SW Engineering CSC 648/848 Section 4

EduBridge

By

Syntaax Squaad

Team 4

Member	Role
Naisarg Halvadiya	Team Lead
Shail Patel	Backend Lead
Dylan Nguyen	Scrum Master I
James Dixon	Scrum Master II
Pankuri Khare	Frontend Lead
Riken Kapadia	GitHub Master

Milestone 2

03/22/2024

Milestone	Date
0	02/21/2024
1	03/06/2024
2	03/22/2024

1. Data Definitions

1. user: An entity that stores registered user info

a. Attributes:

i. userID - users unique Identifier

ii. displayName - users name

iii. email - user email

iv. password - user password

v. userRole - the user role used to identify privileges, ie student, teacher, admin

b. Privileges:

i. Privileges are provided through the users role

2. role: Stores role type and permissions

a. Attributes:

i. roleID - the roles identifier

ii. roleName - student , teacher, admin

b. Privileges:

i. student

1. View content
2. Upload content

ii. teacher

1. View content
2. Upload content
3. Filter / Moderate content

iii. admin

1. Set user roles

3. **userRole: This entity sets the relationship between roles and users**
    a. Attributes:
        i. userID - the id of the user
        ii. roleID - the id of role assigned to the user
4. **content: Represents uploaded material the application serves**
    a. Attributes:
        i. contentID - content identifier
        ii. userID - uploader id
        iii. name - name of content
        iv. description - contents description
        v. type - file type of item
        vi. uploadDate - date content was uploaded
        vii. location - crm location / file path
        viii. viewCount - view count of content
        ix. visibility - ie public/private
5. **comment:**
    a. Attributes:
        i. commentID - comment identifier
        ii. contentID - content the comments on
        iii. userID - id of commenter
        iv. text - comment
        v. uploadDate - comment date
6. **like:**
    a. Attributes:
        i. likeID - identifier
        ii. contentID - content liked
        iii. userID - user who liked
        iv. date - date liked

7. dislike:

a. Attributes:

i. dislikeID - identifier

ii. contentID - content disliked

iii. userID - user who disliked

iv. date - date disliked

2. Functional Requirements

1. User Creation / Management

- ID: 01
- Description: Users should be able to register and log in. Users shall have roles such as admin, student, instructor, and assistant.
- Priority: 1

1.1. Registration:

- Users shall register using a username, email, and password.

1.2. Login:

- Users shall be able to log in using their username and password

1.3. Role Management:

- A role must be assigned to each user, by default with lowest permissions student
- Admin accounts manage user roles.

2. Student Authentication

- ID: 02
- Description: Users must have a .edu email.
- Priority: 1

2.1. Email Verification:

- Users will only be allowed to register with a .edu email.
- Users will have to access this email to retrieve a OTP to verify the email.

3. Content Upload

- ID 03/1
- Description: Users must be able to upload diverse types of content
- Priority: 1

3.1. File Format Compatibility:

- Users shall be able to upload common video,audio, and text files.
  - MP4
  - PDF
  - DOCX

4. Content Playback

- ID: 03/2
- Description: The application shall offer appropriate playback for common file types
- Priority: 1

4.1. Video Playback:

- Users shall be able to watch uploaded videos in application.

4.2. Text Viewer:

- Users shall be able to view text files through a built in application renderer.

5. Content Download

- ID: 04
- Description: Users shall download content from the platform for offline viewing
- Priority: 2

6. Comment / Feedback

- ID: 05
- Description: Users shall comment on uploaded content and reply to other comments. Additionally Users shall like or dislike content.
- Priority: 2

6.1. Comment Posting:

- Users shall be able to post comments on content
- Users shall be able to post private feedback on content for uploaders
- Comments shall support rich text formatting and attachments

6.2. Reply System:

- Users shall be able to comment on other comments.

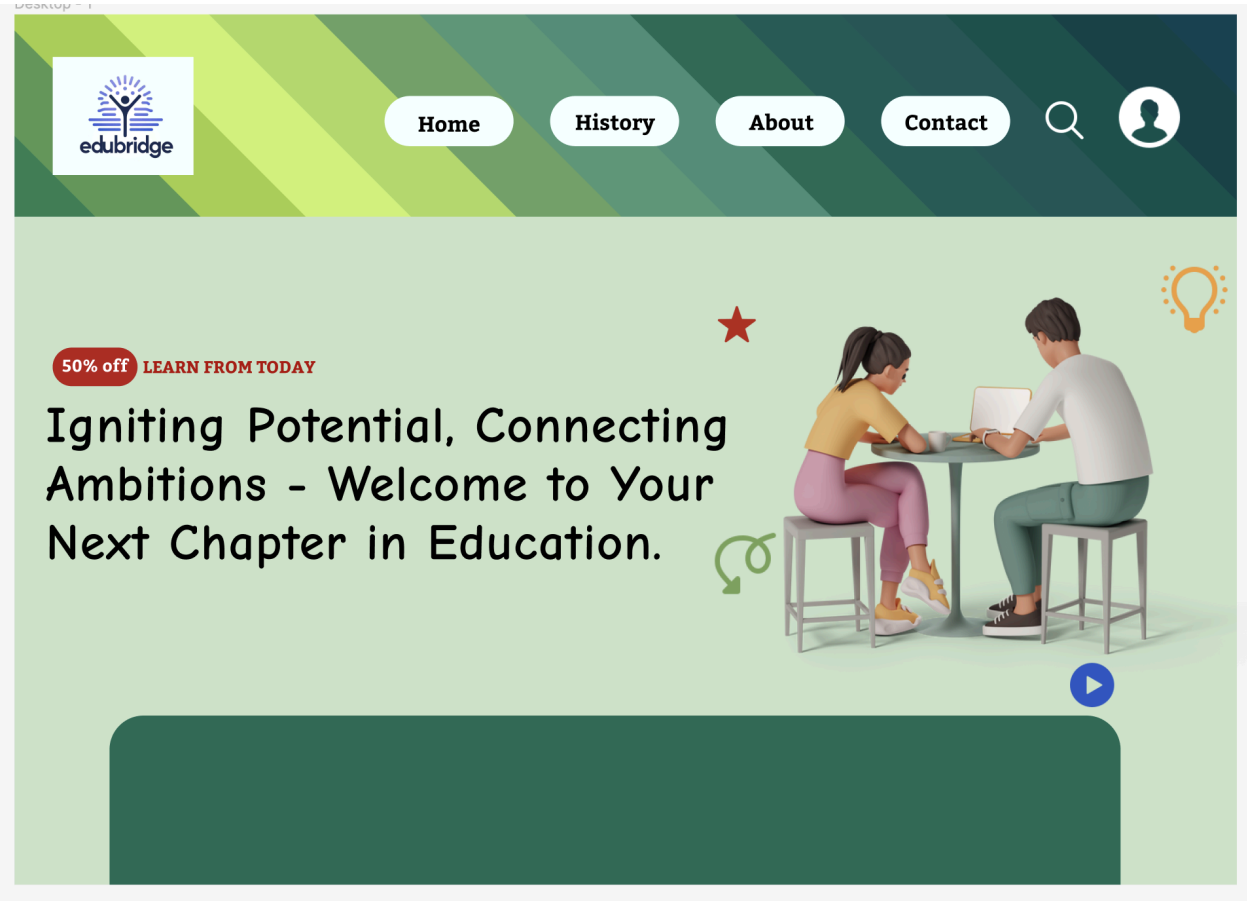
6.3. Like System:

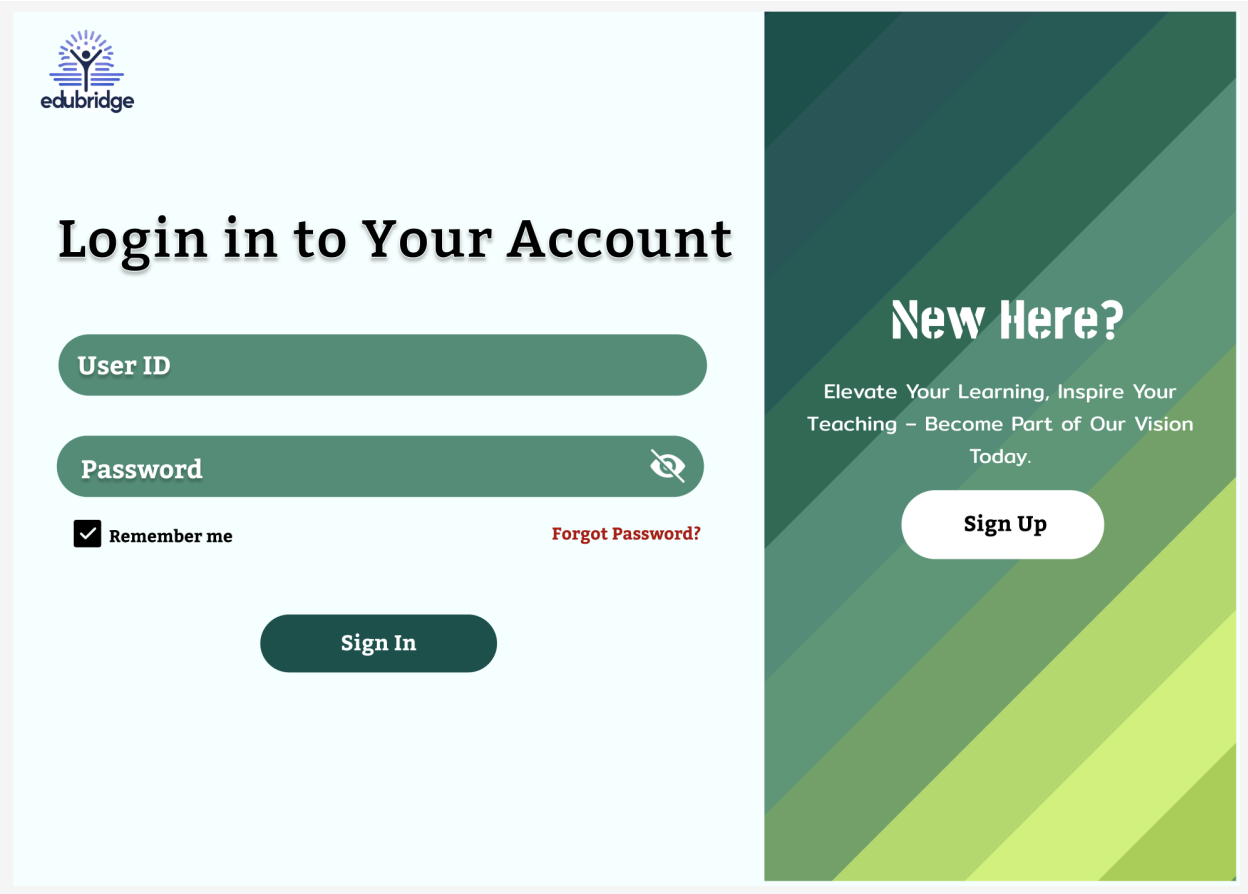
- Users shall be able to like content or comments.
- Users shall be able to dislike content or comments.

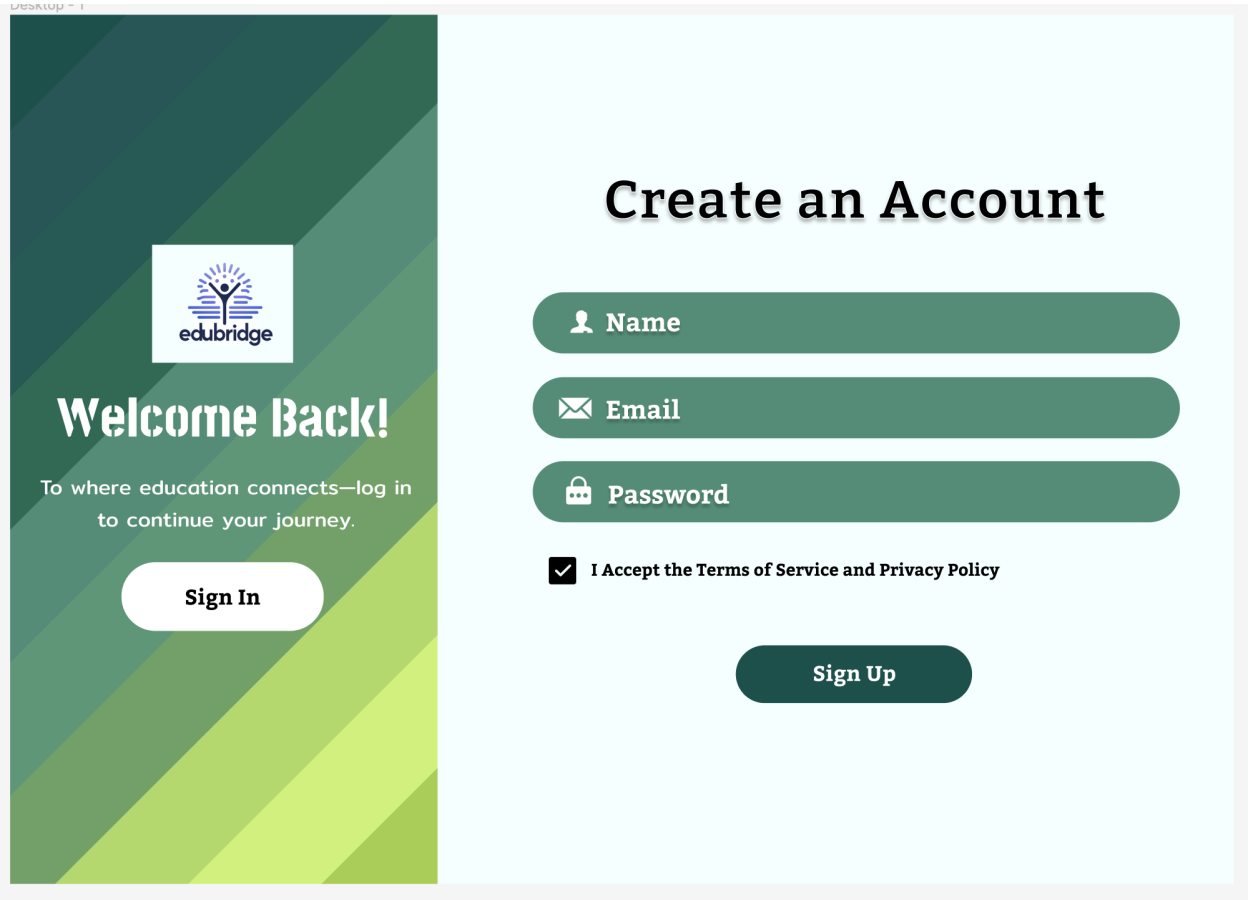
3. UI Mockups and UX Flows:

[Click Here for Interactive Module of UX Flows]

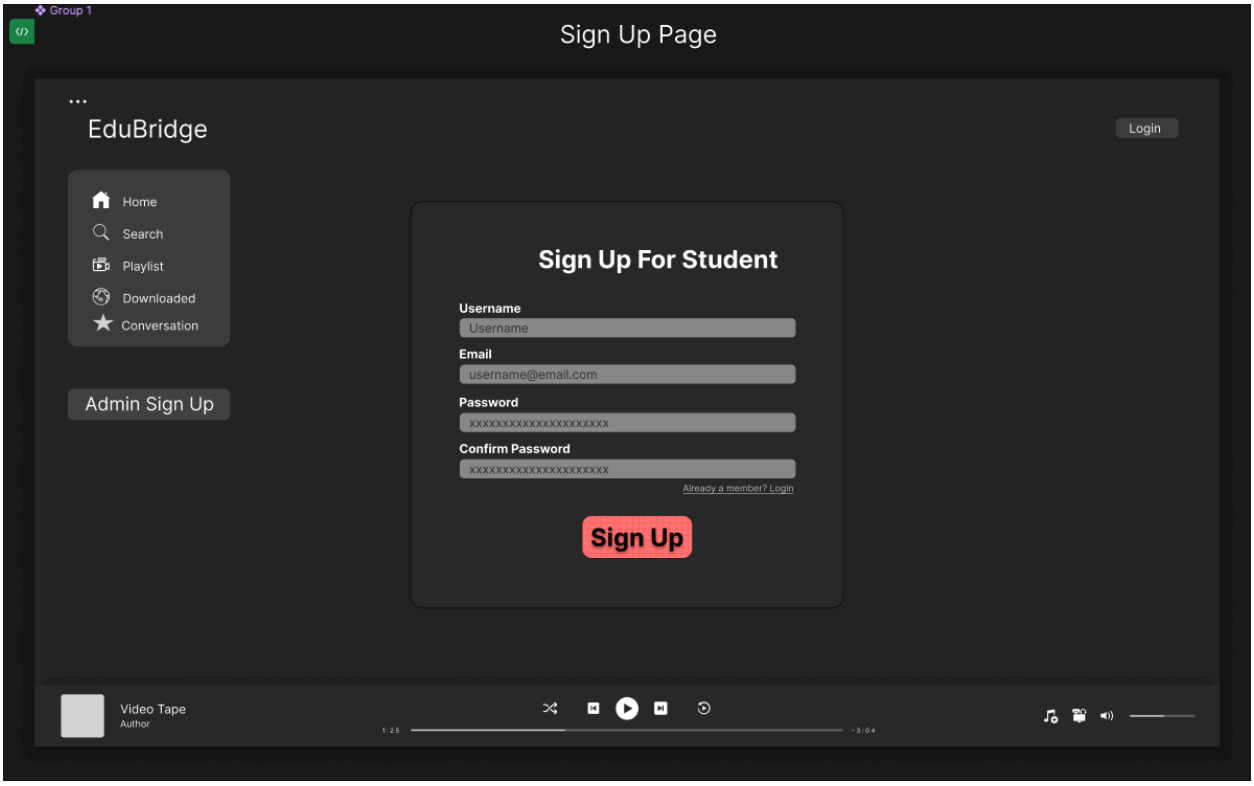
Basic Expected Output:

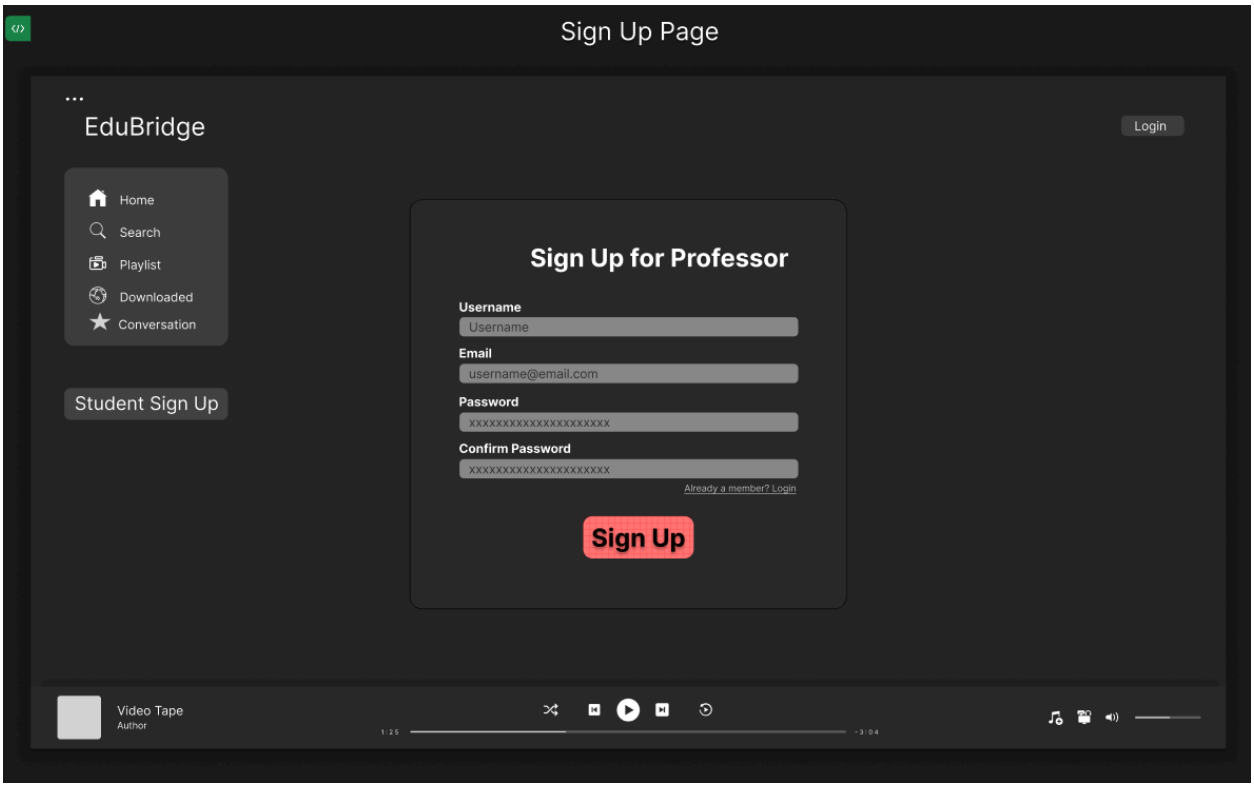

edubridge
Home
History
About
Contact
50% off
LEARN FROM TODAY
Igniting Potential, Connecting Ambitions - Welcome to Your Next Chapter in Education.

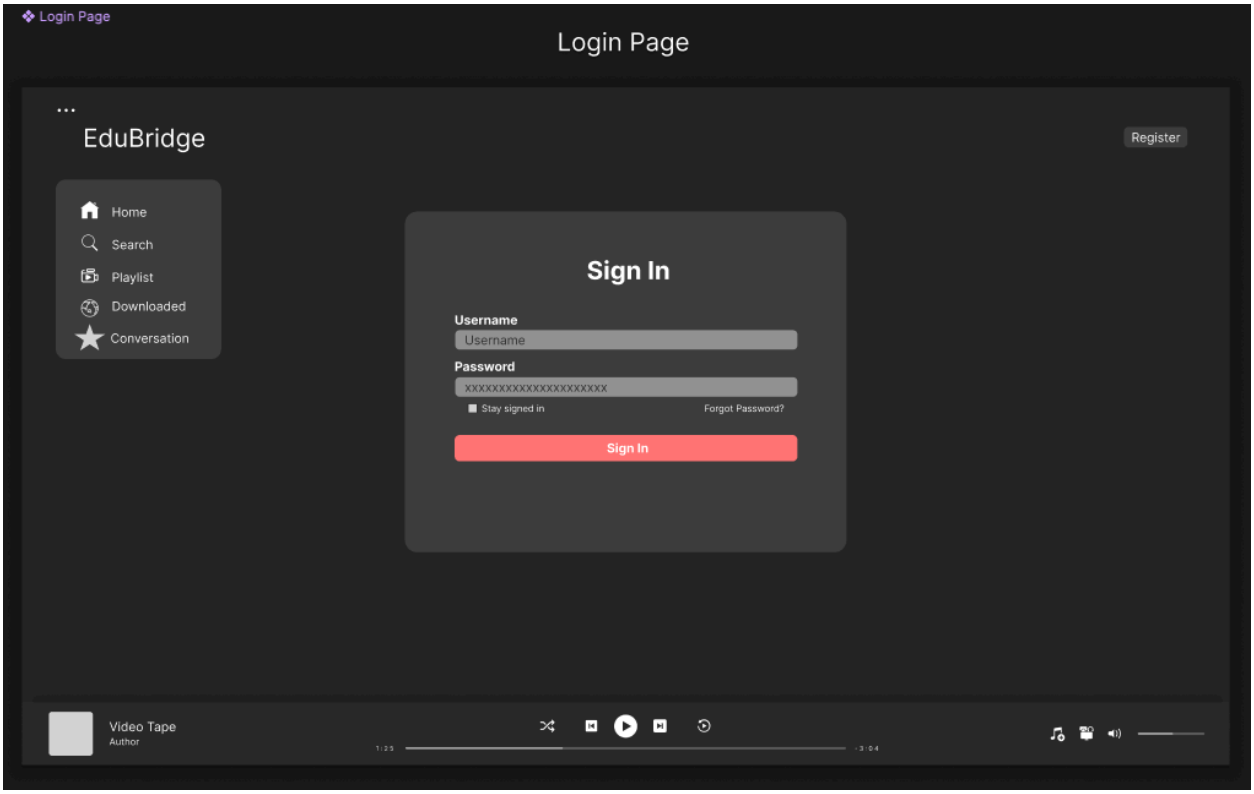

edubridge
Login in to Your Account
User ID
Password
Remember me
Forgot Password?
Sign In
New Here?
Elevate Your Learning, Inspire Your Teaching – Become Part of Our Vision Today.
Sign Up

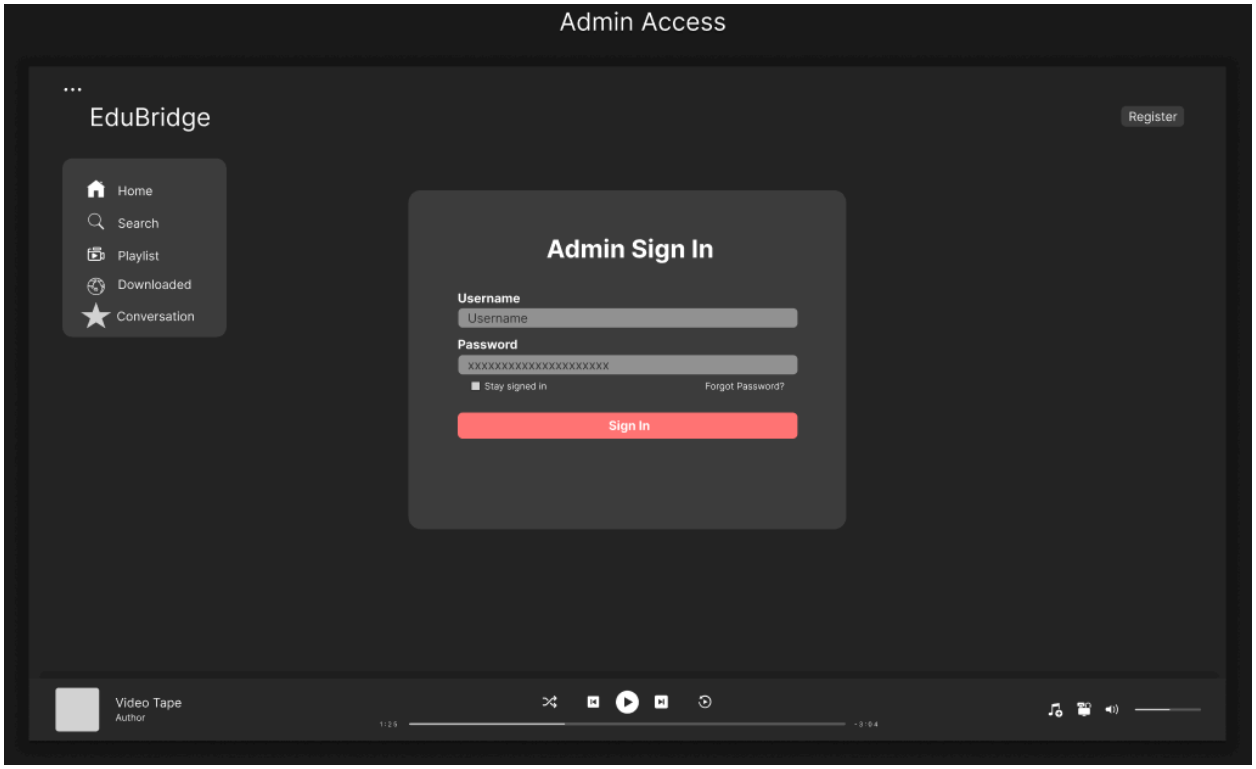

edubridge
Welcome Back!
To where education connects—log in to continue your journey.
Sign In
Create an Account
Name
Email
Password
I Accept the Terms of Service and Privacy Policy
Sign Up

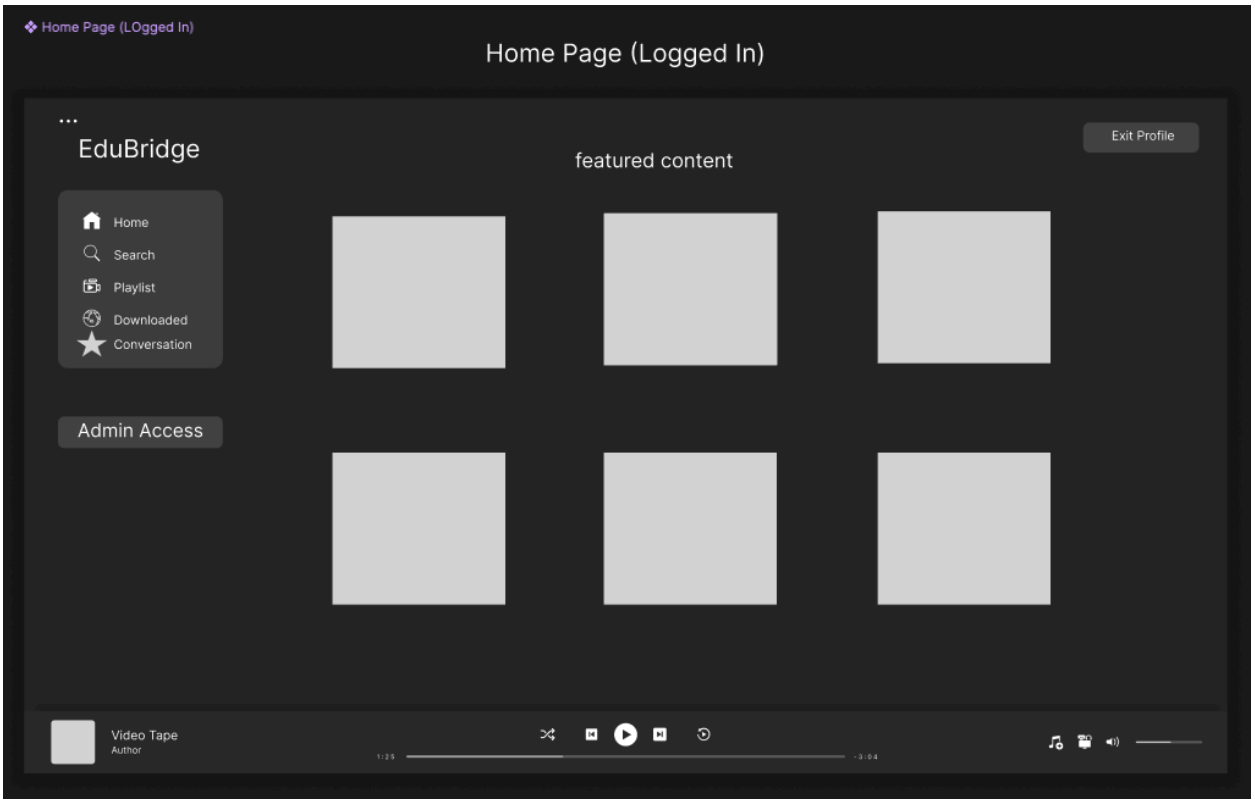
UI/UX Mockups:

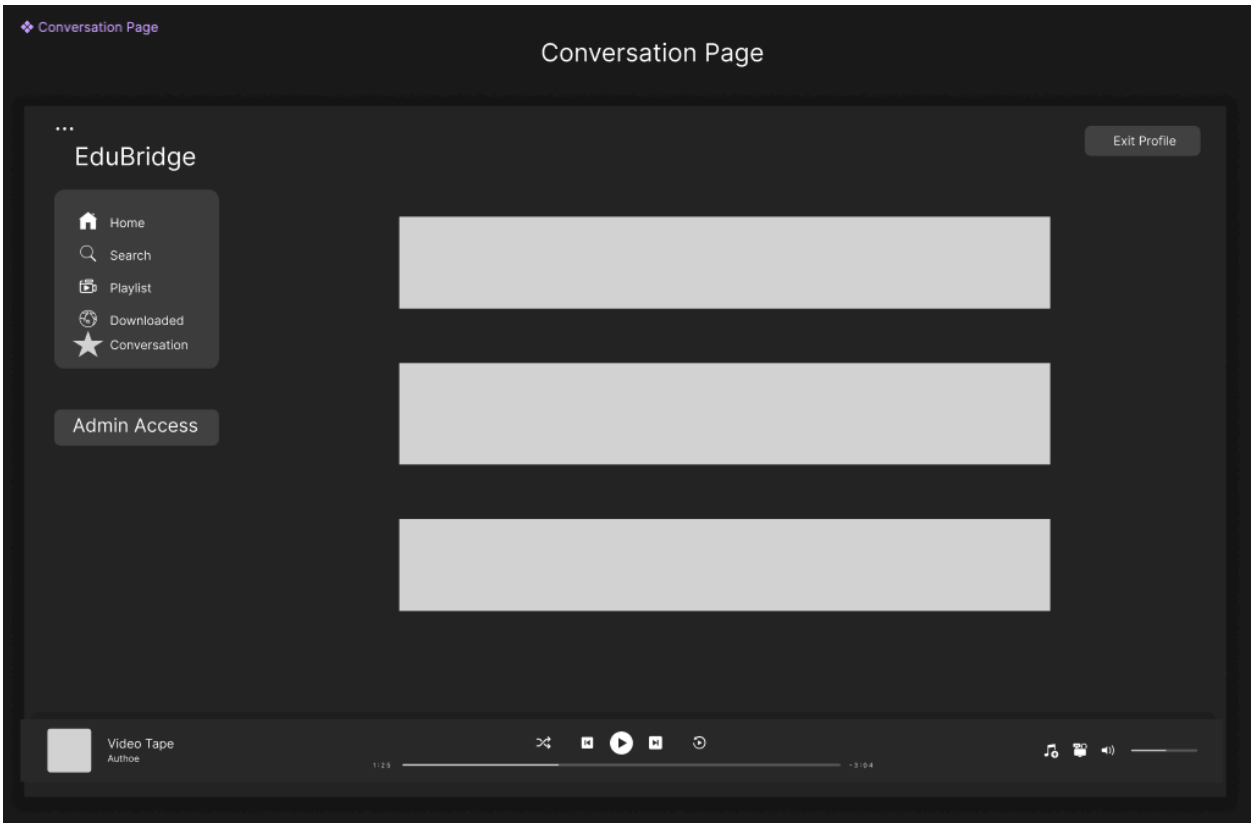


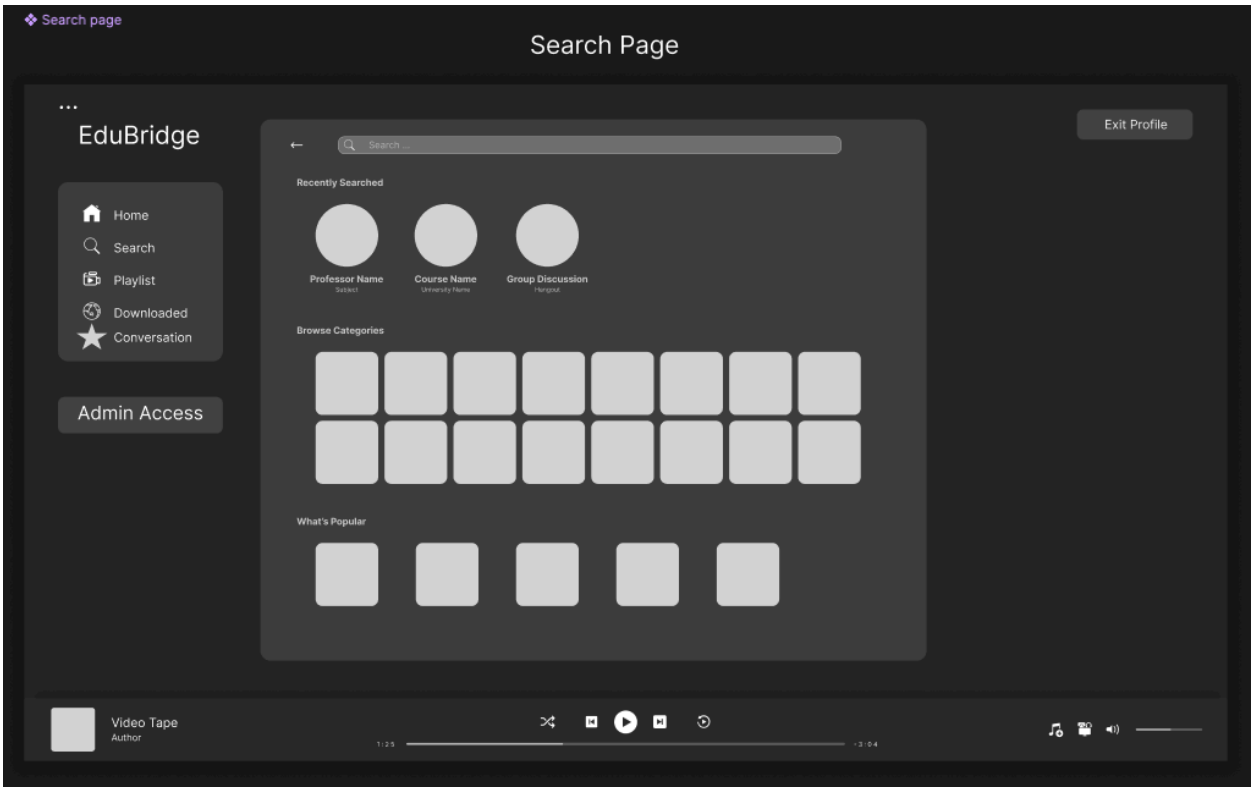

Sign Up Page
EduBridge
Login
Home
Search
Playlist
Downloaded
Conversation
Student Sign Up
Sign Up for Professor
Username
Username
Email
username@email.com
Password
xxxxxxxxxxxxxxxxxxxx
Confirm Password
xxxxxxxxxxxxxxxxxxxx
Already a member? Login
Sign Up
Video Tape
Author

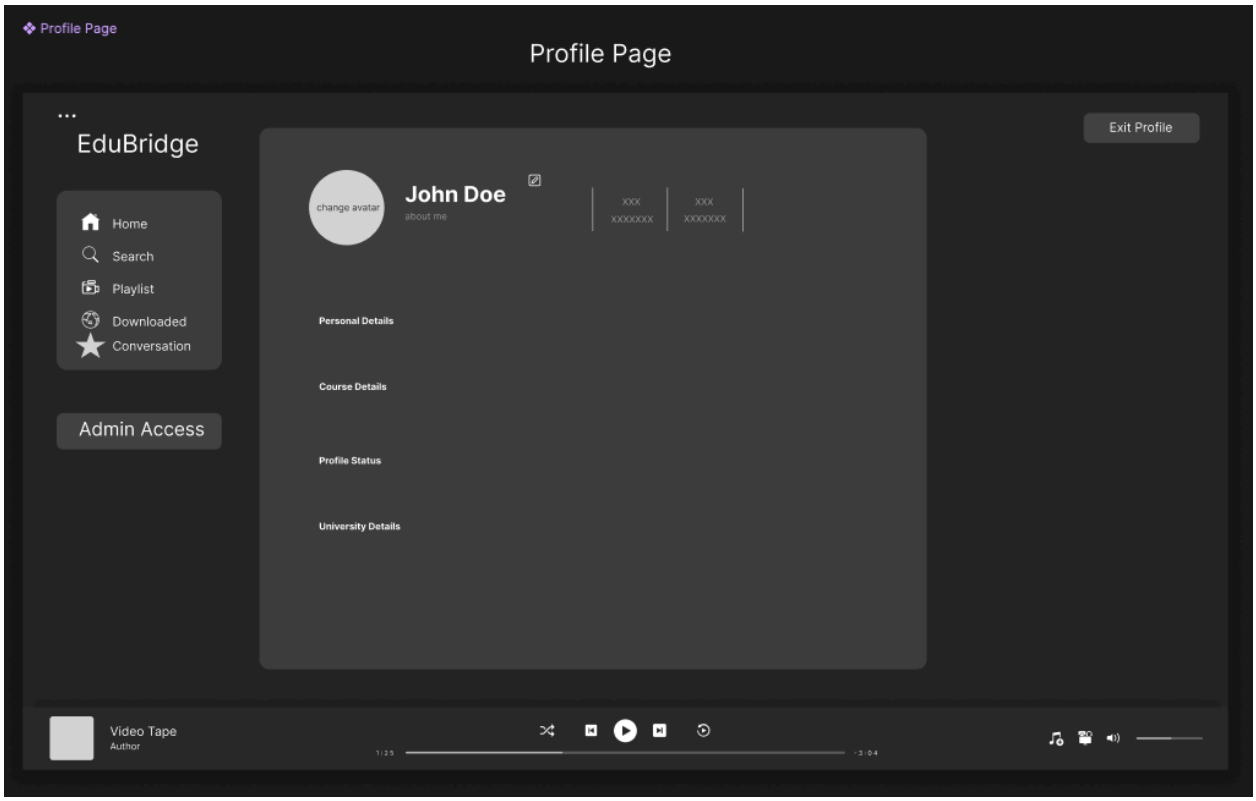

Login Page
Login Page
EduBridge
Register
Home
Search
Playlist
Downloaded
Conversation
Sign In
Username
Username
Password
xxxxxxxxxxxxxxxxxxxx
Stay signed in
Forgot Password?
Sign In
Video Tape
Author


Admin Access
EduBridge
Register
Home
Search
Playlist
Downloaded
Conversation
Admin Sign In
Username
Username
Password
xxxxxxxxxxxxxxxxxxxx
Stay signed in
Forgot Password?
Sign In
Video Tape
Author


Home Page (LOgged In)
Home Page (Logged In)
EduBridge
Exit Profile
featured content
Home
Search
Playlist
Downloaded
Conversation
Admin Access
Video Tape
Author


Conversation Page
Conversation Page
EduBridge
Exit Profile
Home
Search
Playlist
Downloaded
Conversation
Admin Access
Video Tape
Authoe


Search page
Search Page
EduBridge
Exit Profile
Search ...
Recently Searched
Professor Name
Course Name
Group Discussion
Home
Search
Playlist
Downloaded
Conversation
Admin Access
Browse Categories
What's Popular
Video Tape
Author


Profile Page
Profile Page
EduBridge
Exit Profile
change avatar
John Doe
about me
xxx
xxxxxxx
xxx
xxxxxxx
Home
Search
Playlist
Downloaded
Conversation
Admin Access
Personal Details
Course Details
Profile Status
University Details
Video Tape
Author

4. High Level Architecture, Database Organization

MySQL Database Organization:

USERS // added, deleted,searched, displayed

- user_id (primary key)
- name
- email
- password
- is_Authenticated

ROLES // added, deleted

- role_id (primary key)
- role_name

USER_ROLES // added, deleted, displayed

- user_id (primary key, foreign key to USERS.user_id)
- role_id (primary key, foreign key to ROLES.role_id)

CONTENT // added, deleted,searched, displayed

- content_id
- user_id (foreign key to USERS.user_id)
- title
- description
- type

upload_date

location

view_count

visibility

COMMENTS // added, deleted, displayed

comment_id (primary key)

content_id(foreign key to CONTENT.content_id)

user_id (foreign key to USERS.user_id)

text

upload_date

LIKES // added, deleted, displayed

like_id (primary key)

content_id(foreign key to CONTENT.content_id)

user_id (foreign key to USERS.user_id)

date

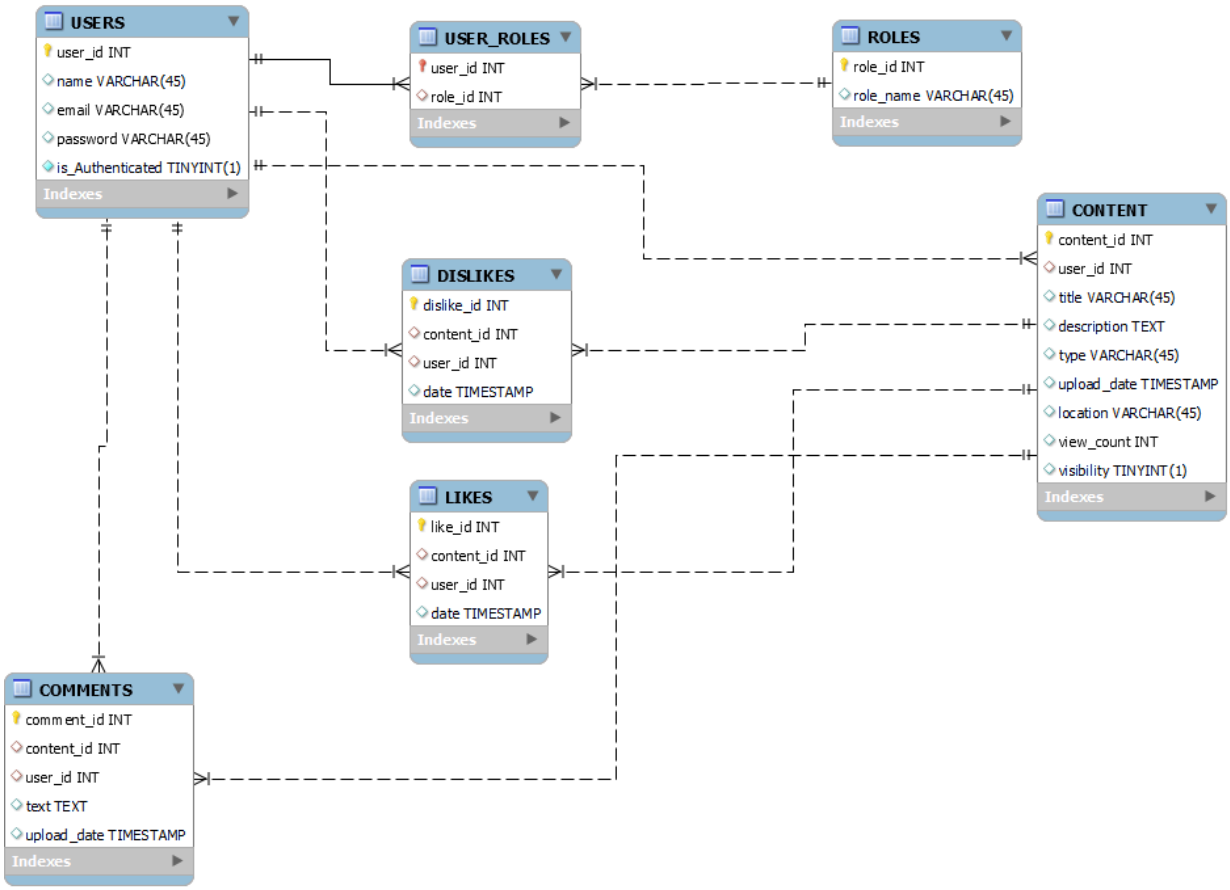
DISLIKES // added, deleted, displayed

dislike_id (primary key)

content_id(foreign key to CONTENT.content_id)

user_id (foreign key to USERS.user_id)

date



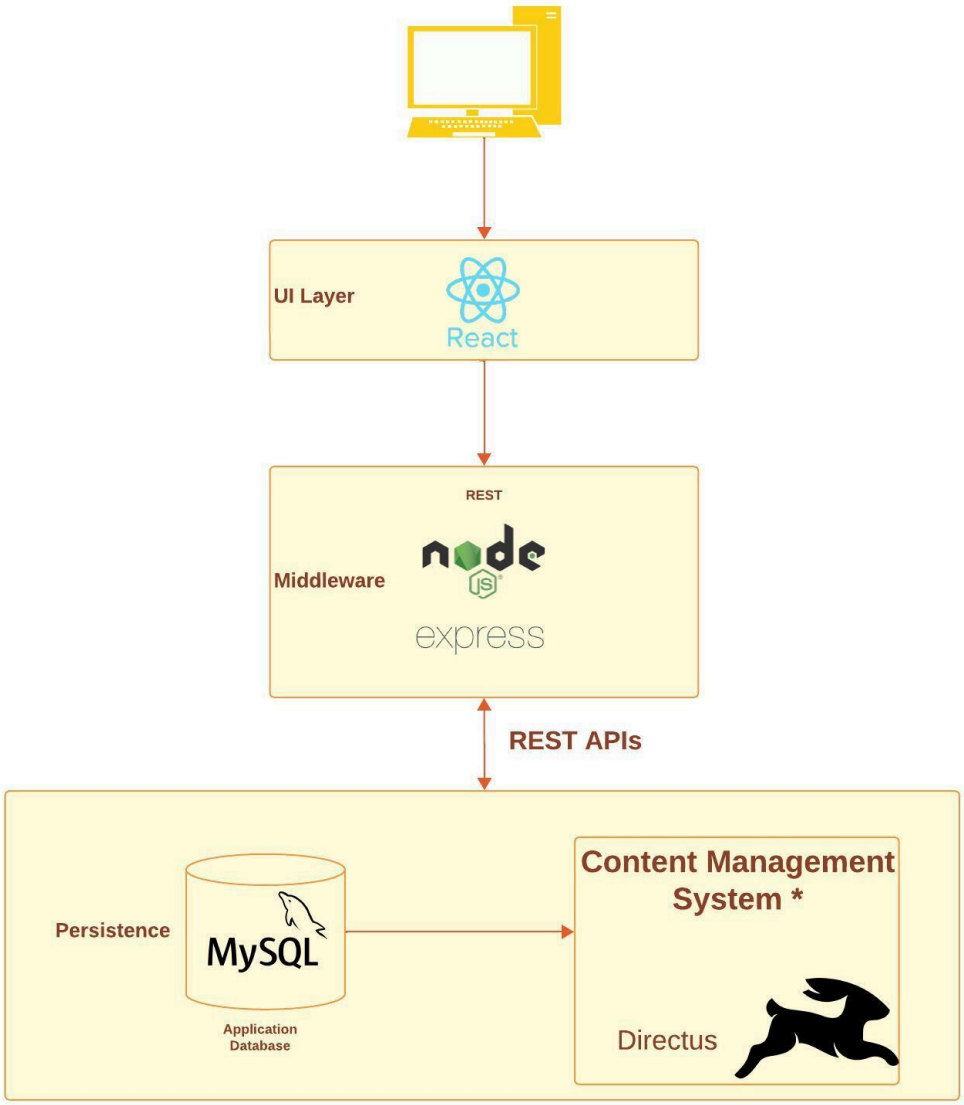
Current APIs & Architecture:

Express Endpoints:

Endpoint	Method	Description	Input	Process	Output
/signup	POST	Receives email and password from the user.	Email, Password	Checks if the user is already created, checks if the email is .edu, encrypts password and adds user to the database.	Confirmation message
/login	POST	Receives email and password from the user.	Email, Password	Checks if a user is in the database.	Returns authorization
/users	GET	Returns all users in the database.	None	N/A	List of users

/verify-otp	POST	Enters email, password, and OTP to validate the user’s authenticity.	Email, Password, OTP	Checks the OTP validity for the given user.	Verification status

Architecture



*** Note: Content Management System is subject to change as we are still researching**

5. Identify actual key risks for your project at this time

Skills Risks

Risk

- We are currently looking into CMS to handle our content. It is possible we may need help finding a CMS that will work and is affordable.
- Need to authenticate the API calls.

Mitigation

- Research more on various open-source CMS.
- Add basic auth to the API calls.

Schedule / Teamwork Risks

Risk

Arranging meetings is difficult, individually everyone is very busy especially now that we are later in our education where the class intensity may rise more.

Mitigation

We have meetings as often as we can and often meet in smaller teams for specific tasks, i.e. frontend team may have to meet each other to discuss specific ui elements, or those working on the database need to meet with each other.

Everyone is very active in discord where we keep each other up to date, and coordinate. Additionally we get to utilize online video calls to help mitigate the time cost from commuting.

Legal/Content Risks

Risk

As with any content hosting platform, we come to the problem of user-uploaded content. Users have the potential to upload any kind of illegal, immoral, or unkind content.

Mitigation

We will have user uploads verified through teacher/administrator roles.

Project Management

In our scrum meetings, which were held twice a week, each member started with their given tasks and proceeded to carry them out. Additionally, updates about their assigned tasks were updated through Discord which allowed everyone to stay aware of the current progress.

- Naisarg focused on the comparison of functionalities offered by Alfresco, drupal, directus, and Nuxeo CMS while setting them up locally to see which one best fits the use case.
- Shail crafted the backend functionalities for the users (login and sign-up).
- Pankuri and Riken collaborated on the UX as well as the project representation of UX flow through Figma.
- James and Dylan focused on the database functionalities and schema design.