

## **B12-A11\_category-02**

**Explanation Video: [B12-A11\\_category-02.mp4](#)**

### **Assignment - 11: ScholarStream (Scholarship Management System)**

**Dear Candidates,**

We are pleased to inform you that you have successfully passed the first round of the selection process! Your application and skills have impressed us, and we are excited to move forward with you in the next stages.

We are excited to have you explore this opportunity with us. This project is designed to assess your skills, creativity, and problem-solving abilities. It will help us understand how you approach challenges and your ability to deliver high-quality solutions.



---

### **Project Name: ScholarStream (Scholarship Management Platform)**

#### **Project Overview and Discussion**

**What is the Project?** "ScholarStream" is a full-stack MERN application designed to connect students with scholarship opportunities. It allows universities or organizations to post scholarships and students to search and apply for them.

**Why should we develop the project?** To simplify the complex process of finding financial aid for education. A centralized platform helps students discover opportunities they might miss and streamlines the application review process for administrators.

#### **How the system works:**

1. **Students** browse scholarships, view details, and apply by paying an application fee.
2. **Moderators** review these applications, provide feedback, and update the application status (Pending -> Processing -> Completed).
3. **Admins** Manage users, add/edit scholarships, and view analytics.

**Similar websites / Design links:**

- [Scholarships.com](#)
- [Fastweb.com](#)

**Data Design:** You must have at least four collections:

1. **Users Collection:**
  - Stores user information, including **name**, **email**, **photoURL**, and their **role** (e.g., "Student", "Moderator", "Admin").
2. **Scholarships Collection:**
  - Stores scholarship data: **scholarshipName**, **universityName**, **universityImage**, **universityCountry**, **universityCity**, **universityWorldRank**, **subjectCategory**, **scholarshipCategory** (Full fund/Partial/Self-fund), **degree** (Diploma/Bachelor/Masters), **tuitionFees** (optional), **applicationFees**, **serviceCharge**, **applicationDeadline**, **scholarshipPostDate**, **postedUserEmail**.
3. **Applications Collection:**
  - Stores application data: **scholarshipId**, **userId**, **userName**, **userEmail**, **universityName**, **scholarshipCategory**, **degree**, **applicationFees**, **serviceCharge**, **applicationStatus** (default: pending), **paymentStatus** (unpaid/paid), **applicationDate**, **feedback** (added by moderator).
4. **Reviews Collection:**
  - Stores students review data: **scholarshipId**, **universityName**, **userName**, **userEmail**, **userImage**, **ratingPoint**, **reviewComment**, **reviewDate**.

---

## Ensure the Following things to get 100% mark ( No mark will contain)

- Include at least **20 meaningful commits on the client side & 12 meaningful commits on the server side** with descriptive messages.
- Include a **README.md** file with the project name, purpose, live URL, key features, and any npm packages you have used.
- Secure Firebase configuration keys using environment variables.
- Secure your MongoDB credentials using the environment variable.

- Create a design that encourages recruiters. Color contrast should please the eye & ensure that the website has proper alignment, space, and the website does not express a "gobindo" design.
- If we find your project similar to any project of your module / conceptual / assignment, you will get 0.

## Deployment Guideline

If your Deployment is not okay you will get 0 and may miss the chance of our upcoming rewards.

- Ensure that your server is working perfectly on production and not throwing any **CORS / 404 / 504** Errors.
- Ensure that your Live Link is working perfectly and that it is not showing errors on Landing in your system.
- **⚠️** ensure that the page doesn't throw any error on reloading from any routes.
- **⚠️** Add your domain for authorization to Firebase if you use **Netlify / surge / Vercel**
- **⚠️** Logged in User must not redirect to Login on reloading any private route

---

## Layout & Page Structure

**Main Layout:** Your site will have a main layout with a Navbar and Footer visible on most pages (excluding the dashboard layout).

**Navbar:**

- **Always Visible:** Logo, Home, All Scholarships.
- **Not Logged In:** Login Button, Register Button.
- **Logged In:** User Profile Image (with dropdown), Dashboard link, Logout.

**Footer:**

- A standard footer with Logo, Copyright, and Social Media links.

**Page List:**

- **Home**
- **All Scholarships** (Shows all Scholarships)

- ~~Scholarship Details~~ (Shows details for one Scholarship)
  - ~~Login / Register~~
  - Payment
    - Checkout Page ( Private )
    - Payment Success Page
    - Payment Failed Page
  - Dashboard (This will have its own layout)
  - ~~404 Error Page~~
- 

## Authentication System

### ~~Register Page:~~

- Form Fields: Name, Email, Photo URL, Password.
- **Default Role:** When a user registers, their role should be "**Student**" by default.
- ~~Implement password validation (e.g., min 6 chars, capital letter, special character).~~



### ~~Login Page:~~

- Form Fields: Email, Password.
  - ~~Social Login: Must include at least one social login (e.g., Google).~~
- 

## Home Page

**Banner:** A hero section with a title, description, and a "**Search Scholarship**" button.

**Top Scholarships (Dynamic):** A section displaying the top 6 scholarships (e.g., those with the lowest application fees or most recent post date). Each card must have a "**View Details**" button.

**Animation:** You must implement animation on the Home page using **framer-motion**.

**Two Extra Sections:** Add two extra static sections. For example:

1. A "Success Stories" or "Testimonials" section.
  2. A "**Contact Us**" or "**F.A.Q**" section.
- 

## All Scholarships Page

~~Display all scholarships available on the platform in a responsive grid layout.~~

**Search Functionality:** Include a search bar to find scholarships by Scholarship Name, University Name, or Degree.

**Filter Functionality:** Include options to filter scholarships by specific criteria (e.g., Scholarship Category, Subject Category, or Location).

**Card Information:** Each scholarship card must display the University Image, University Name, Scholarship Category, Location, Application Fees (if any), and a "View Details" button. **"View Details" Button:** Clicking this button navigates the user to the Scholarship Details page.

---

## Scholarship Details Page

Display comprehensive information about the selected scholarship, including University Image, Scholarship Name, University World Rank, Deadline, Location, Application Fees, Scholarship Description, and Stipend/Coverage details.

**Apply Button:** A prominent "Apply for Scholarship" button.

- By clicking this redirects the user to the Payment/Checkout page.

**Reviews Section:** Display all reviews associated with this scholarship/university. Show the reviewer's image, name, date, rating, and comment. This data comes from the **Reviews Collection**.

---

## Dashboard Layout: ( 3 Roles)

You must create a single, separate dashboard layout (e.g., `/dashboard`) with a sidebar.

The links shown in the sidebar **must be conditional** based on the user's **role (Student, Moderator or Admin)**.

### 1. Admin Dashboard

- **My Profile:** View profile info.
- **Add Scholarship:** A form to create new scholarships.
  - Fields: Scholarship Name, University Name, Image, Country, City, World Rank, Subject Category, Scholarship Category, Degree, Tuition Fees

~~(optional), Application Fees, Service Charge, Deadline, Post Date, User Email.~~

- ~~Manage Scholarships:~~ Display a table of all scholarships added to the platform.
  - ~~Action: "Update"~~ (Button to edit scholarship details).
  - ~~Action: "Delete"~~ (Button to remove the scholarship).
- ~~Manage Users:~~ A table of all users.
  - ~~Filter:~~ Dropdown to filter by role (Student/Moderator/Admin).
  - ~~Action: Change Role~~ (Promote Student to Moderator or Admin, or demote).
  - ~~Action: Delete User.~~
- ~~Analytics:~~ Visualize platform data.
  - Total Users, Total Fees Collected, Total Scholarships.
  - ~~Chart:~~ A Bar Chart or Pie Chart showing application counts per University or Scholarship Category.

## 2. Moderator Dashboard (Management Role)

- **My Profile:** View profile info.
- **Manage Applied Applications:** Display a table of all applications submitted by students.
  - **Columns:** Applicant Name, Applicant Email, University Name, Application Feedback, Application Status, Payment Status, Actions.
  - **Actions:**
    - **Details:** Opens a modal with full applicant and scholarship details.
    - **Feedback:** A button that opens a modal to write feedback.
    - **Status Update:** A dropdown or buttons to change status to "Processing" or "Completed".
    - **Cancel:** A button to reject the application (changes status to "rejected").
- **All Reviews:** See all reviews posted by students.
  - **Action: "Delete"** (to moderate inappropriate content).

## 3. Student Dashboard (User Role)

- **My Profile:** View profile info.
- **My Applications:** Display a table of all scholarships the student has applied for.
  - **Columns:** University Name, University Address, Feedback (from moderator), Subject Category, Application Fees, Application Status, Actions.
  - **Actions:**
    - **Details:** (Button) Opens a modal with the full application details.
    - **Edit:** (Button) Visible *only* if the application status is "pending".
    - **Pay:** (Button) Visible *only* if the application status is "pending" AND payment status is "unpaid".
    - **Delete:** (Button) Visible *only* if the application status is "pending".
    - **Add Review:** (Button) Visible *only* if the application status is "completed". Clicking this opens a modal with:
      - Rating Input (e.g., 1-5 stars)
      - Comment Input (Text area)
      - Submit Button
- **My Reviews:** Display a table of all reviews the student has written.
  - **Columns:** Scholarship Name, University Name, Review Comment, Review Date, Rating, Actions.
  - **Actions:**
    - **Edit:** Opens a modal to update the review and rating.
    - **Delete:** Removes the review.

## Payment System

- **Checkout Page:** When a Student clicks "Apply" on a **Scholarship Details** page, they must be redirected to a payment page.
- **Stripe Integration:** You must implement **Stripe** payment.
- **Save Application:**

- **On successful payment:** The application data is saved to the applications collection, paymentStatus is set to “**paid**”, and the user is redirected to the **Payment Success page**.
  - **On payment failure:** The application data is saved to the applications collection, paymentStatus is set to “**unpaid**”, and the student can try to pay again from their dashboard.
- **Payment Success Page:**
    - Must display: “**Payment Successful**” message, **Scholarship Details (e.g., scholarshipName, universityName etc)**, and **Amount Paid**.
    - **Action:** Button to “**Go to My Applications**”.
  - **Payment Success Page:**
    - Must display: “**Payment Failed**” message, **Scholarship Name**, and an **Error Message** (if available).
    - Action: Button to “**Return to Dashboard**” (where they can retry payment).

## Additional

- **Loading:** Add a loading spinner/skeleton on all pages that fetch data.
- **Error Page:** Create a custom 404 Error page.

## UI Design Requirements:

1. **Unique Design:** Your design must be unique and not copied from course modules or conceptual sessions.
2. **Consistency:** Keep the main heading style (font, size, color) consistent.
3. **Readability:** Keep paragraph spacing balanced and text easily readable.
4. **Uniformity:** Maintain uniform image sizes and spacing.
5. **Buttons:** Use the same button style as on the home page.
6. **Alignment:** Ensure good spacing and proper alignment.
7. **Navbar:** Keep the heading/logo style consistent.
8. **Layout:** Use a grid layout with equal image sizes and keep all cards equal height and width.

9. **Logo:** Use the new X logo instead of the old Twitter bird if applicable.
10. **Responsiveness:** Must be responsive for all devices (mobile, tablet, desktop).
11. **Loading:** Add a loading spinner/skeleton on all data-fetching pages.
12. **Error Page:** Custom 404 page.

### **Dashboard UI Requirements:**

- Must be fully responsive for mobile and tablet screens.
- Must have a consistent color theme.
- Must be a full-width dashboard.
- Must include charts and graphs for quick data visualization (Admin).
- Must have a user profile section (or link).

### Resources:

- <https://uiverse.io/>
- <https://devmeetsdevs.com/>
- <https://bootcamp.uxdesign.cc/free-images-and-resources-collection-for-website-c77f2fc46ce5>
- <https://themeforest.net/>
- <https://codecanyon.net/>

---

## Challenge Requirement Guideline

1. **JWT Token Verification:**
  - Implement JWT for secure API calls.
  - Create **verifyAdmin** middleware to protect Admin routes.
  - Create **verifyModerator** middleware to protect Moderator routes.
2. **Search & Filter (All Scholarships Page):**
  - Implement **Server-side Search** by Scholarship Name, University Name, or Degree.
  - Implement **Server-side Filter** by Country or Category.
3. **Sort (All Scholarships Page):**

- Implement **Server-side Sort** by Application Fees (Ascending/Descending) or Post Date.
4. **Pagination:**
- Implement pagination on the "All Scholarships" page.

► ► Alternative CSS frameworks (like **ShadCN**, **Chakra UI**, **Material React**) cannot be directly included in the **challenge part**. Students should build their projects using **DaisyUI** only.

## What to Submit

Admin Email:

Admin Password:

Live Site Link :

Github Repository ( server ) :

Github Repository ( client ) :



## Optional Requirement Guideline

- **Scholarship Recommendations:** On the Scholarship Details page, show a "You may also like" section displaying scholarships from the same category.
- **Wishlist:** Allow students to bookmark scholarships they are interested in (saved to a "wishlist" collection or user array).

文  
CA