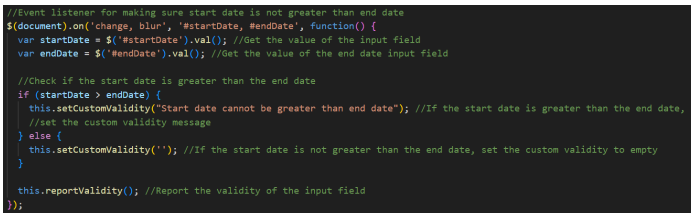

Figure 8: seller.js example code

'ViewItemsPage.html' also calls the 'SellerView.js' file to populate the items table on the page. This file includes a jQuery function to fetch the item data from the 'iBayItems' database. The function contains AJAX code calling the php file 'fetchItems.php' which uses 'connect.php' to connect to the database. Items belonging to the current user are then added to an array and returned to 'SellerView.js' in a JSON format. Next, using AJAX, we loop through the items and append them to the table and then the table is displayed.

7 Database

We have created the 3 databases using the SQL provided in the project specification. In addition to this, we have added 'username','firstname','surname','gender' and 'phone number' to the iBayMembers table. The iBayItems table remains the same but with the iBayImage table, we have changed 'imageSize' to 'imageSizeKB.'

The iBayItems and iBayMembers tables were populated using the example data from the cob295db database and then the iBayMembers table was further populated through the use of AI, in particular chatGPT. The iBayImages table was populated by us with the images used originating from the web.

In our files, connection to the database is gained using one file 'connect.php' Other files then call the 'connect.php' file when it needs to connect to the database.

8 Improvements

8.0.1 Seller's Page

The whole page does not fit so we would adjust the scale accordingly.