2. Design

2.1 Overall System Design

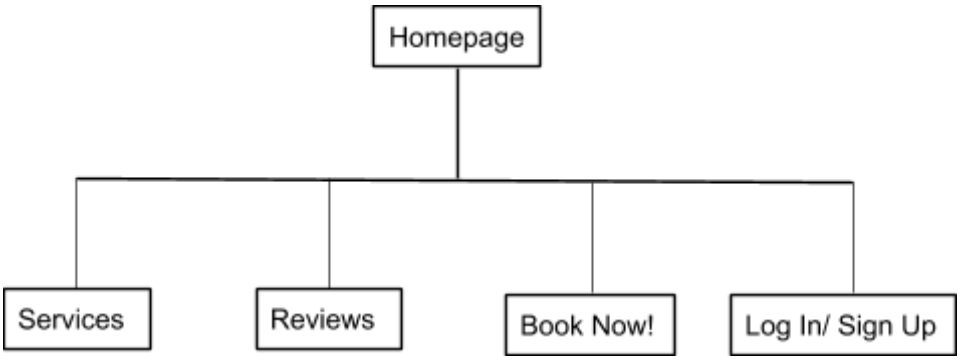
Input:
- Firstname
- Surname
- Email
- Phone Number
- Password
- Date
- Time
- Services
- Reviewer Name
- Star Rating
- Comment

Process:
- Create a customer booking
- Input reviews about the business and the services they provide
- Edit appointments and delete appointments
- Register customer accounts
- Log into customer and barber accounts

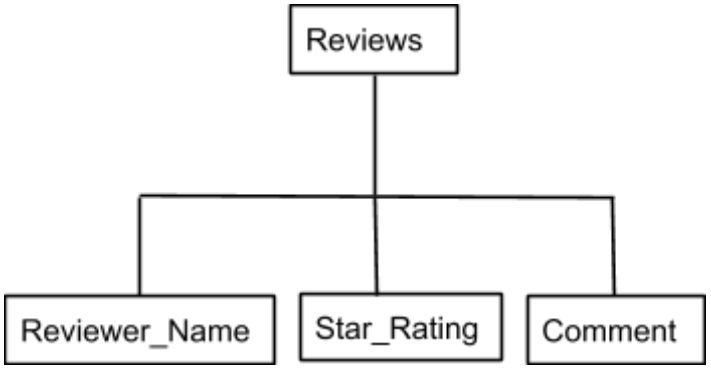
Storage:
- Customers_ID
- FirstName
- Surname
- Email
- PhoneNumber
- Password
- App_ID
- Barber_ID
- Date
- Time
- Services
- Reviewer_Name
- Star_Rating
- Comment

Outputs:
- Display services available
- Display reviews
- Display appointments
- Display customer details

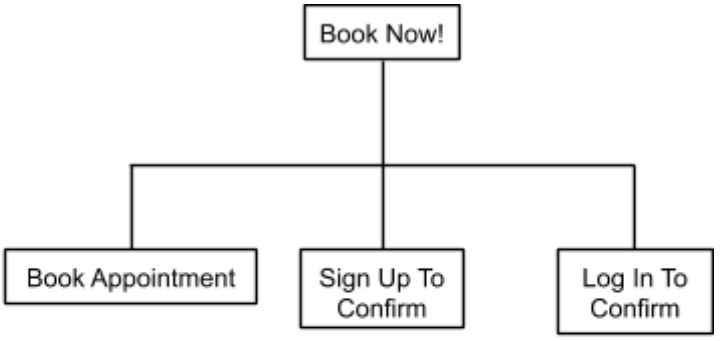
2.2 Description of Modular Structure of System

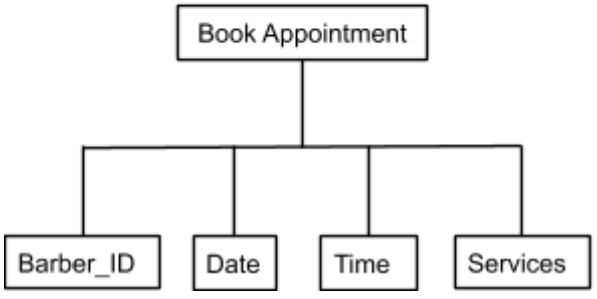


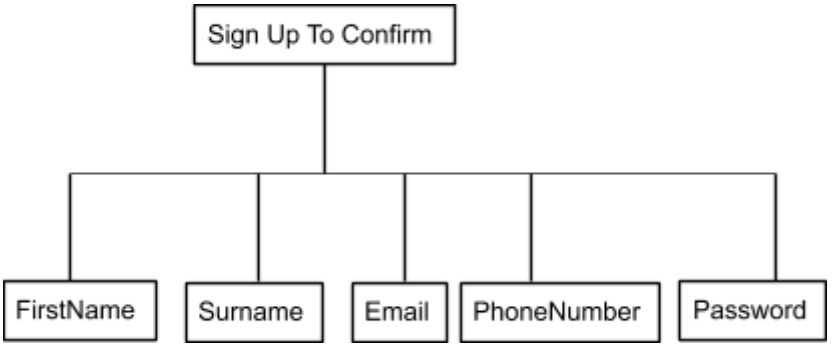
This is the navigation bar that will be accessible to the users when they first visit the homepage.

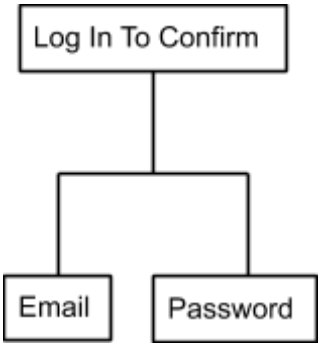


When the user comes to this web page to submit their review, they will first be asked to input their name, their star rating out of 5 and a comment if they would like to share their experience.

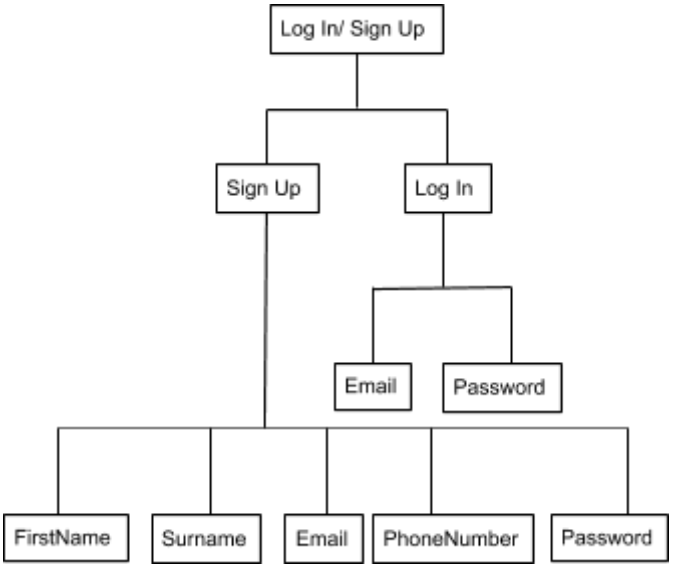


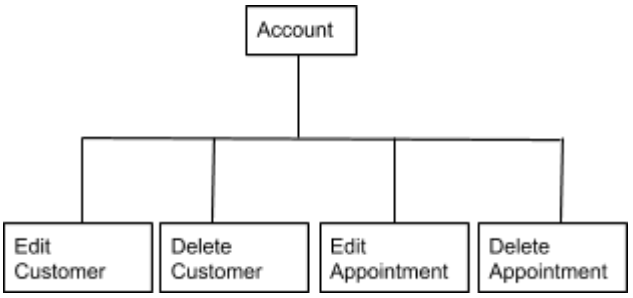






The ‘Book Now!’ page is where customers will book their appointments. Here they will input their appointment details and either confirm the booking by signing up or logging in if they are an existing customer.





The Log In/Sign Up page allows users to login to their accounts with their email and password and also allows new users to register an account with the following inputs. From here, users will be redirected to their ‘account’ page where they can edit their customer details as well as delete their account and also allows them to edit and delete the appointments they have booked.

2.3 Definition of data requirements (Design Data Dictionary)

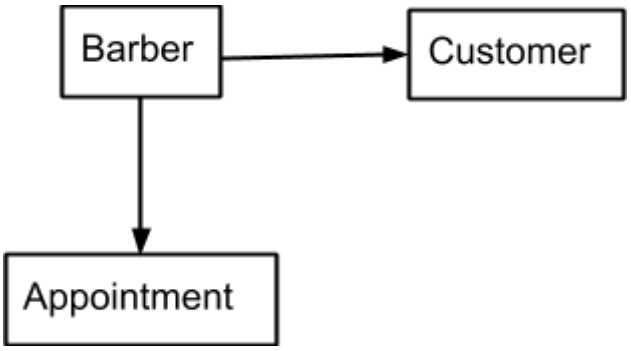
Table Name: Appointments						
Field Name	**Sample Data**	**Data Type**	**Size**	**Validation**	**Default Value**	**Key Field (PK/FK)**
App_ID	1	INT	11	Presence Check	0	PK
Customers_ID	1	INT	11	Presence Check	0	FK
Barber_ID	1	INT	11	Presence Check	0	FK
Date	23.4.20	VARCHAR	50	Presence Check	None	None
Time	11:00 AM	VARCHAR	50	Presence Check	None	None
Services	All	VARCHAR	50	Presence Check	None	None

Table Name: Customers						
Field Name	**Sample Data**	**Data Type**	**Size**	**Validation**	**Default Value**	**Key Field (PK/FK)**
Customers_ID	1	INT	11	Presence Check	0	PK
FirstName	Bob	VARCHAR	50	Presence Check	None	None
Surname	Smith	VARCHAR	50	Presence Check	None	None
Email	Bob@email.com	VARCHAR	50	Presence Check	None	None
Phone Number	09876543211	VARCHAR	50	Presence Check	None	None
Password	Password1*	VARCHAR	50	Presence Check	None	None

Table Name: Barber						
Field Name	**Sample Data**	**Data Type**	**Size**	**Validation**	**Default Value**	**Key Field (PK/FK)**
Barber_ID	1	INT	11	Presence Check	0	PK
FirstName	Bob	VARCHAR	50	Presence Check	None	None
Surname	Smith	VARCHAR	50	Presence Check	None	None
Email	Bob@email.com	VARCHAR	50	Presence Check	None	None
Password	Password1*	VARCHAR	50	Presence Check	None	None

Table Name: Reviews						
Field Name	**Sample Data**	**Data Type**	**Size**	**Validation**	**Default Value**	**Key Field (PK/FK)**
Reviewer_Name	Bob	VARCHAR	50	Presence Check	None	None
Star_Rating	5	VARCHAR	50	Presence Check	None	None
Comment	Very Good	VARCHAR	50	Presence Check	None	None

2.4 Database Design (Normalised ER)



Appointment (AppointmentID, **CustomerID**, **StaffID,** AppDate, AppTime,Services)
Customer (CustomerID, FirstName, Surname, MobileNumber,Email,Password)
Staff (StaffID, FirstName, Surname, Email, Password)

One Barber can have many customers. One customer can have one barber.
One Barber can have many appointments. One appointment can have one barber.

2.5 File organisation and Processing

Table Name: Appointments			
Field Name	**Maximum Size of Field (Bytes)**	**Data Type**	**Size**
App_ID	4	INT	11
Customers_ID	4	INT	11
Barber_ID	4	INT	11
Date	52	VARCHAR	50
Time	52	VARCHAR	50
Services	52	VARCHAR	50

Table Name: Reviews			
Field Name	**Maximum Size of Field (Bytes)**	**Data Type**	**Size**
Reviewer_Name	52	VARCHAR	50
Star_Rating	52	VARCHA	50

		R	
Comment	52	VARCHAR	50

Table Name: Customers			
Field Name	**Maximum Size of Field (Bytes)**	**Data Type**	**Size**
Customers_ID	4	INT	11
FirstName	52	VARCHAR	50
Surname	52	VARCHAR	50
Email	52	VARCHAR	50
Phone Number	52	VARCHAR	50
Password	52	VARCHAR	50

Table Name: Barber			
Field Name	**Maximum Size of Field (Bytes)**	**Data Type**	**Size**
Barber_ID	4	INT	11
FirstName	52	VARCHAR	50
Surname	52	VARCHAR	50
Email	52	VARCHAR	50
Password	52	VARCHAR	50

2.7 User Interface Rationale

Every web page of the website will have the same background and navigation bar. The navigation bar will be placed on the top of the web page where users can navigate to other parts of the website. The navbar will overlap the top part of the background. The background chosen is a simple picture which has a table and barbering equipment placed on this table. The picture has been slightly blurred out to create a better background for the website. The main font I have used all around the website is Arial as i think this is a simple yet easy to

read font. The font size will vary around the website and also the font colour will change from black and white throughout the website. This is to make some elements of the website more appealing and stand out compared to other areas of the website.

2.8 UI Sample of Planned Data Capture and Entry Designs (Contains 2.6 Identification of suitable algorithms for data transformation)

1. **Review Page**

This is the review form that customers will use to post their reviews. The ‘Name’ box is where the reviewer will type his name. After that, they will select their ‘Star Rating’ by using the dropdown select menu. This will allow the reviewer to rate his experience out of 5 stars. A comment box is then placed underneath which will allow the reviewer to share the experiences with the barber shop. After these 3 boxes have been filled out, the reviewer will press ‘Submit’ which will send the inputted values to the database and store the review. The box at the bottom containing ‘Name:’, ‘Star Rating:’ and ‘Comment:’ will then display the submitted review and post it onto the website.

Variable Name:	**Description:**
dbconnect	Stores the connection code which allows the website to connect to the database.

query	Stores the query that is used to input the values submitted from the reviewer.

SQL Query and Algorithms used:

Query 1:

```
query = "INSERT INTO reviews (reviewer_name, star_rating, comment)
VALUES ('$reviewer_name', '$star_rating', '$comment')";
```

This query inserts the values the reviewer has inputted into the database.

Query 2:

```
query = mysqli_query($dbconnect, "SELECT * FROM reviews")
or die (mysqli_error($dbconnect));
```

This query displays all the reviews in the review table of the database.

Algorithm 1:

Set Up Connection
Open Connection
Execute Query
IF Database Is Connected and Query is not executed successfully
Error
Else
Go to review page

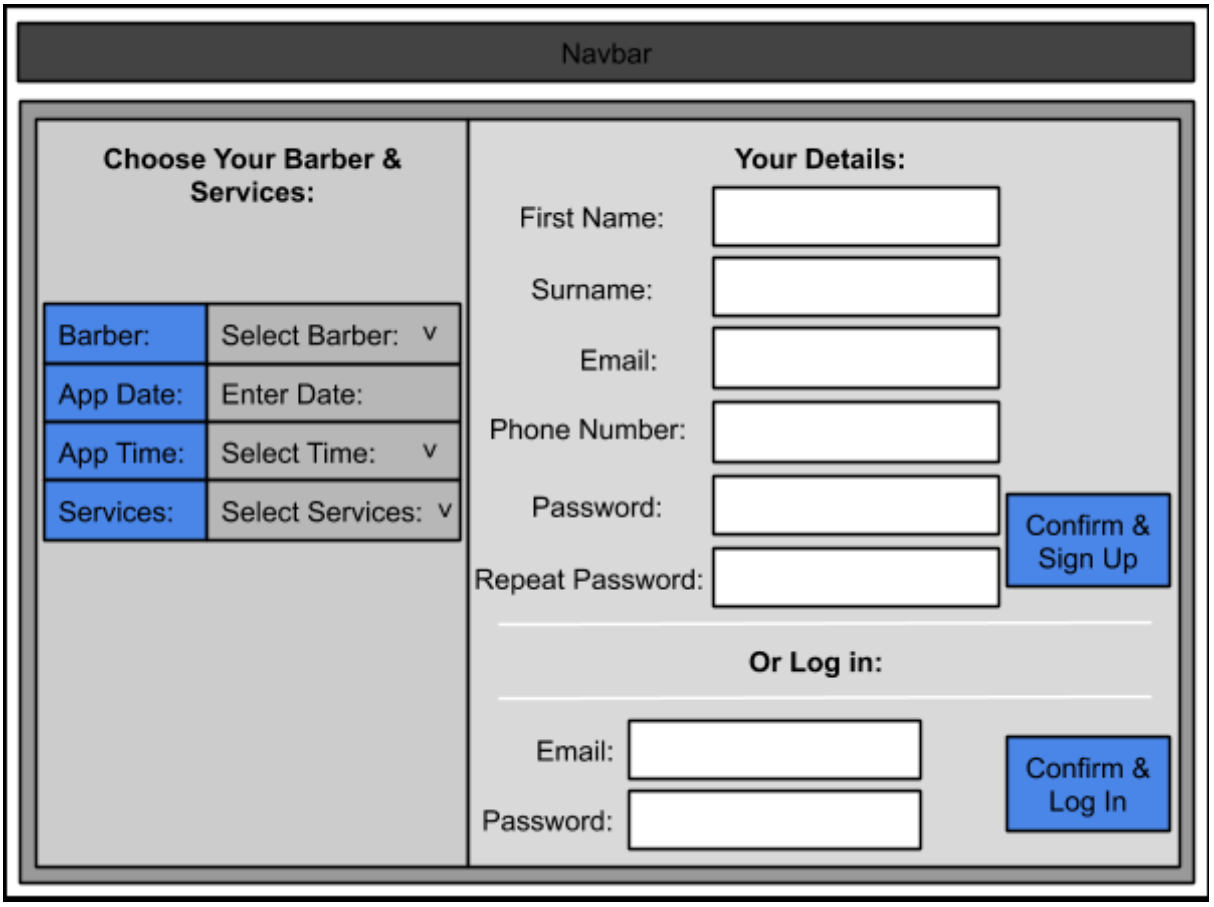
This algorithm first sets up a connection between the website and the database that is being used to store the values. This connection is then opened if the connection is successful. The query is then executed by inserting the values the reviewer has given into the database. If the database is connected but the query is not executed successfully, an error comes up telling the reviewer why the review was not submitted but if the database has connected successfully and the query has been executed successfully then the submit button will head to the review page.

Algorithm 2:

Set Up Connection
Open Connection
Execute Query
IF Database Is Connected and Query is not executed successfully
Die
While
Fetch the values stored
Echo into a table

This algorithm first sets up a connection between the website and the database that is being used to store the values. This connection is then opened if the connection is successful. The query is then executed and if not done successfully, the connection dies. If done successfully, the values stored in the table are fetched and echoed into a designed table.

2. **Book Now Page**



This is the booking form the customers will use to book their appointments. They are first made to fill out the appointment details where they will select their barber, time and service needed from a dropdown select menu and will manually type the date they would like to book the appointment for. After this they must confirm their booking by inputting their details. Users who do not have an account will have to create one and then press ‘Confirm & Sign Up’. Users who already have an account will firstly fill out the appointment form and then log in with their registered account email and password which will confirm their booking.

Variable Name:	**Description:**
conn	Stores the connection code which allows the website to connect to the database.
sql	Stores the sql that is used to input the values submitted from the user.

SQL Statements and Algorithms used:

SQL Statement 1:

```
sql = "INSERT INTO customers (`firstname`,`surname`,`email`,`phonenumber`,`password`)
        VALUES ('$firstname','$surname','$email','$phonenumber','$password')";
```

This SQL statement stores the values inputted by the user to sign up to the website and create an account.

SQL Statement 2:

```
Sql = "SELECT id, email FROM customers WHERE email='$email'";
```

This SQL statement locates the customer by locating the email in the customers table and then SELECTING the id and email.

SQL Statement 3:

```
sql = "INSERT INTO appointments (`customers_id`,`barber_id`,`date`,`time`,`services`)
            VALUES ('$customers_id','$barber_id','$date','$time','$services')";
```

This SQL statement stores the values inputted by the user to create an appointment. This statement is used twice for the same function.

SQL Statement 4:

```
sql = "SELECT id, email FROM customers WHERE email='$email' and
password='$password'";
```

This SQL statement locates a registered user by locating the email and password in the customers table and then SELECTING the id and email.

Algorithm 1:

Set Up Connection
Open Connection
Execute SQL
IF Connection and SQL is true
Execute SQL
Fetch Customers_ID
Execute SQL Query
IF Connection and SQL is true

Confirm booking and sign up
Else
Echo Error
Close Connection

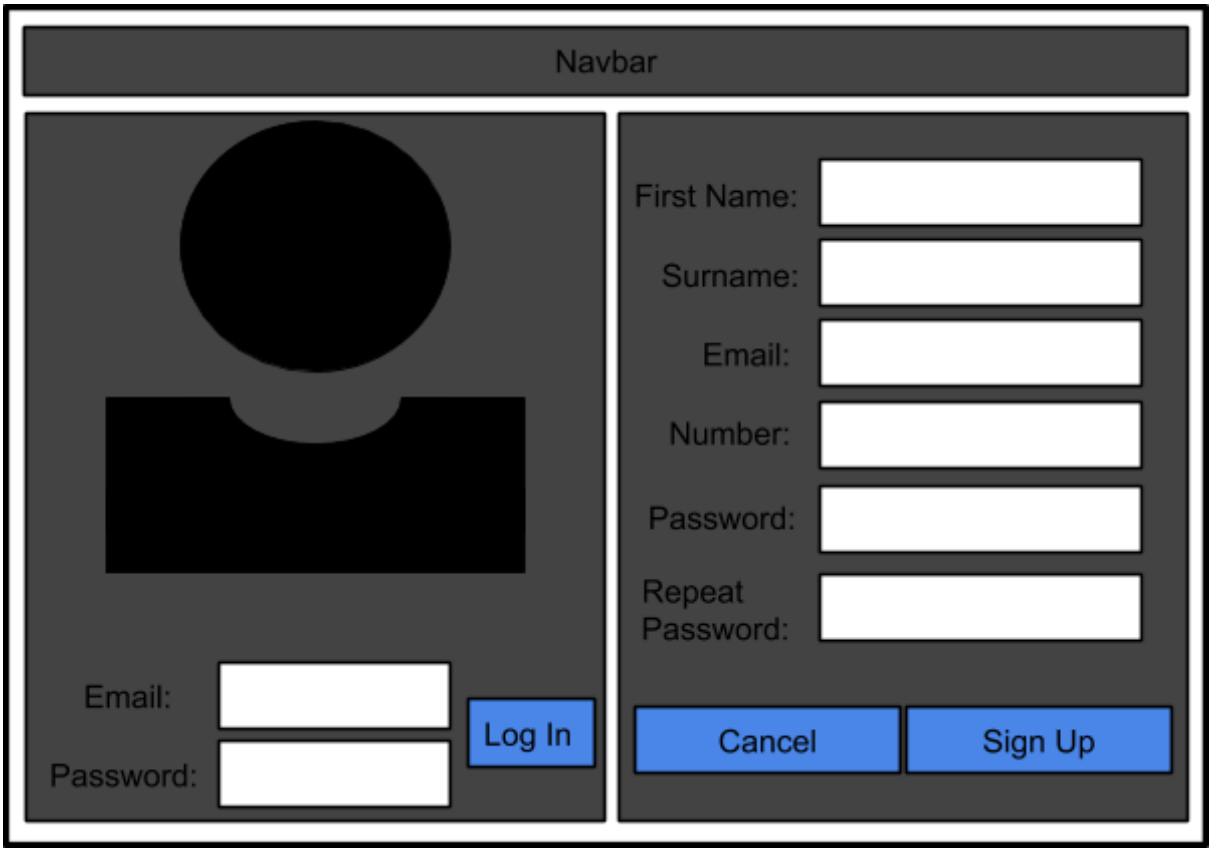
This algorithm first sets up a connection between the website and the database that is being used to store the values. This connection is then opened if the connection is successful. The SQL statement inserts the values the user has inputted to sign up to the website and make an account. The Unique ID of the customer is fetched by using the registered email as a method of locating the customer. The SQL statement inserts the values the user has inputted to book an appointment with the barber shop. The Customer ID is linked to the booking and if successful, the user is sent to their account page. If not successful, an error is echoed.

Algorithm 2:

Set Up Connection
Open Connection
Execute SQL Query
Fetch result
Execute SQL Query
IF Connection and SQL is true
Confirm booking and log in
Else
Echo Error
Close Connection

This algorithm first sets up a connection between the website and the database that is being used to store the values. This connection is then opened if the connection is successful. The email and password the user has inputted into the login form is checked by requesting the email and password from the database table to check if the user exists. The SQL statement is executed to fetch the ID of the specific user by using the verified email and password as a method of locating the customer. The ID is fetched. The SQL statement is executed and inserts the values the user has inputted to book an appointment with the barber shop. The Customer ID is linked to the booking and if successful, the user is sent to their account page. If not successful, an error is echoed.

3. **Log In/ Sign Up Page**



This page allows users to sign up an account with the website or to log into their account. Users will input their values and will either make an account or log in. The result of this will refer them to their account page where they can change their account or booking details as well as delete their bookings and account.

Variable Name:	**Description:**
conn	Stores the connection code which allows the website to connect to the database.
sql	Stores the sql that is used to input the values submitted from the user.

SQL Statements and Algorithms used:

SQL Statement 1:

sql = "SELECT * FROM customers WHERE email='$email' and password='$password'";

This SQL statement locates a registered user by locating the email and password in the customers table and then SELECTING the id and email.

SQL Statement 2:

```
sql = "INSERT INTO customers (`firstname`,`surname`,`email`,`phonenumber`,`password`)
  VALUES ('$firstname','$surname','$email','$phonenumber','$password')";
```

This SQL statement stores the values inputted by the user to sign up to the website and create an account.

Algorithm 1:

Set Up Connection
Open Connection
Execute SQL Query
Fetch row
IF row = 1
Refer to account page
Else
Echo Error
Close Connection

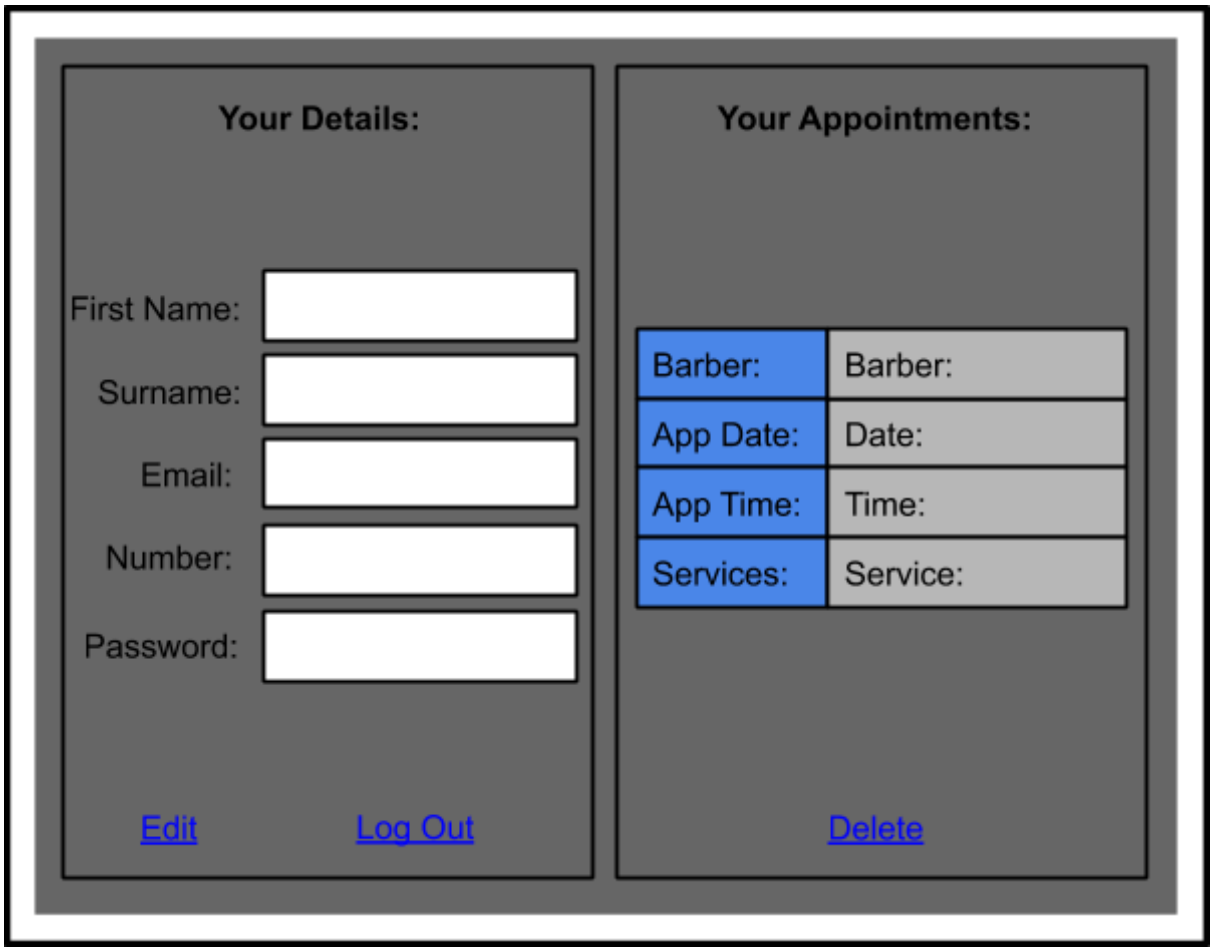
This algorithm first sets up a connection between the website and the database that is being used to store the values. This connection is then opened if the connection is successful. The email and password the user has inputted into the login form is checked by requesting the email and password from the database table to check if the user exists. If the user exists, the user is referred to their account page. If the user does not exist, an error is occured.

Algorithm 2:

Set Up Connection
Open Connection
Execute SQL
IF Connection and SQL is true
Refer to account page
Else
Echo Error
Close Connection

This algorithm first sets up a connection between the website and the database that is being used to store the values. This connection is then opened if the connection is successful. The SQL statement inserts the values the user has inputted to sign up to the website and make an account. If the connection and SQL is true, the user is referred to their account page. If the user does not exist, an error is occured.

4. **Account Page**



This is the account page which will allow users to manage their bookings and account details. Here they can change their account or booking details as well as delete their bookings and account.

Variable Name:	**Description:**
conn	Stores the connection code which allows the website to connect to the database.
sql	Stores the sql that is used to input the values submitted from the user.

SQL Query and Algorithms used:

Query 1:

```
query = mysqli_query($dbconnect, "SELECT * FROM customers")
  or die (mysqli_error($dbconnect));
```

This query displays all the reviews in the customers table of the database.

Query 2:

```
query = mysqli_query($dbconnect, "SELECT * FROM appointments")
  or die (mysqli_error($dbconnect));
```

This query displays all the reviews in the appointments table of the database.

Algorithm 1:

Set Up Connection
Open Connection
Execute Query
IF Database Is Connected and Query is not executed successfully
Die
While
Fetch the values stored
Echo into a table

This algorithm first sets up a connection between the website and the database that is being used to store the values. This connection is then opened if the connection is successful. The query is then executed and if not done successfully, the connection dies. If done successfully, the values stored in the table are fetched and echoed into a designed table.

Algorithm 2:

Set Up Connection
Open Connection
Execute Query
IF Database Is Connected and Query is not executed successfully
Die
While
Fetch the values stored
Echo into a table

This algorithm first sets up a connection between the website and the database that is being used to store the values. This connection is then opened if the connection is successful. The query is then executed and if not done successfully, the connection dies. If done successfully, the values stored in the table are fetched and echoed into a designed table.