

Fullstack Challenge

Event Management application

Develop and Enhance a pseudo Front-end web app to manage events.

You are expected to upgrade a Web app written in Angularjs, Angular, React and NodeJS stack, MongoDB should be used as the database.

The source code for this application has been provided along with a description of the tasks needed to accomplish.

To run skeleton code use:

Angular 2:

ng serve

Angular 1:

grunt serve

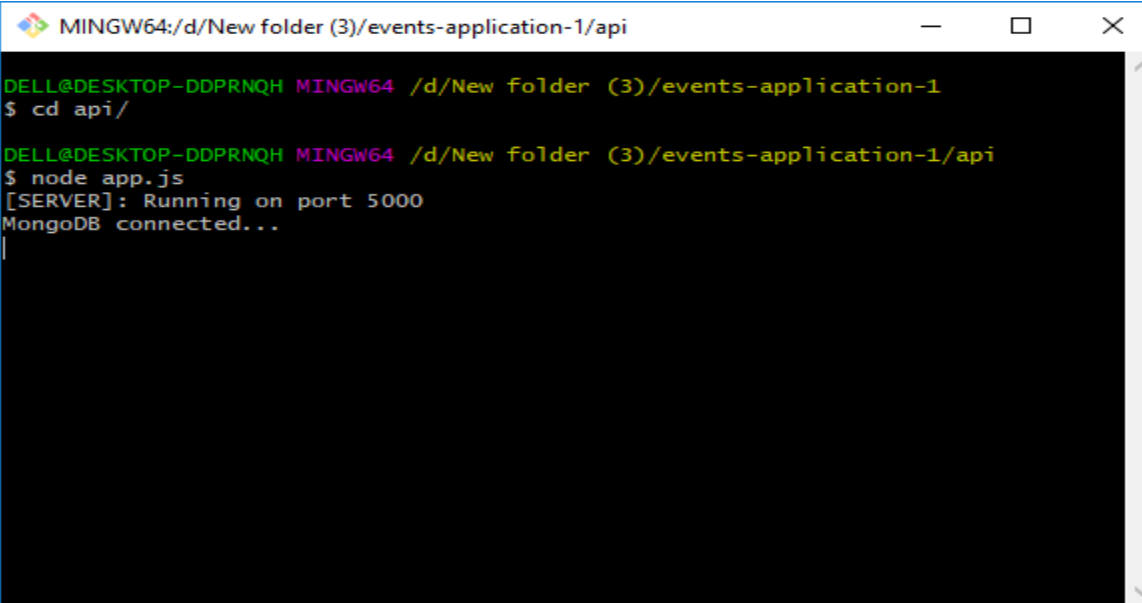
React :

npm start

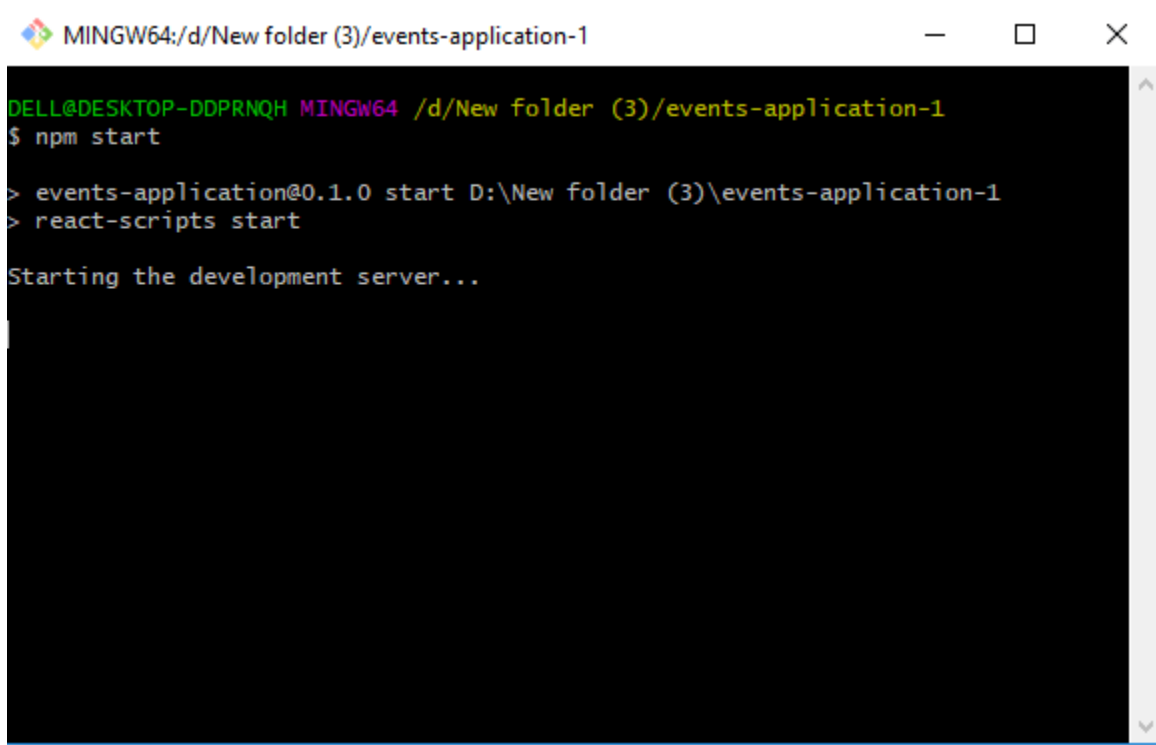
node [backend]: Path : projectName/api/app.js

node app.js

Sample Screenshot :



```
MINGW64:/d/New folder (3)/events-application-1/api
DELL@DESKTOP-DDPRNQH MINGW64 /d/New folder (3)/events-application-1
$ cd api/
DELL@DESKTOP-DDPRNQH MINGW64 /d/New folder (3)/events-application-1/api
$ node app.js
[SERVER]: Running on port 5000
MongoDB connected...
```



```
MINGW64:/d/New folder (3)/events-application-1
DELL@DESKTOP-DDPRNQH MINGW64 /d/New folder (3)/events-application-1
$ npm start

> events-application@0.1.0 start D:\New folder (3)\events-application-1
> react-scripts start

Starting the development server...
```

****With the problem statement, we have also provided the skeleton of the solution which is incomplete. You are expected to understand the skeleton thoroughly and complete most of the tasks specified here.****

You are required to extend the application and build the following additional features:

[Minimum requirement] (level 0)

[Frontend Task]

Add View Events Component: It should consist of the following features:

- **Event Table:** The table should have four columns, Event Name, Type, Length, and Views. The table contains all the created events. Also, the table should contain event's cost in the table. UI for the same needs to be created.
- Proper REST API calls at (\add), (\update<id>) and (\delete<id>)
- **Add Event button:** On clicking **Add Event button**, the view should switch to **Add Event Component**.

[Backend Task]

- Create Endpoint to add an event
- Create an event to get all events

[Plus point] (level 1)

[Frontend Task]

Create Add Event Component: It should consist of the following features:

- **Event Name Input:** The user should be able to enter the event name in it. Should take a string input with length between 3 and 24 characters
- **Type Drop-down:** The user should be able to select the type of the event using it.
- **Length Input:** The user should be able to enter the duration of the event in it. Should take a numeric input with length between 10-1000.
- **Save Button:** The event should be added and should redirect to the **View Events Component** where the new event should be reflected along with the existing events.
- Sort and filter based on date, category and alphabetical order
- Implement additional features like tags bookmarks ratings
- Implementing bulk delete functionality
- Responsive web design for all the views
- Developing presentational components which are reusable
- Provide timestamp for add/modify activities on the home page
- UI layer is responsive. Mobile UI should match that of mock provided. Provide a breakpoint at 960px

[Backend Task]

- Create and endpoint to create an event
- Pagination of 20 items. After reaching the end the application should make another call for the database for the next 20 items.
- Implementation of Login/Logout functionality, on the home screen, including user authentication. Either use Google API for authentication or create a manual login system and store password in hash.
- Proper validations should be given if forms are used(messaging/alerts etc.,)
- Proper unit tests and code coverage for Crud functionalities

[Advanced] (level 2)

[Frontend Task]

Create Update Event Component: It should consist of the following features:

- **Event Name Input:** The user should be able to enter the new event name in it. By default, the input box should have the name of the event being edited.
- **Type Drop-down:** The user should be able to select the type of the event using it. By default, the type of the event should be selected in the drop-down, for the event being edited.
- **Length Input:** The user should be able to enter the duration of the event in it. By default, The input box should have the length of the event being edited.
- **Save Button:** The event should be edited and redirected to the **View Events Component** where the updated event should be reflected.
- Provide UI test cases using Karma, Selenium etc.
- Session and state management across the application
- Connecting the complete end to end flow through routing mechanism (includes login/logout)
- Acceptable scaffolding/asset management and code management by using gitlab

Create Delete Event Component

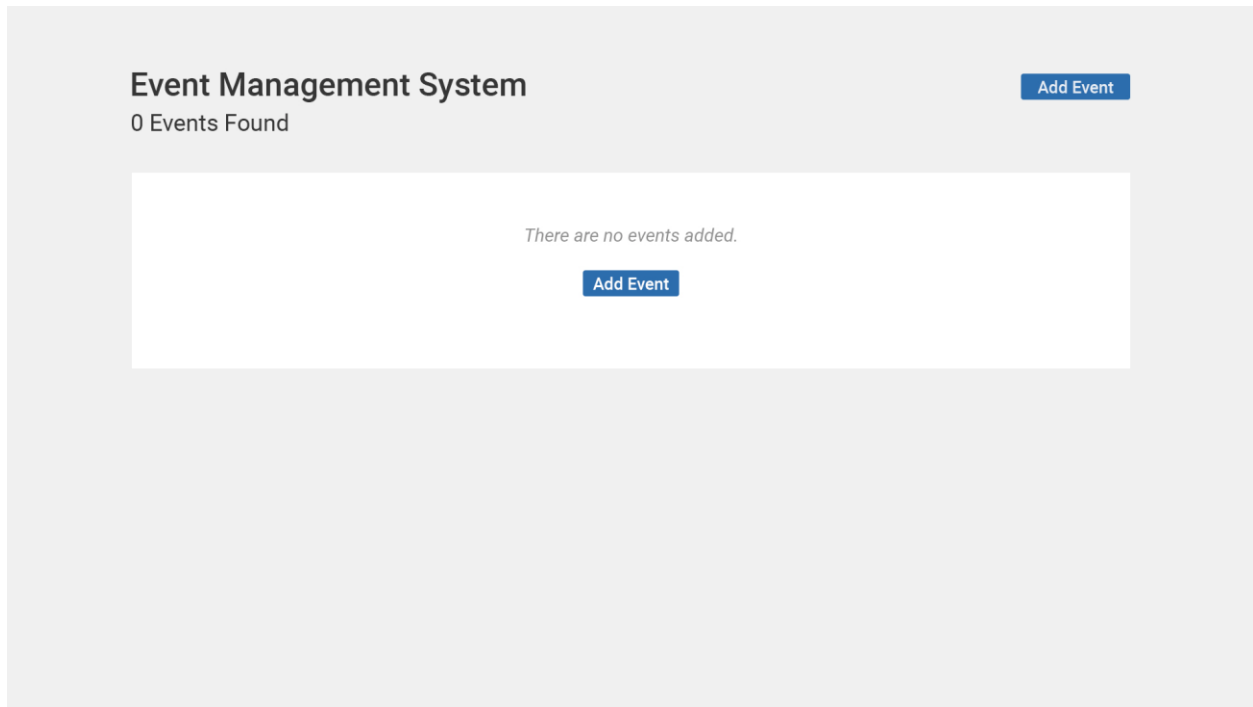
- Create an onclick events that deletes the corresponding event.

[Backend Task]

- Create an endpoint to update an event
- Use of git to manage source code. Create different branches for FrontEnd, Backend etc,
- Use of proper design patterns like Factory, Service, Singleton for all CRUD functionalities

[Mockup References]:

Homepage



Event Management System

0 Events Found

Add Event

There are no events added.

Add Event

On making a call at (“\add”)

Event Management System

Manage Event

Event Name

Type

Length (Mins)

Save

Event Management System

Events

Event Name	Type	Length (Mins)	Views
La Liga At Mad	christmas	90	2
Linkin Park collision	concert	60	1
Jio Premier by AR Rehman	concert	30	3
Euphoria - Thursday Live	concert	90	1
Broadway company Launch show	christmas	45	0

Add Event

On making a call at ("update\<id>")

Update Event

Event Name

Grimm 1

Type

Party

Length (Mins)

28

Save

Recommended Mockup

Web Mockup

Event Management System

133 Events Founds

Add Event

Event Name: La Liga at Mad
Event Type: Christmas
Length: 90 mins.
Views: 2000
Cost: 600/-

Update

Event Name: Linkin Park Collision
Event Type: Concert
Length: 30 mins.
Views: 8500
Cost: 5000/-

Update

Event Name: Jio Premier by AR Rahman
Event Type: Concert
Length: 2 Hours
Views: 4000
Cost: 1200/-

Update

Event Name: Euphoria - Thursday Live
Event Type: Concert
Length: 45 mins.
Views: 700
Cost: 750/-

Update

Event Name: Broadway Company Launch Show
Event Type: Christmas
Length: 60 mins.
Views: 350

Event Name: Comedy Friday
Event Type: Concert
Length: 45 mins.
Views: 600

Mobile Mockup

Event Management System

133 Events Found

ADD EVENT

Event Name: La Liga at Mad
Event Type: Christmas
Length: 90 mins.
Views: 2000
Cost: 600/-

Update

Event Name: La Liga at Mad
Event Type: Christmas
Length: 90 mins.
Views: 2000
Cost: 600/-

Update

Event Name: La Liga at Mad
Event Type: Christmas
Length: 90 mins.
Views: 2000
Cost: 600/-

Update

[Accepted Solution]

By the end you should have an application with the following features:

- You should be able to create an event with a name, type and length.
- You should be able to view all the events created.
- You should be able to edit the name, type and length of an event

[Solution Approach]

- No knowledge of CSS is required, though you are required to build the HTML and bootstrap forms
- Also you are required to configure routing for the application using react-router-dom which has already been installed.
- You should also be able to configure Redux with thunk. All these libraries have already been installed and stubs written to help you out.
- Also all the necessary files have been created so you don't need to create any extra files.
- For actions and reducers, you need to fill out the code files where necessary and connect your components to the Redux state.
- All backend code should be written in the app.js file found in the api folder

[Guide]

Code Skeleton:

TechStack :

Frontend: Angular, Angularjs, Reactjs

Backend: NodeJS

Database: MongoDB