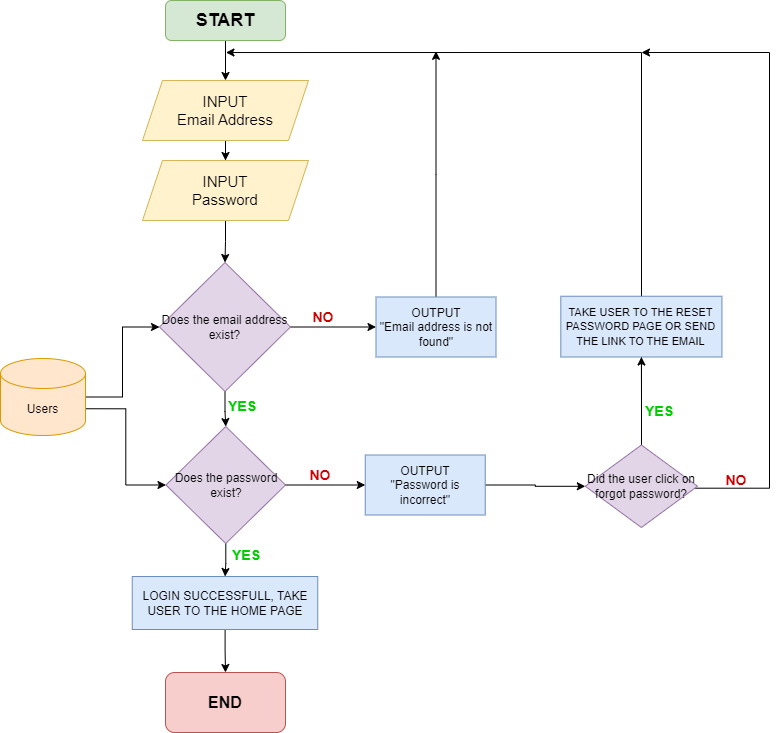
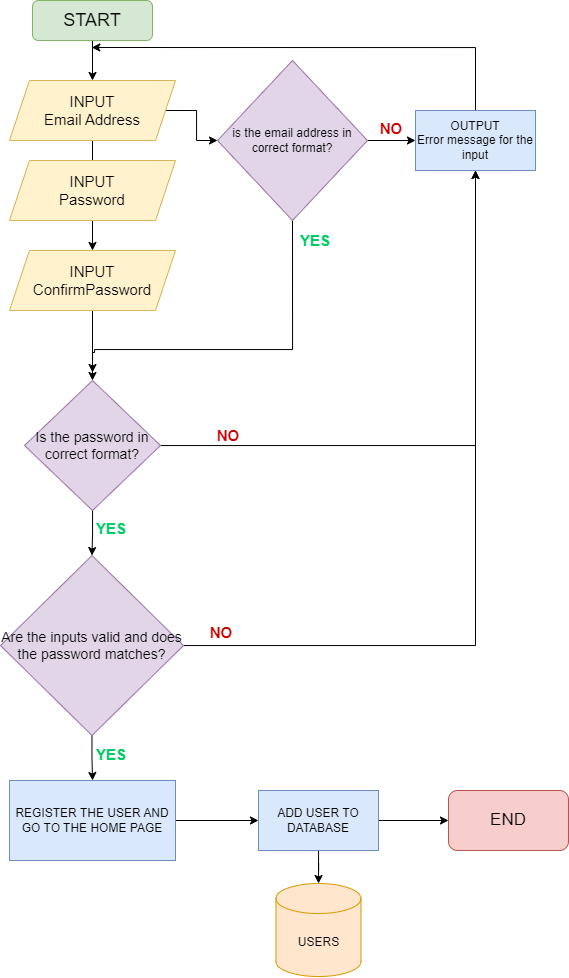
## Flowcharts

Login:



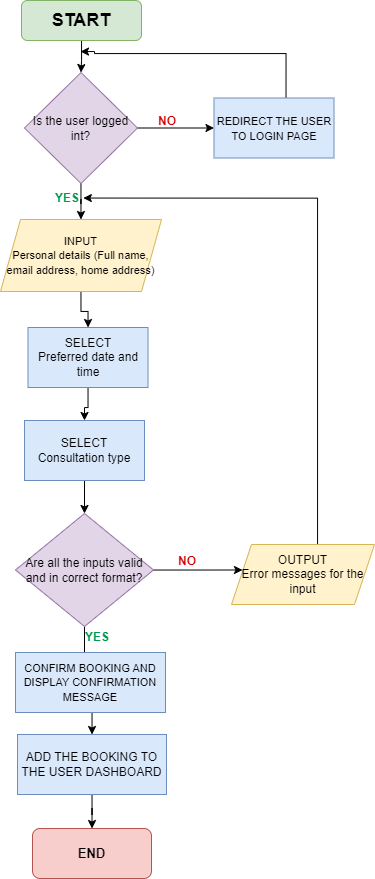
In the login process the user will need to enter their email address and password. The system will then first check if the email exists in the database, if the email is not found it will display an error message. If the email is found, the system then checks if the passoword matches with the email stored in the database. If it is not correct or not found, an error message will be displayed. If the user clicks on “forgot password” they will be taken to the reset password page or an email will be sent to reset it. If both email and password are wrong the user will have to register the account or re-enter the correct credentials. If both email and password does appeard to be correct, the user will be successfully logged in and directed to the home page.

**Register:**



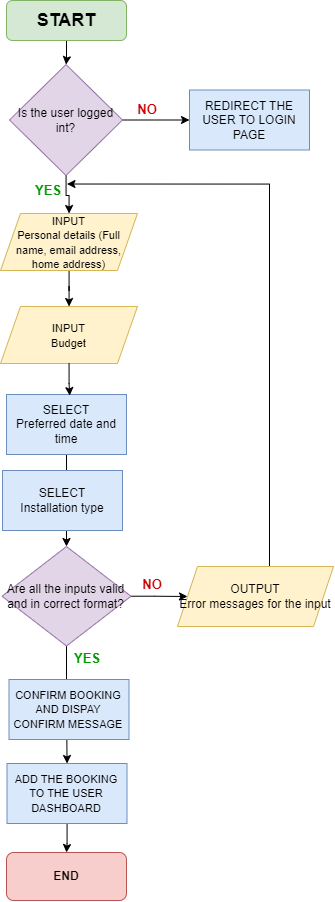
When creating an account the user will need to enter their email address, password and then confirm password (to check if it is a human). The system checks if the email and passowrd are valid and in correct reuired format. If either is incorrect an error message will be displayed and the user will have to enter correct detials. If everything is correct, the account will be created and users details will be saved in the database for future logins and the user will be taken to the home page.

**Booking consultation:**



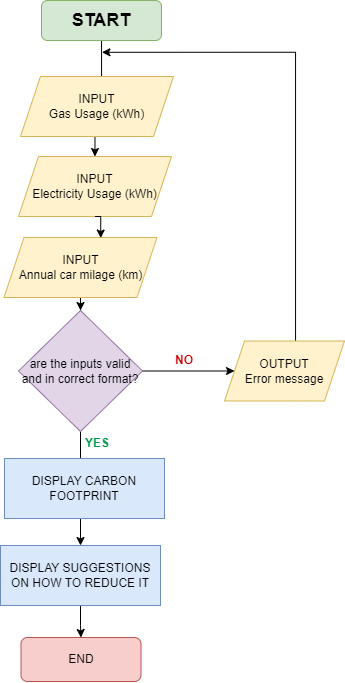
When a user wants to book a consultation, they will go to the booking consultation page. If the user is not logged in, they will first need to log in (so they can save their booking in the dashboard). If the user is logged in, they will need to fill in their personal details such as name, home address, email, phone number etc. After that, they will need to select their preferred date, time and consultation type. The system will then check if the inputs are in the correct format and valid. If any input is invalid or if any required filed is missing an error message will be displayed. If all the inputs are correct the booking will be confirmed and a confirmation message will appear. The booking will then be added to the user’s dashboard where they can track it or cancel it.

**Booking installation:**



When a user wants to book a installation, they will go to the booking installation page. If the user is not logged in, they will first need to log in (so they can save their booking in the dashboard). If the user is logged in, they will need to fill in their personal details such as name, home address, email, phone number etc. After that, they will need to select their preferred date, time and installation type and add their budget. The system will then check if the inputs are in the correct format and valid. If any input is invalid or if any required filed is missing an error message will be displayed. If all the inputs are correct the booking will be confirmed and a confirmation message will appear. The booking will then be added to the user’s dashboard where they can track it or cancel it.

**Carbon footprint calculator:**



When a user wants to calculate their carbon footprint they will need to enter their gar usage, electrecity usage and annual car milage. The system checks if the inputs are valid and if all required fields are filled. If everything is correct it will calculate the user’s carbon footprint and display the results with tips and suggestions on how to reduce their footprint and the postivie impact they can make on the planet. If any input is missing or invalid an error message will be displayed.

## Database design

**Introduction to ASP.NET Core MVC and System Architecture**

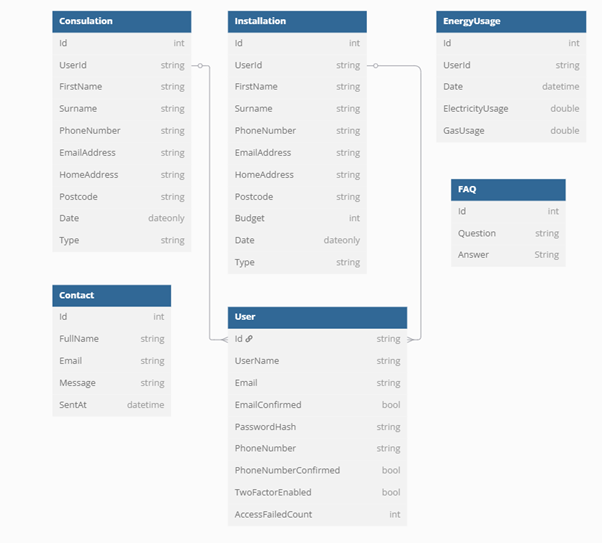
ASP.NET is a framework used to build web applications. It uses MVC structure that divides the application into 3 parts: Model, View and Controller.

**Model:** represents the data and deals with logic**.**

**View:** is a frontend for the users to see the data and interact with it

**Controller:** is the brain behind all the requests or logic and connects the model with the view.

In this system models are used to store data and they help the system to fetch, store and manage information for users.



## Data dictionary

Consultation Booking

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Filed Name | Data type | Validation | Required | Primary/Foreign key |
| Id | int |  | Yes | Primary key |
| UserId | String | Valid user reference | Yes | Foreign key |
| FirstName | String | Max length:100 | Yes |  |
| Surname | String | Max length: 100 | Yes |  |
| PhoneNumber | String | Valid phone format | Yes |  |
| Email Address | String | Valid email format | Yes |  |
| HomeAddress | String | Max length: 255 | Yes |  |
| Postcode | String | Valid postcode format | Yes |  |
| Date | DateTime | Must be future date | Yes |  |
| Type | string | Max length:50 | Yes |  |

Installation booking

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Filed Name | Data type | Validation | Required | Primary/Foreign key |
| Id | int |  | Yes | Primary key |
| UserId | String | Valid user reference | Yes | Foreign key |
| FirstName | String | Max length:100 | Yes |  |
| Surname | String | Max length: 100 | Yes |  |
| PhoneNumber | String | Valid phone format | Yes |  |
| Email Address | String | Valid email format | Yes |  |
| HomeAddress | String | Max length: 255 | Yes |  |
| Postcode | String | Valid postcode format | Yes |  |
| Budget | Int | Budget value >=0 | Yes |  |
| Date | DateTime | Must be future date | Yes |  |
| Type | string | Max length:50 | Yes |  |

Energy Usage table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Filed Name | Data type | Validation | Required | Primary/Foreign key |
| Id | int |  | Yes | Primary key |
| UserId | String | Valid user reference | Yes | Foreign key |
| Date | Datetime | Must be a valid date and time | Yes |  |
| ElectricityUsage | Double | Value >=0 | Yes |  |
| GasUsage | Double | Value >=0 | Yes |  |

FAQ Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Filed Name | Data type | Validation | Required | Primary/Foreign key |
| Id | int |  | Yes | Primary key |
| Question | String | Max length: 255 | Yes |  |
| Answer | String | Max length: 255 | Yes |  |

Contact Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Filed Name | Data type | Validation | Required | Primary/Foreign key |
| Id | int |  | Yes | Primary key |
| FullName | String | Max length: 255 | Yes | Foreign key |
| Email | String | Must be a valid email format | Yes |  |
| Message | String | Max length: 2000 | Yes |  |
| SentAt | Datetime | Must be a valid datetime | Yes |  |

User Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Filed Name | Data type | Validation | Required | Primary/Foreign key |
| Id | int |  | Yes | Primary key |
| Username | string | Max Length:100 | Yes |  |
| Email | String | Valid email format | Yes |  |
| EmailConfirmed | Bool | True or false | Yes |  |
| Password Hash | String | Must be hashed | Yes |  |
| PhoneNumber | String | Valid phone format | Yes |  |
| PhoneNumberConfirmed | Bool | True or false | Yes |  |
| TwoFactorEnabled | Bool | True or false | Yes |  |
| AccessFailedCount | int | >= 0 | Yes |  |

## Test strategy

|  |  |  |  |
| --- | --- | --- | --- |
| Date of test | Component to be tested | Type of test to be carried out | Prerequisites and dependencies |
|  | Home Page | -White box testing  -Unit testing  -Integration testing  -Black box testing | No date inputs needed  Must have completed the CSS and layout design for the home page and functionalities. Testing button functionality, home page unit and its integration within the website. Making sure that the homepage components such as buttons links and dropdown menus interact properly with other sites elements. |
|  | Navigation bar | -Black box testing  -Functional testing  -Integration testing | No data inputs needed  Must have completed the navigation bar functionality and the linked pages must also exist and completed.  Navigate through the navbar to verify correct page loads when clicked on the link. For example, clicking on FAQ should load FAQ page. |
|  | User login system | -White box testing  -Black box testing  -Functionality testing  -Security testing  -Integration testing | Test data is needed.  Navbar must be created with link or button to access this page  User account must be registered  User login page must be created and the migration and update-database commands must have been ran and completed. The controller logic must have been also added.  Testing logic with correct and incorrect credentials.  Example test data:  **-Correct:**  Email=” username@example.com”  Password= “Password123@”  -**Erroneous**:  Email = “username.com”  Password=”1234”  -**Extreme**:  Email= “user@example.com”  Password = “User1@”  Ensures login authentication connects to the database, validates credentials, and handles incorrect login attempts securely. |
|  | User register system |  | Test data is needed.  Navbar must be created with link or button to access this page  User register page must be created and the migration and update-database commands must have been running and completed. The controller logic must have been also added.  Testing logic with correct and incorrect credentials.  Example test data:  **-Correct:**  Email=” username@example.com”  Password= “Password123@”  -**Erroneous**:  Email = “username.com”  Password=”1234”  -**Extreme**:  Email= “user@example.com”  Password = “User1@”  Ensures registration authentication connects to the database and validates credentials |
|  | About us page | -Black box testing  -Unit testing | No test data needed  Navbar must be created with link or button to access this page  About us page must be created and exists.  Page content must be completed with correct information and proper layout |
|  | Contact us section on the home page | -Black box testing  -White box testing  -Functional testing: Integration testing, unit testing  -Form validation | Test data is needed  Navbar must be created with link or button to access this page  Contact us section must be created with proper layout and correct input fields.  There must be necessary validation rules  Testing form submission with valid and invalid data  Checking if submitted form data is stored and if confirmation message display correctly |
|  | FAQ’s page | -Black box testing  -Functional testing: integration and unit testing | Test data is needed  Navbar must be created with link or button to access this page  FAQ model must be created  The model must be scaffolded  FAQ page must be created and completed. Making sure questions and answers are displayed correctly.  Contact us section must be created and form submission should be tested with correct and invalid data.  Testing admin ability to create, edit or delete questions and answers and also verifying database updates and UI response. |
|  | Green energy products page | -Black box testing  -Functional testing: content validation | No test data is needed  Navbar must be created with link or button to access this page  Green energy products page must be created with CSS and proper layout. Page content must be available.  Testing readability and quality of images or videos (if shown) |
|  | How to reduce carbon footprint information page | -Black box testing  -Functional testing: content validation | No test data is needed  Navbar must be created with link or button to access this page  This page must be created with CSS and proper layout. Page content must be available.  Verifying is the information is displayed correctly, easy to read and quality of images and videos (if shown) |
|  | Consultation booking page | -Black box testing  -White box testing  -Functional testing: integration and unit testing  -Form validation | Test data is needed  Navbar must be created with link or button to access this page  Booking model must be created  Booking system must be implemented with correct controller logic.  Model must be scaffolded  Booking consultation page must be design and created  Testing the form with correct and invalid inputs. Ensuring the preferred date is for future only.  Checking is submitted data is stored and if the confirmation message is displayed. |
|  | Installation booking system page | -White box testing  -Black box testing  -Functional testing: integration and unit testing  -Form validation | Test data is needed  Navbar must be created with link or button to access this page  Installation model must be created  Installation booking system must be implemented with correct controller logic.  Model must be scaffolded  Installation consultation page must be design and created  Testing the form with correct and invalid inputs. Ensuring the preferred date is for future only. |
|  | Carbon footprint calculator system page | -White box testing  -Black box testing  -Functional testing: unit and integration testing  -Calculation accuracy  -Performance testing | Test data is needed  Navbar must be created with link or button to access this page  Calculator page must be created and designed  Calculation logic must be implemented.  Testing different inputs to make sure correct carbon footprint calculation results are shown  Checking is the correct tips and suggestions are displayed according to the results.  Testing with incorrect and correct input fields for example for gas usage entering words instead of numbers |
|  | Energy usage tracking tool | -Black box testing  -Functional testing: integration and unit testing  -Usability testing  -Performance testing | Test data is needed  Navbar must be created with link or button to access this page  Model must be created and scaffolded  Correct logic must be implemented  Energy usage tracking page must be created and designed  Testing with correct and incorrect details  Testing If the entered data is saved on the index page and testing options to add more energy usage depending on different users |
|  | Navigation links (footer) | -Functional testing  -Integration testing  -Usability testing | No test data is needed  Footer must be created with working links  Testing all the links if thy take the user to the correct pages for example about us will take to about us page |
|  | Results (carbon footprint calculator) | -Function testing  -Integration testing  -Unit testing  -black box and white testing | No test data is needed  Carbon footprint page must be created  Checking if the correct result and trips are loaded |
|  | Accessibility feature  (Text resizing and changing between dark and light mode) | -Black box testing  -Usability testing  -Compliance testing  -Functional testing: unit and integration testing | No test data is required  Navbar must be created with link or button to access this page  Accessibility settings must be implemented.  Accessibly settings must eb available for each page on the website  Testing text resizing and toggle between dark and light mode |
|  | Admin panel | -White box testing  -Black box testing  -Functional testing: unit testing  -Security testing  -Access control | No test data is needed  Admin account must be created and only available to certain admins to give full access  Admin functionality must be implemented  Testing crud operations for users and booking  Managing and deleted bookings  Managing accounts and FAQ questions and answers |
|  | Forgot password (login page) | -Black box testing  -Functional testing  -Validation testing  -Security testing | No test data is needed  Login page needs to be first developed and implemented  Password reset button must be implemented  Testing if the reset forgot password button works and takes us to the correct reset page |
|  | Change password page | -Functional testing  -Unit testing | Test data is needed  Change password page must be created and implemented with correct and working logic  Testing with correct and invalid credentials for example entering the password which is already set |
|  | Booking management dashboard | -White box testing  -Black box testing  -Functional testing  -User management and security testing  -Performance testing | No test data is needed  User accounts must be registered  Bookings must be made to track them  Dashboard page must be created and implemented with correct logic to store bookings that the user makes  This makes sure users can view and track their bookings or cancel them. |
|  | Profile management page | -White box testing  -Usability testing  -Functional testing  -user management and security testing | No test data is needed  User account must be registered ensuring users can update profile information or delete account.  Profile management page must be created and implemented with correct logic |