1. ReactJS - CollegeAdmissionSystem

Create an admission portal for a college that enables students to apply for available courses. This application makes use of redux toolkit to implement redux logic.

Follow the instructions given for each component and complete the application.

IDE Instruction:

To run the application: **Run -> Run**. (from menu bar)

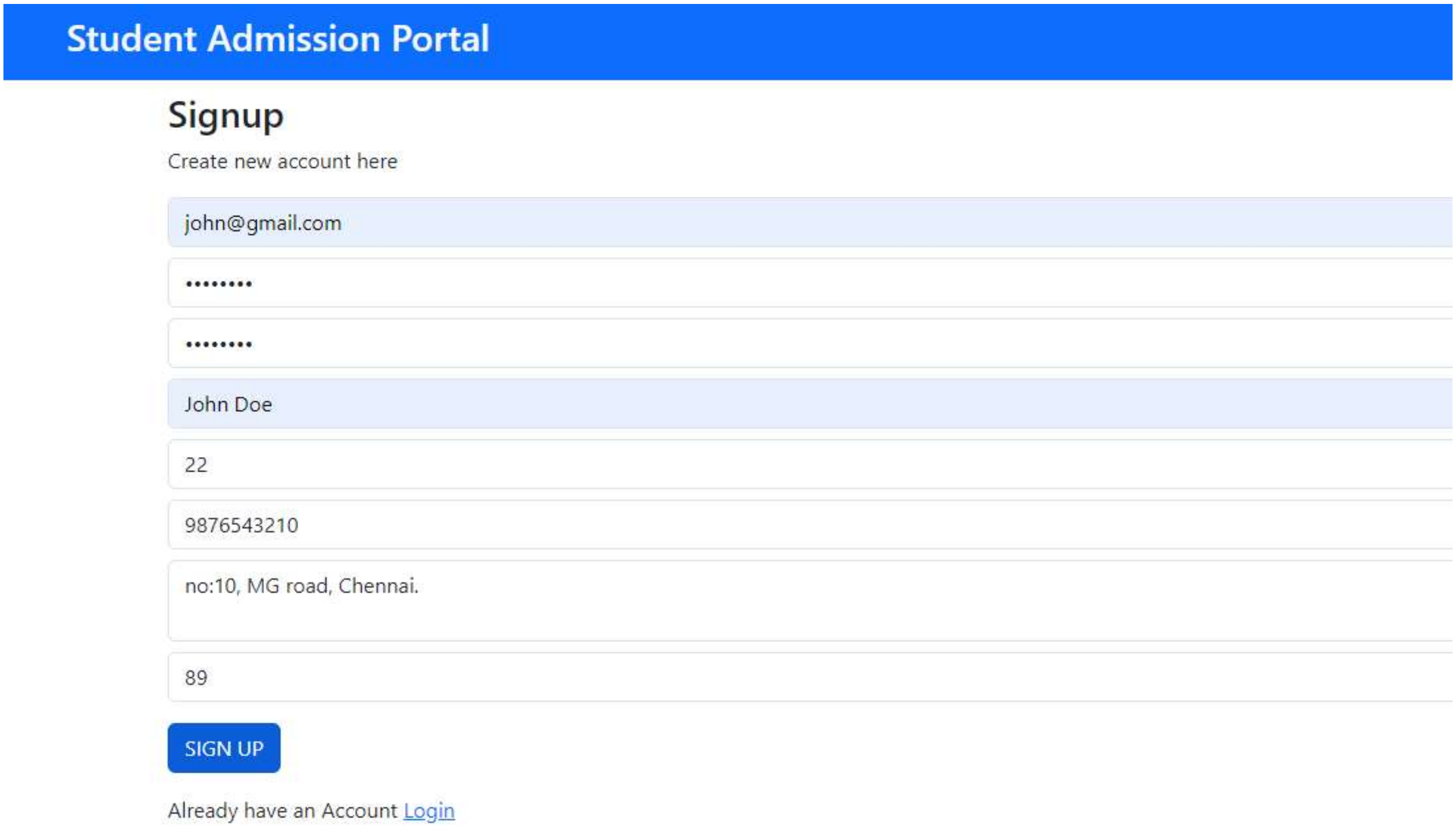
To test the application: **Run -> Test**.

Note: Your application should be running in port 8000 before you start the test.

Signup Component

- This page enables students to signup in this portal.
- Get all the inputs from the user. On submit validate the inputs.
- On invalid input display the error message in a paragraph with id 'errorMessage'.
- Make sure that the given email is not registered already.
- Finally store user details to json db using the URL specified in the code.

Note: Drag preview images and open them new tab to get best view.



Student Admission Portal

Signup

Create new account here

john@gmail.com

••••••••

••••••••

John Doe

22

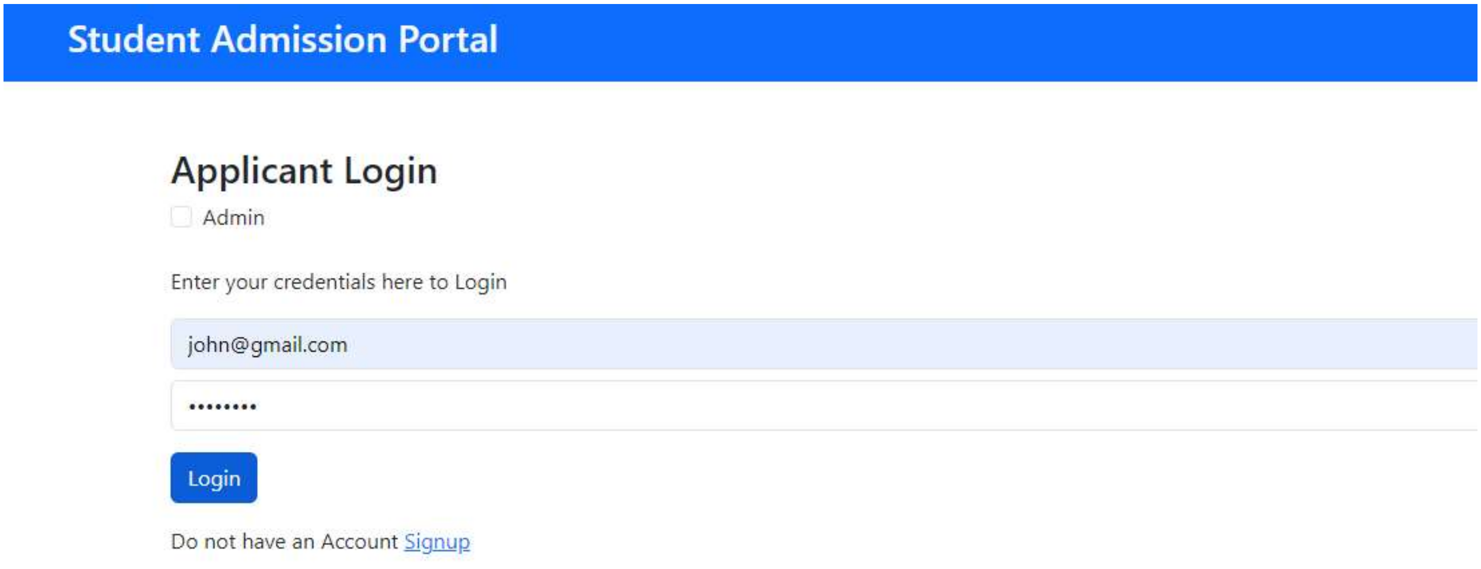
9876543210

no:10, MG road, Chennai.

89

Login Component

- This page should be used as login way for both applicants and admin by toggling the checkbox with id 'userType'.
- Verify the entered credentials with data in db.
- The api to get the user credential from the db is given in code.
- Redirect the admin to '/applications' on successful login.
- Redirect the applicants to '/apply' on successful login.
- Admin credentials- admin@abz.com , Admin@123 .



Student Admission Portal

Applicant Login

□ Admin

Enter your credentials here to Login

john@gmail.com

•••••••••

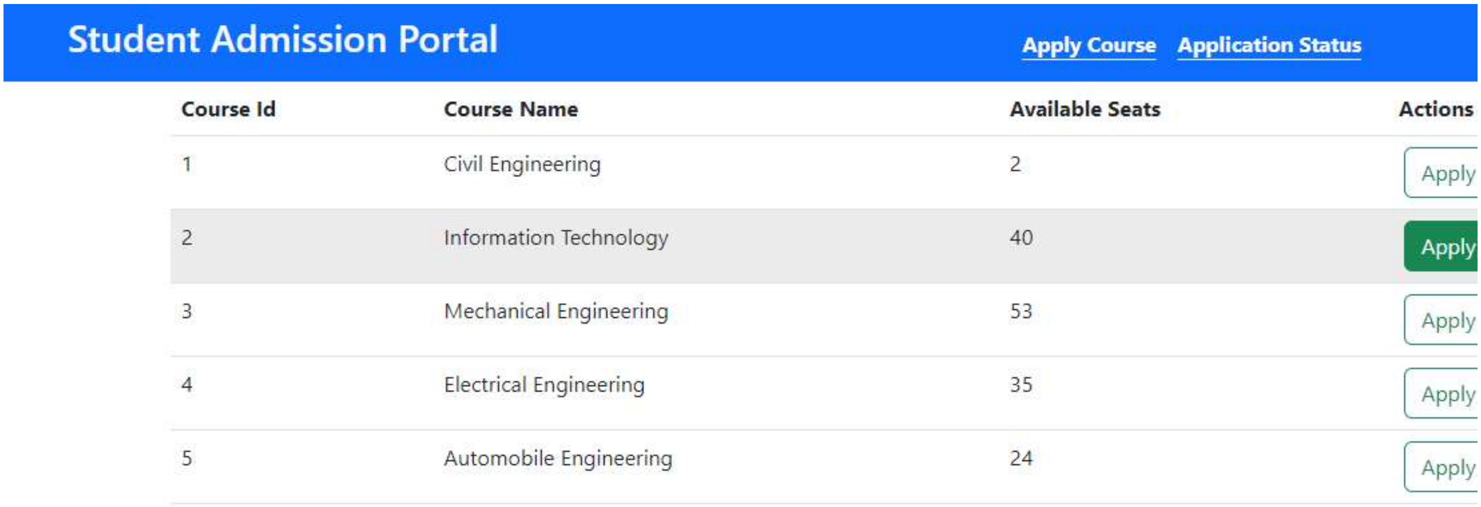
Do not have an Account Signup

Navbar Component

- This component is common for both applicants and admin.
- For admins, first nav item is the link to '/applications' page.
- Second nav item is the link to '/addseats' page.
- Third nav item should be username. Sample- 'User: Admin'.
- For applicants, first nav item is the link to '/apply' page.
- Second nav item is the link to '/status' page.
- Third nav item should be username. Sample- 'User: John Doe'.
- Both users should have Logout button in fourth nav item.

ApplyCourse Component:

- This page allows applicant to apply for courses offered by the college.
- Make use of 'getCourses' action function and display the courses in given table template.
- Clicking on apply button for any course should submit an application for the course with user details.
- The api and request body format are provided in the code
- On successful submission display the alert with success message along with application id.
- Make sure, user should submit only one application for each course.
- User should not be able to submit application to courses with zero seats.
- Get all applications of logged user by dispatching 'userApplications' action function.

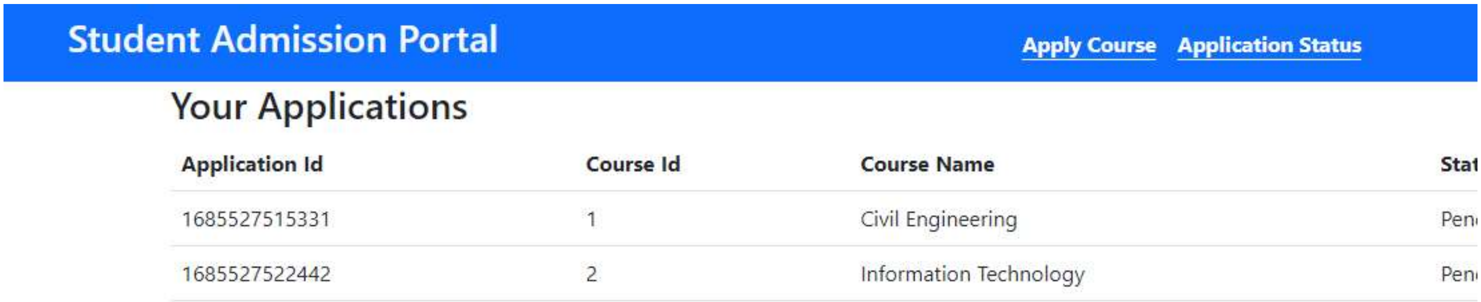


Student Admission Portal

Course Id	Course Name	Available Seats	Actions
1	Civil Engineering	2	Apply
2	Information Technology	40	Apply
3	Mechanical Engineering	53	Apply
4	Electrical Engineering	35	Apply
5	Automobile Engineering	24	Apply

ApplicationStatus Component:

- This page is used to display status of applied applications.
- Get all applications of current user by dispatching 'userApplications' action function.
- Display the application details in given table.

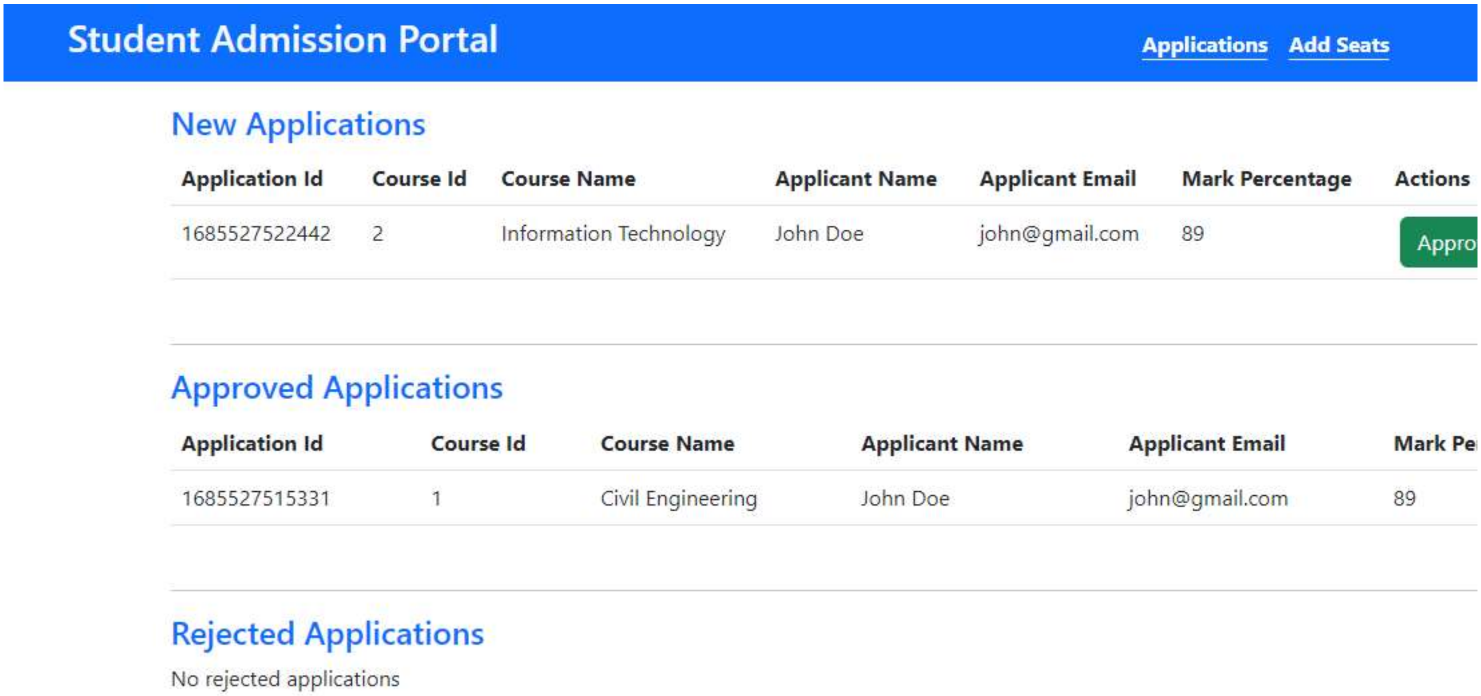


Your Applications

Application Id	Course Id	Course Name	Stat
1685527515331	1	Civil Engineering	Pen
1685527522442	2	Information Technology	Pen

ViewApplications Component:

- This Page is used by admin to approve or reject user application.
- Get all the applications of applicants by dispatching 'getApplications' action.
- In table with id 'newApplicationsTable' display new applications with status 'Pending'.
- Clicking on Approve should set the application status as Approved. For this operation use the action 'modifyApplicationStatus'.
- Approving an application should also decrement one seat in particular course.
- Admin should able to approve only finite number of applications for available seats.
- When a course seats are emptied, no more applications can be approved.
- Clicking on Reject should set the application status as Rejected. Use 'modifyApplicationStatus' action.
- In table with id 'approvedApplicationsTable' display the application details of approved applications.
- In table with id 'rejectedApplicationsTable' display the application details of rejected applications.



Student Admission Portal

New Applications

Application Id	Course Id	Course Name	Applicant Name	Applicant Email	Mark Percentage	Actions
1685527522442	2	Information Technology	John Doe	john@gmail.com	89	Appro

Approved Applications

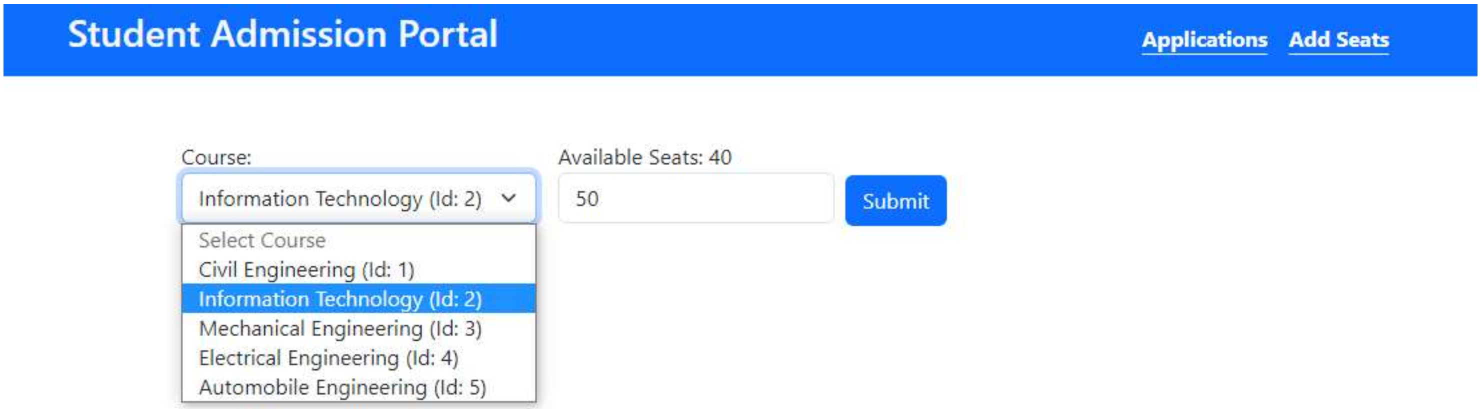
Application Id	Course Id	Course Name	Applicant Name	Applicant Email	Mark Pe
1685527515331	1	Civil Engineering	John Doe	john@gmail.com	89

Rejected Applications

No rejected applications

AddSeats Component:

- This page is used by admin to modify seat count of any particular course.
- Get all course details by dispatching 'getCourses' action.
- Populate all course name with course ids as options for select box.
- Sample option- Civil Engineering (Id: 1)
- Selecting any option should display current available seats of that course inside the label with classname 'availableSeats'.
- When admin enters new seat count and submits, seat count should be changed in db.
- For this operation dispatch the action 'addSeats'.



Student Admission Portal

Course: Information Technology (Id: 2)

Select Course
Civil Engineering (Id: 1)
Information Technology (Id: 2)
Mechanical Engineering (Id: 3)
Electrical Engineering (Id: 4)
Automobile Engineering (Id: 5)

Available Seats: 40

50

Slice.js

- This page contains action function to perform asynchronous data operations.
- Complete those functions where api's and request bodies are given in the code.
- Write the required reducer functions to save the data returned by action function into the store.

Software Instructions

The question(s) requires **Node 18 LTS or above**.

- Download & Install Node.JS

Git Instructions

Use the following commands to work with this project

run Copy

```
npm run json-server & npm start
```

test Copy

```
npm test
```

install Copy

```
sudo apt-get install psmisc && npm install
```