

EXERCISE-1

1. Serve the Application to Run Through a Single Port:

- Ensure that both the frontend (React) and backend (Express) applications are running on the same port.

2. Adding File Upload Field in Frontend:

- Modify the frontend React code to include a file upload field in the student registration form. This field will allow users to upload an image of the student along with other details.

3. Sending Image to Express Backend:

- Modify the frontend code to send the image file along with other student details to the Express backend.
- Ensure that the frontend makes a POST request to the appropriate endpoint in the backend (e.g., `/api/submitdata`) and sends the image file as part of the form data.

4. Storing Image in S3 Bucket in Express Backend:

- Modify the Express backend code to handle file uploads using Multer middleware.
- Configure the Multer storage engine to upload files to an S3 bucket. Ensure that the AWS SDK is properly configured with access keys and permissions.
- When a file (image) is uploaded, store it in the specified S3 bucket and obtain the URL (location) of the uploaded image.
- Save the student details along with the image URL (location) in MongoDB. Ensure that the student schema includes a field for storing the image URL.

5. Update Frontend to Display Students with Images:

- Modify the frontend code to display student information along with their images.
- Fetch the list of students from the Express backend, ensuring that the student objects include the image URLs.
- Use the image URLs to display student images in the frontend application.

6. Testing:

- Test the application to ensure that students can be registered with images, images are successfully stored in the S3 bucket, and student information along with image URLs are stored in MongoDB.
- Verify that the frontend displays the list of registered students along with their images correctly.