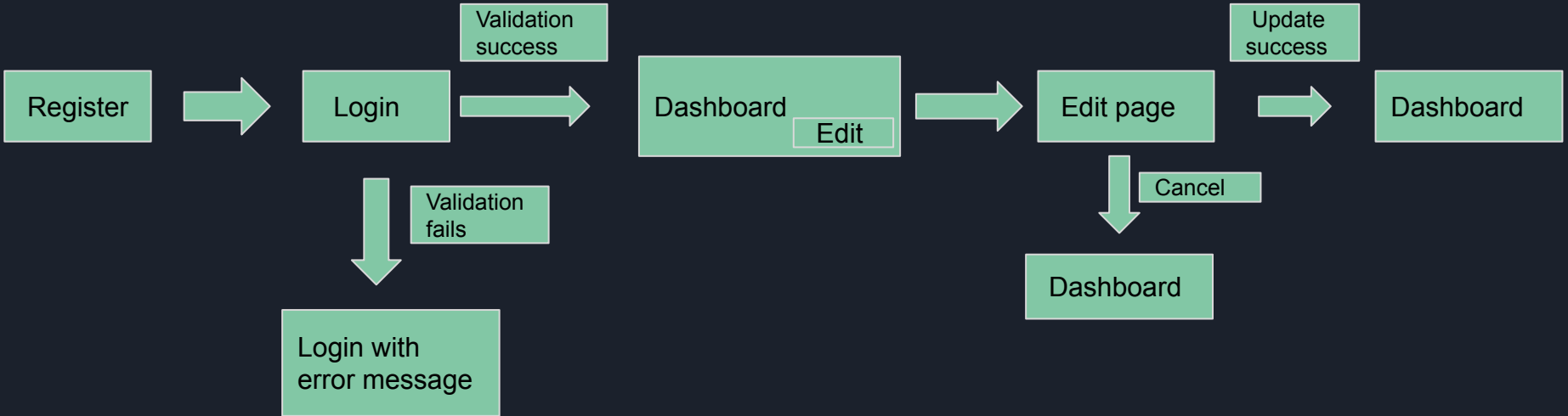
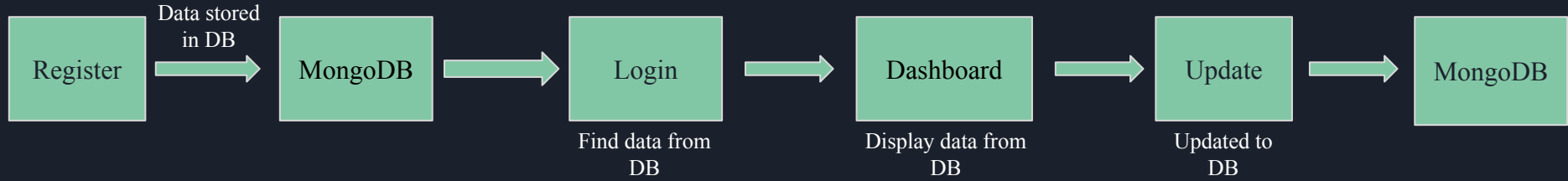


ONBOARDING



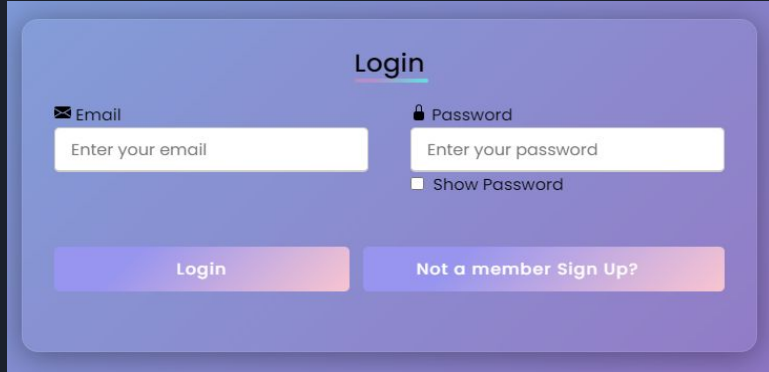
1st Request :



Frequent Request :

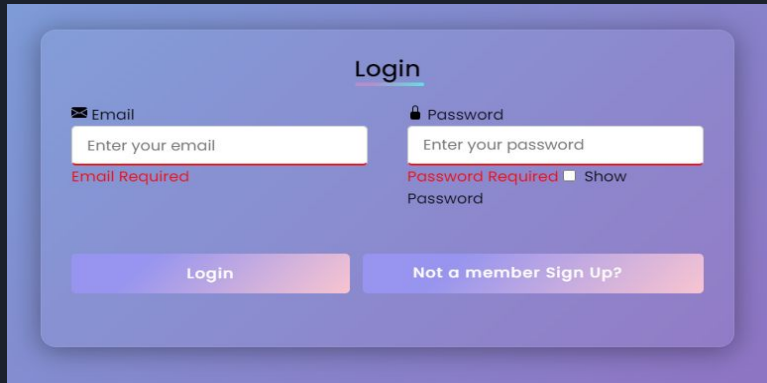


Login

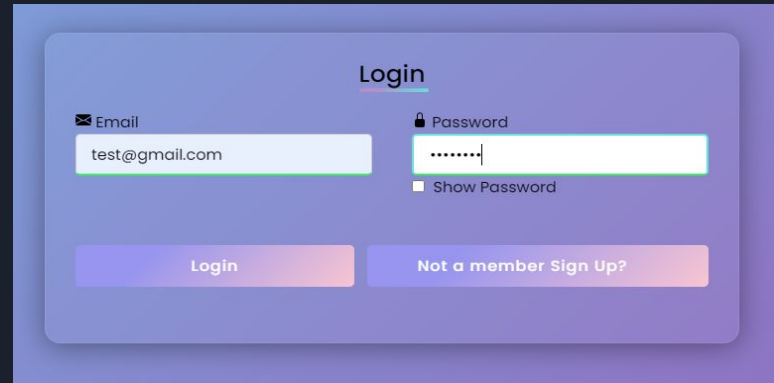


The image shows a login form titled "Login" with a blue header. It contains two input fields: "Email" with a placeholder "Enter your email" and "Password" with a placeholder "Enter your password". A checkbox labeled "Show Password" is next to the password field. Below the fields are two buttons: "Login" and "Not a member Sign Up?".

- ❖ The Login page is constructed using HTML and CSS.
- ❖ Javascript was utilized on the client-side for validation.
- ❖ Following validation, the POST method is used to make an ajax connection to the API.
- ❖ The Model API uses PHP fetch method to check the email and password in the server-side database of MongoDB.
- ❖ PHP receives the data if the login is successful.
- ❖ If the login is successful, PHP retrieves the user's data from the database and saves it in the redis (stored as cache).
- ❖ Ajax redirects the login page to the View profile page.

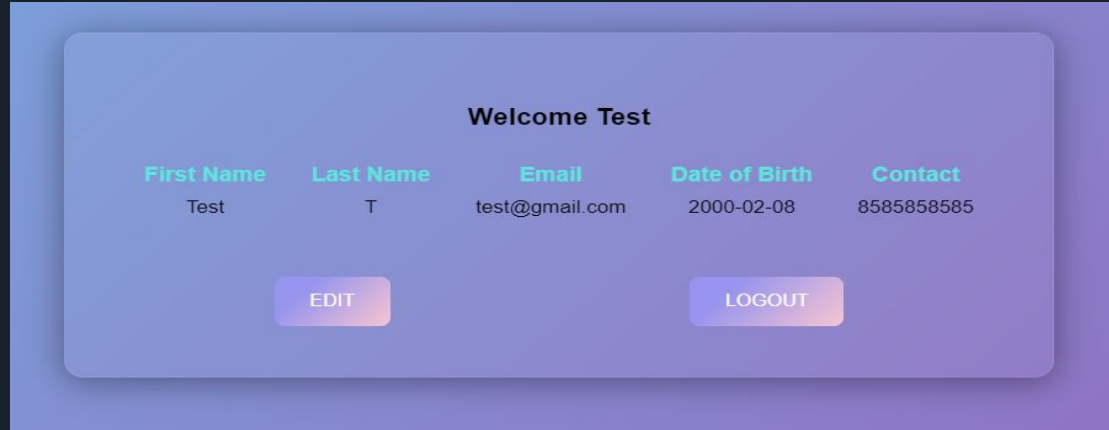


The image shows the login form with validation errors. The "Email" field has a red border and a red error message "Email Required" below it. The "Password" field has a red border and a red error message "Password Required" below it. The "Show Password" checkbox is also visible. The "Login" and "Not a member Sign Up?" buttons are at the bottom.



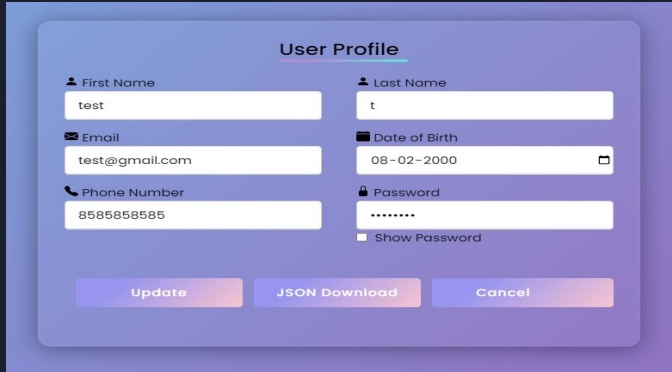
The image shows the login form with user input. The "Email" field contains "test@gmail.com" and the "Password" field contains ".....". The "Show Password" checkbox is visible. The "Login" and "Not a member Sign Up?" buttons are at the bottom.

View Profile



- ❖ HTML and CSS were used to create the see profile page.
- ❖ First time it displays the data from the mongodb, after the first request it displays the data from the redis (stores the cache data).
- ❖ This page displays the read-only data, which has been converted to JSON format via an API call.
- ❖ The edit button on the view profile page allows you to make changes to the user's information except the email field.

Edit Page



The 'User Profile' edit form is a light blue rounded rectangle. It contains two columns of input fields. The left column has fields for 'First Name' (value: test), 'Email' (value: test@gmail.com), and 'Phone Number' (value: 8585858585). The right column has fields for 'Last Name' (value: t), 'Date of Birth' (value: 08-02-2000, with a calendar icon), and 'Password' (value:). Below the password field is a 'Show Password' checkbox. At the bottom are three buttons: 'Update', 'JSON Download', and 'Cancel'.

User Profile

First Name: test

Last Name: t

Email: test@gmail.com

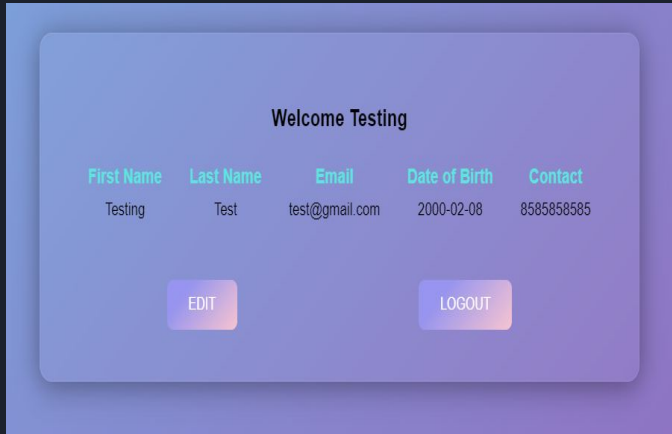
Date of Birth: 08-02-2000

Phone Number: 8585858585

Password:
☐ Show Password

Update JSON Download Cancel

- ❖ HTML and CSS were used to create the edit. Javascript was utilized on the client-side for validation.
- ❖ Data is automatically entered on the edit page with the help of a redis (data stored as cache).
- ❖ When you click the "Update" button once you've finished updating, the revised data is saved in both the server-side database of mongodb and redis (data stored as cache).
- ❖ The "JSON Download" button allows users to download the data in JSON format.



The 'Welcome Testing' profile card is a light blue rounded rectangle. It features a table with user details. Below the table are two buttons: 'EDIT' and 'LOGOUT'.

Welcome Testing

First Name	Last Name	Email	Date of Birth	Contact
Testing	Test	test@gmail.com	2000-02-08	8585858585

EDIT LOGOUT

- ❖ After the update is complete, the website will redirect to the view profile page with the updated data.
- ❖ The updated data will appear on the view profile page.