**Assignment12\_category\_018**

|  |
| --- |
| **📽️** [**Requirement Explaintion Video**](https://drive.google.com/file/d/1YxY-cHwmWiqI4atFS0FUGM6IqL6mUfmg/view?usp=sharing)**-PART-1 📽️** |
| **📽️** [**Requirement Explaintion Video**](https://drive.google.com/file/d/1SWZQHdMWQ0qUir5i5jsWwItpwQxclm2x/view?usp=sharing)**-PART-2 📽️** |

**Objective**

Welcome to your next exciting challenge! Your task is to **develop a Matrimony platform** using the powerful **MERN stack** (MongoDB, Express, React, and Node.js). This online platform will enable users to connect with potential life partners in an intuitive, user-friendly environment. Here’s what this project entails:

**Main Tasks**

**🚨Make the homepage responsive for mobile, tablet, and desktop**

**🚨You can’t use Daisy UI for designing**

**Key Rules:**

* Include a minimum of 20 notable GitHub commits on the client side
* Include a minimum of 12 notable GitHub commits on the server side
* Add a meaningful readme.md file with the name of your website, Admin email, password, and live site URL. Include a minimum of 10 bullet points to feature your website.
* Make it responsive for all devices. You need to make it responsive for mobile, tablet, and desktop views. Make the dashboard responsive as well.
* After reloading the page of a private route, the user should not be redirected to the login page.
* Use the Environment variable to hide the Firebase config keys and Mongodb credentials.
* Don’t use any Lorem ipsum text in your website.
* Show sweet alert/toast/notification for all the crud operations, successful authentication login, and sign-up. Don’t use the default browser alert.
* Implement tanstack query in all the data fetching functionality (For GET method only)

**HomePage:**

* Navbar has a **logo + website name, Home, Biodatas, About Us, Contact Us,** and **Login**. If the user is logged in, the **Dashboard** appears on the navbar.
* If the user clicks on the **Dashboard**, the user is redirected to the dashboard page.
* **Banner/Slider:**  Add a beautiful banner/ slider.  Try to make it eye-catching and impressive.
* Under the banner section, six premium member profile cards must be shown.

**This section** **has the following information:**

* Biodata Id  [1 | 2 | 3]
* Biodata Type [ Male | Female ]
* Profile Image
* Permanent Division name [Dhaka, Chattagram, Rangpur, Barisal, Khulna, Mymensingh, Sylhet]
* Age [23 | 45 | 30]
* Occupation [Student, Job, House wife]
* View Profile

**Note:** When Clicking the view profile button, the user will be redirected to the details page, where he can see all the details of the biodata. This details page must be private, which means if the user is not logged in, the user can not access this page. The user will be automatically navigated to the login page.

**Note:** You must show the cards in ascending or descending based on the age. There will be a dropdown select menu where you can select ascending or descending, You can do client-side or server-side ordering, it's your choice

**How it works section:** Under the premium membership card, Here you will design how this website works. You can do it in your own way. [Visit any matrimony website for ideas.]

**Success Counter Section:** Under the **How it Works** section, you can implement a counter section. Users will see the total biodata of this website how many girls biodata have on the website, how many boys biodata have, and how many marriages have been completed through the website.

**Success Story Section:** Under the **Success** counter section, You can Show all the marriage success stories that the married couple has posted.  You can think of this section as a review section.

This Success story **has the following information :**

* Couple Image or male or female image.
* Marriage Date
* Review Star
* Success Story Text.

**Note:** You must show the success story ascending to descending based on the marriage date.

**Footer:**  Create a beautiful footer.  Try to make it catchy and impressive.

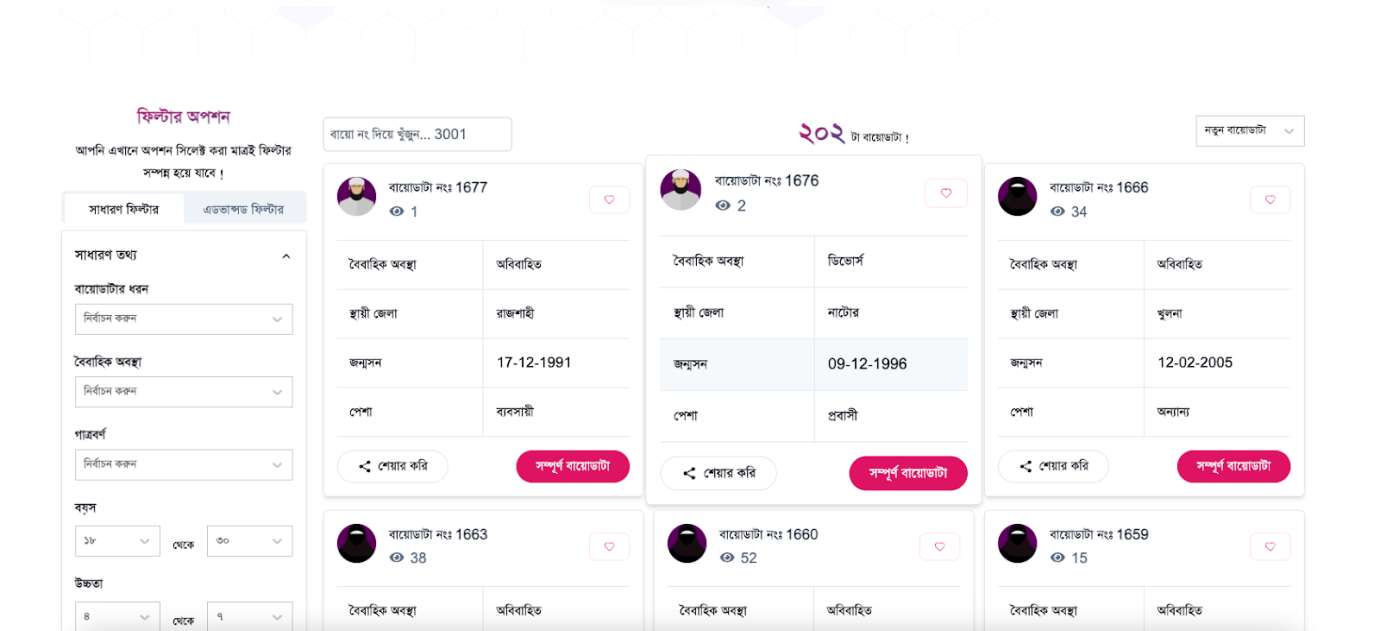
**BiodatasPage:**

This page will be divided into two parts.

1. There will be filter options on the left side.

2.  All Created biodata.

Example Layout :



In the filter section, user can filter by age [Range], biodata type [male|female], filter by division [Dhaka, Chattagra, Rangpur, Barisal, Khulna, Mymensingh, Sylhet]

On the other hand, show all the created biodata. You must show 20 biodatas in this section. Every Biodata **having**

**the following information:**

* Biodata Id [1 | 2 | 3]
* Biodata Type [ Male | Female ]
* Profile Image
* Permanent Division name [Dhaka, Chattagra, Rangpur, Barisal, Khulna, Mymensingh, Sylhet]
* Age
* Occupation
* View Profile

**Note:** When Clicking the view profile page user sees the details biodata information on the details page. This details page must be private. it means if the user is not logged in, the user can not access this page. User will navigate the login page.

**Login & Registration systems**

On the Registration and Login pages, display relevant error messages when necessary.

**Login Page**: When a user clicks on the login button, they will be redirected to the login page having the following:

○ Email/Password

○ Google Sign-in

○ A link that will redirect to the registration page

**Registration Page:** The Registration page will have the Email/Password form having the following fields:

○ Name

○ Email

○ Password

○ Photo URL

Note: Do not enforce the email verification method and forget & reset password method, as it will inconvenience the examiner. If you want, you can add it after receiving the assignment result.

**Biodata Details Page (Private route):**

Show details of a single biodata on this page, and on the bottom show some similar biodata (maximum 3). For example, If you are visiting a male biodata details then show some other male biodata on the bottom

**Biodata details information**

* All the information that a user has filled in to create a biodata should be displayed here **except Contact Information (Premium members can see this)**. But When The user will be a premium member. Users can see the contact information. It means a normal user can't see the contact information. By default, no one can see biodata contact information **except Premium members**. Like Email and phone numbers.

* There is an Add to Favourites button. When the user clicks the button this biodata is added to the favourites biodata collection.

**Note:** See the User Dashboard Requirement for understanding the functionality of My Favourites **Page.**

* There is a **Request Contact Information Button (Only see the normal user. Premium members can directly see contact information without payment)**. When Normal Users click the button they will be redirected to the checkout contact information dynamic page like: checkout/:biodataId. See the checkout page now :

**Checkout Page (Private Route):**

1.  On the checkout page, To request contact information, the user must fill out a form. The form will contain these input fields.

* biodataId (Which biodata contact do you need) **[Must Readonly]**
* Self Email (Who wants to contact information) **[Must Readonly]**
* **Stripe Card Number Input**
* **Submit Button**

**Once the form submission is completed after paying the fee [5 USD], the contact information request will go to the admin for verification and show this request data in the user MyContactRequest Page in the user dashboard. By default, Contact requests will be pending. When the admin approves this request then the user can see the contact information from MyContactRequest Page in the user dashboard.**

**Note:** See the User Dashboard Requirement for understanding the functionality of MyContactRequest **Page.**

**Normal User Dashboard (Private Route):**

**Note: This must be a dashboard layout**

1. When a user clicks on the Dashboard, he/she will be redirected to a page where there will be the following routes:
2. Edit Biodata.
3. View Biodata **(See the biodata info).**
4. My Contact Request.
5. Favourites Biodata.
6. Logout (Button)

**# Edit Biodata:** In this Page, When users login successfully they can create / edit their biodata information. Every Biodata **has the following information:**

* Biodata Type [ Male | Female ] **[Must Select Input]**
* Name
* Profile Image Link or Image input
* Date of birth **[Date Select Input]**
* Height**[Must Select Input]**
* Weight**[Must Select Input]**
* Age
* Occupation **[Must Select Input]**
* Race **[Must Select Input] (Skin color)**
* Fathers name
* Mothers name
* Permanent Division name [Dhaka, Chattagra, Rangpur, Barisal, Khulna, Mymensingh, Sylhet] **[Must Select Input]**
* Present Division name [Dhaka, Chattagra, Rangpur, Barisal, Khulna, Mymensingh, Sylhet] **[Must Select Input]**
* Expected Partner Age
* Expected Partner Height **[Must Select Input]**
* Expected Partner Weight **[Must Select Input]**
* Contact Email (User Email Readonly)
* Mobile Number (Required)

When the user fills in all the information, the user can publish their biodata. There is a Save And Publish Now Button. When the User clicks the Save And Publish Now Button button the biodata will be created successfully or edited successfully.

**Note Important:** In the biodata created time BiodataId will be auto-generated from the server side.

Suppose, There is no biodata in the database then when the user first time creates a biodata then her/his biodata id is 1.

And then when the second biodata is created then her/his biodata id is 2.

Show this biodata id is generated dynamically from the backend.

Hints: Find the last created biodata id and sum 1. Like

Let lastId = 5

NewId = lastId + 1

Set the new id in the database.

# View Biodata: On this Page, When users can see their biodata information. Every Biodata **having the following information:**

* Biodata Type
* Name
* Profile Image
* Date of birth
* Height
* Weight
* Age
* Occupation
* Race **(Skin color)**
* Fathers name
* Mothers name
* Permanent Division name
* Present Division name
* Expected Partner Age
* Expected Partner Height
* Expected Partner Weight
* Contact Email
* Mobile Number

In the view Biodata Page There is a  Make biodata to **premium** Button.

When the User clicks the button, shows a modal and asks if are you sure to make you premium. When the user clicks Yes Then the biodata is sent to the admin for premium Approval. When the Admin Approves the biodata to premium then Make the biodata premium. See the admin dashboard premium approval page.

**# My Contact Request:**  Show all contact request information. When the user successfully pays for biodata contact and admin approves it from the admin dashboard then the user can see the information here in the table layout. You can use any table except Daisy UI.

Show the information in the table:

1. Name
2. Biodata Id
3. Status [Approve | Pending]
4. Mobile No [If the status is approved]
5. Email [If the status is approved]
6. Delete Button

**# My Favourites Biodata:**  Show all The Favourites Biodata information. Users can see the information here in the table layout. You can use any table except Daisy UI.

Show the information in the table :

1. Name
2. Biodata Id
3. Permanent Address
4. Occupation
5. Delete Button

**Admin Dashboard (Private Route):**

**Note: This must be a dashboard layout**

1. When an admin clicks on the Dashboard, he/she will be redirected to a page where there will be the following routes:

1. Admin Dashboard
2. Manage Users
3. Approved Premium
4. Approved Contact Request
5. Logout (Button)

**# Admin Dashboard:** /dashboard

* Show the total biodata count.
* Show the male biodata count.
* Show the female biodata count.
* Show the premium biodata count
* Show the total revenue for purchasing contact information.

**# Manage Users  :** /dashboard/manage

Show all the users in a tabular form where each row will have:

* User name
* User email
* Make admin
* Make premium

The admin can make a user admin by clicking on the **Make Admin** button and also premium users by clicking the Make Premium (Those who have requested to make premium their information will show here)

Implement a server-side search functionality to find a specific user (via username).

# ApprovedPremium : /dashboard/approvedPremium

* Show all the premium approval request in a tabular form where each row will have:

\* Name

\* Email

\* Biodata Id

\* Make Premium

 By clicking the Make premium button user biodata will be premium .

**# Approved contact request  :** /dashboard/approvedContactRequest

* Show all the all contact request in a tabular form where each row will have:

\* Name

\* Email

\* Biodata Id

\* Approved contact request

 By clicking the approved contact button approved the contact request and use will see the biodata contact from their contact request page.

**Challenges Task**

1. **User Dashboard :**

* Create a Got Married Route. When the user visits this route they will see the form. The form **has the following input field:**

1. **Self Biodata id**
2. **Partner Biodata id**
3. **Couple Image Link or image input fields.**
4. **Success Story Review [Text Area] Here user can share her/his feelings for using this website.**
5. **Submit Button**

**When the User clicks the submit button this information will be added to the success story database collection. And you the success story in the Home Page Success Story Section.**

**2. Admin Dashboard :**

Create a pie chart to show the total number of biodata, numbers of male biodata, numbers of female biodata, and numbers of premium biodata as well as total revenue from biodataContactRequest Payment.

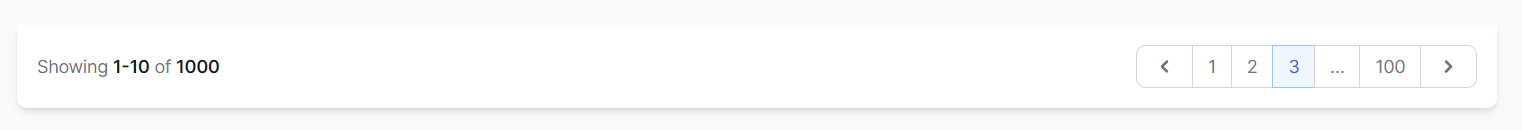
Make a success story route. Here admin can see the success story data in the tabular format.

Show the information in the table :

1. Male Biodata Id
2. Female Biodata Id
3. View Story Button

When the user clicks the view story then shows a modal and shows their success story.

3. Implement JWT on login (Email/Password and social) and store the token.

4. Implement pagination on the Biodatas page 

             For example,

**Some Rules**

1. Do not copy anything from your previous assignment. Like cards, tables, navbar, forms, etc. If you violate the rules you will receive a considerable penalty. Be careful.

**Optional Tasks**

1. Use react select for handle select input. [Link](https://react-select.com/)
2. Explore Ant Design or Material Ui
3. Explore Tanstack query Pagination
4. Implement Axios interceptor.
5. Data must be shown the first time reloading.

**What to Submit**

1. **Admin email:**
2. **Admin password:**
3. **Front-end Live Site Link:**
4. **Client Side Github Repository Link:**
5. **Server Side Github Repository Link:**