**Project Report – Login System**

Introduction

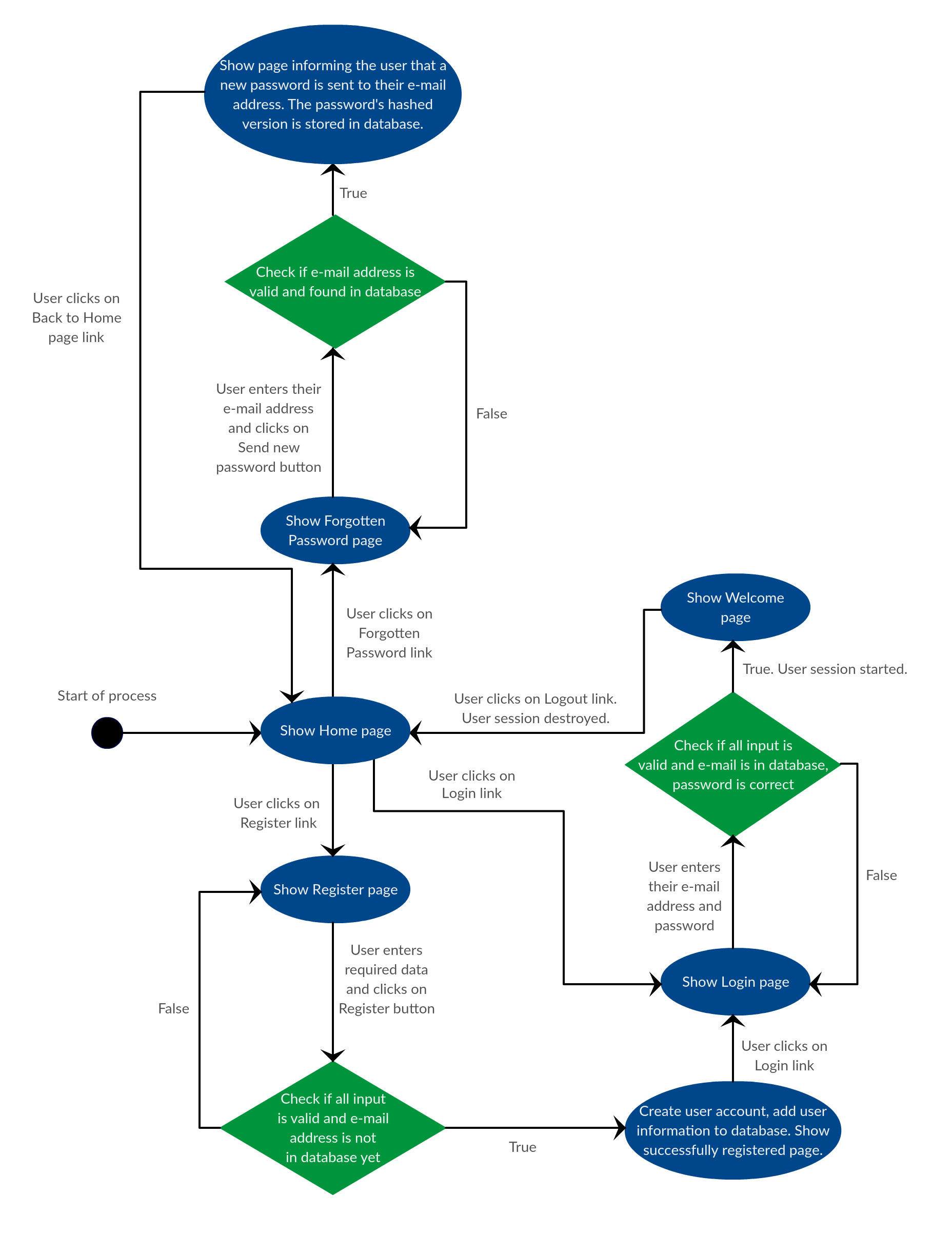
The website created in this project consists of a home page, a register page, a login page, and a forgotten password page. A new user must first register by providing their first and last names, a valid e-mail address, and a valid password. An existing user can log in with their credentials which include their e-mail address (which serves as a username) and their password. The forgotten password page allows a user who forgot their password, to enter their e-mail address, and they are sent a new random password in an e-mail.

Both client-side, using JavaScript, and server-side validation, using PHP, were performed to check that the user input always complies with the requirements. The input in the registration form must comply with several detailed requirements. I decided to check that the first and last names only contain uppercase and lowercase letters using a regular expression in JavaScript. The e-mail address is validated, using JavaScript and a regular expression that matches any valid e-mail address. A PHP script also validates the input, and checks if the e-mail address is already in the database. This check is also done using AJAX on the client side before the form submission. If the e-mail address is not in the database yet, the user can register with it. Otherwise, they must enter a different e-mail address. The requirements for a valid password are that it must have 8 characters or more which can only be letters and numbers, and it must include at least one letter and one number. Other characters are not allowed. The inputs are validated in real time. If an input is valid, the HTML is modified using JavaScript, so the input text box and the bullet points underneath describing input requirements are coloured green as the user finishes typing the input. Otherwise, the text box and the bullet points become red indicating an error. If all text fields contain valid input and the e-mail address is not already in the database, the user can create a new account on click to the Register button. Then, the register.php script inserts a row into the database with the user’s data and the hashed password. If something was invalid when they clicked the button, an alert message notifies the user.

For the login page a similar validation is used both on the client-side and on the server-side to validate the e-mail address and the password that must be entered for logging in. JavaScript is used to validate the e-mail address and the password. AJAX is used to check if the user’s e-mail address is present in the database. On the server side, the login.php script also checks for this and for the validity of the inputs. The user can only log in if their e-mail address was found in the database and was valid, and their password was also valid. After the user has logged in, a session is started, and the welcome page is displayed for them. Then, the user can log out, and the session is destroyed. If a user attempts to access the welcome page without logging in first, the site asks them to log in. User data, which include first name, last name, e-mail address, and the hashed password, are stored in a single table in a MySQL database. The website also has a forgotten password functionality which uses a PHP script to generate a new random password for the user, and send it to them to the e-mail address they provided. The new password is also hashed, and added to the database, so if the user wishes to log in after receiving the new password, they can safely log in with it.

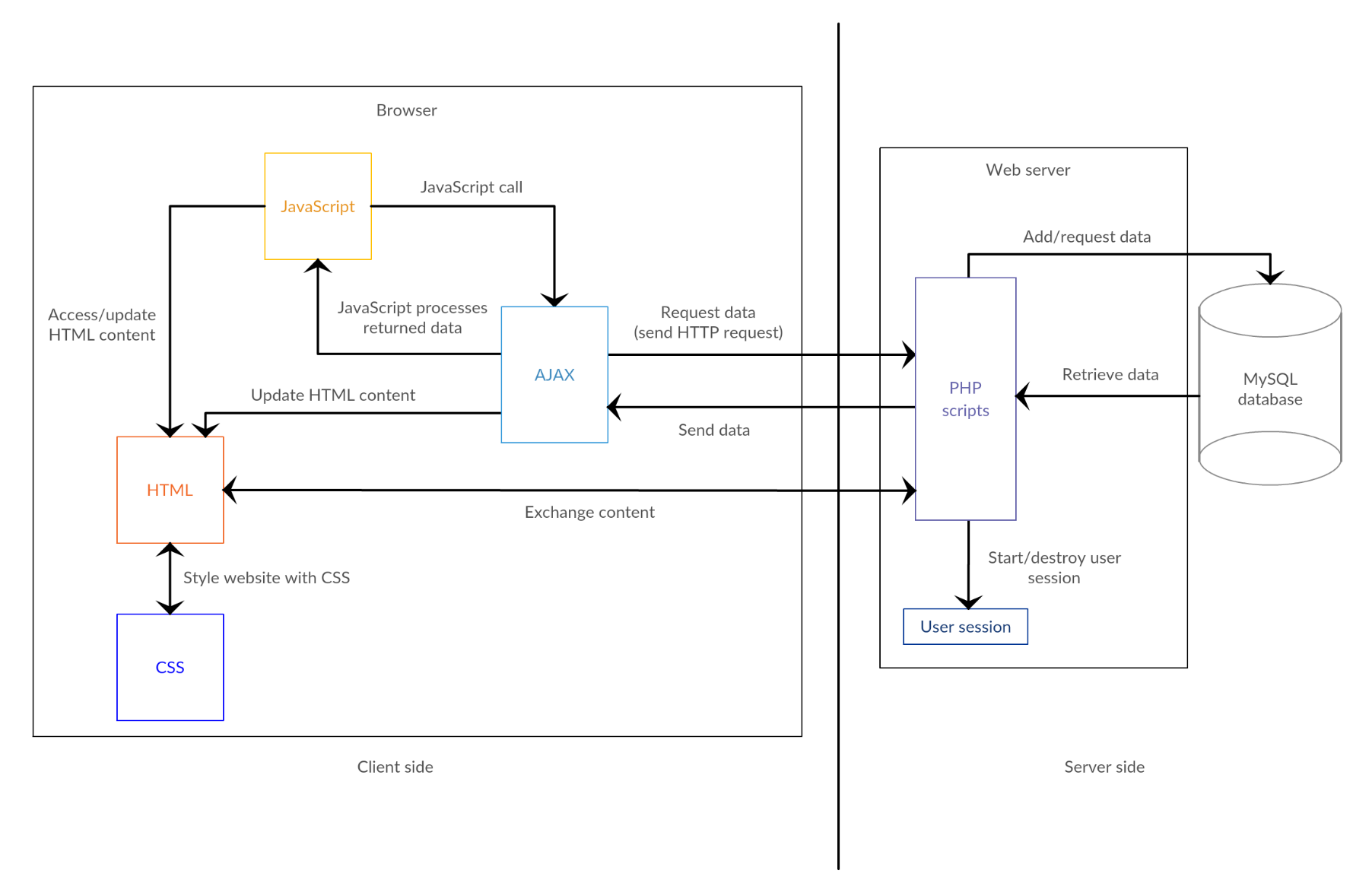
Use Cases

|  |  |
| --- | --- |
| **Use Case 1** | **User Login** |
| Actor | User who wishes to log in on the website |
| Trigger | The user would like to log in on the website using their e-mail address and password which they entered during the registration. |
| Preconditions | The user must have registered before, so they must have a valid e-mail address and a valid password registered on the website and stored in the database. |
| Basic Flow | Initially the user is on the Home page. They click on the Login link which redirects them to the Login page. The user enters their e-mail address, and the website lets them know that it is valid and is registered by making the bullet points green. The user also types in their password which matches all criteria, and the bullet points turn green. If all bullet points are green, the text boxes also become green. The user clicks on the Login button and gets redirected to the Welcome page. There, a welcome message with the user’s name and a Logout link underneath are displayed on the screen. |
| Alternative Flow 1 | After the user clicks on the Login link on the Home page, they must enter their credentials. If they enter an e-mail address which is not found in the database, one of the bullet points and the text box will turn red. Or if the e-mail address is not in the valid format, the bullet point and the text box also become red. If the password does not match any of the required criteria, certain bullet points describing the errors become red as well as the password text box. Or the user can leave a text box empty. If any of these occur, and the user clicks on the Login button, an alert message pops up with the error messages. If the e-mail address and password were valid but do not match an account, the user receives an error message. The user must correct the input before clicking on the Login button. |
| Alternative Flow 2 | After the user has clicked on the Login link on the Home page, they can still go back to the Home page by clicking on the Back to Home page link at the bottom of the page. |
| Alternative Flow 3 | After the user registers, they are asked to log in. They click on the Login link, and are redirected to the Login page. They enter their e-mail address and password, and if both are valid and the e-mail address is found in the database, the text boxes and bullet points turn green. The user clicks on the Login button, and they are redirected to the Welcome page. |
| Alternative Flow 4 | If the user tries to go back to the Welcome page after they have logged out, they must log in again. The welcome.php script shows them a link to the Login page, and they must click on it. On the Login page, after entering the correct user credentials, they are redirected to the Welcome page, and from there they can log out. |
| Postconditions | After filling out the login form correctly with a valid and registered e-mail address, and a valid and registered password, the user is logged in and is redirected to the Welcome page. |
| **Use Case 2** | **Forgotten Password** |
| Actor | A user who forgot their password for their account |
| Trigger | The user forgot their password and would like a new one to be able to log in on the website using their e-mail address and the new password. |
| Preconditions | The user must have registered before, so they must have a valid e-mail address and a valid password registered on the website and stored in the database. The user has tried to log in, but did not manage to remember their password. They must remember their e-mail address used for the registration. |
| Basic Flow | The user has tried to log in on the Login page, but failed to remember their password. So, on the home page they click on the Forgotten password link, and are redirected to the Forgotten password page. Here, the user must enter their registered e-mail address. If the entered e-mail address is valid and is found in the database, the bullet points and the input text box become green. The user clicks on the Send new password button, and a new random password is sent to their e-mail address. They can use this together with their e-mail address to log in on the website. |
| Alternative Flow 1 | After the user has clicked on the Forgotten Password link on the Home page, they can still go back to the Home page by clicking on the Back to Home page link at the bottom of the page. |
| Alternative Flow 2 | If the user clicks on the Forgotten password link on the Home page, they are redirected to the Forgotten password page. If the user has entered their e-mail address and it is found to be invalid, the bullet point describing this error, becomes red. Or, if the e-mail address was not used for registration before, the bullet point describing this error becomes red. If there is an input error, the input text box also becomes red. If the user does not enter anything, the green or red colours are not applied. If any of these occur when the user is on this page and they click on the Send new password button, an alert message pops up stating the errors. No new password is generated and nothing is sent to user’s e-mail address. |
| Postconditions | After entering a valid and registered e-mail address and clicking on the Send new password button, a new random password is generated for the user’s account, the password’s hashed version is stored in the database, and the user receives an e-mail containing the non-hashed version. The user can use this password together with their e-mail address to log in on the website. |

Design: Flow diagram of the system

The flow diagram is a graphical representation of the system’s design. It shows the steps involved in the process of using the website. The starting point is the Home Page.

Architecture: Diagram showing the technical architecture

This diagram shows the technical architecture of the system displaying the communication processes and the flow of data between technologies on the client side and on the server side.