Assignment 3

Things you will learn

In this assignment, you will learn the following:

- Angular
  - Components
  - Routing
  - Services
  - Pipes
- Socket.io
- Google Services
  - Text-To-Speech
  - Translate
- Deploy to Cloud
- Push to GitLab Repository
- Work in teams

Assignment Theme

Due to the huge increase in the number of activities on the app, the clients and developers have noticed that the app has become a little bit not responsive, and it takes some time to load every page. Also, several clients have requested two new features: text translation for non-English speakers and text-to-speech for those who have eyesight difficulties.

Therefore, Monash's IT department has decided to adopt Angular Framework to develop a single-page application that provides the best response time. To improve even further, they decided to make it downloadable so users can access it without needing a browser.
The IT department also decided to use Google services to implement the new two features.

Assignment 3 Specifications

Assignment 3 is due on the Friday of Week 11, October 13, 2023, at 11:59 pm.

In this assignment, you will focus on creating a Single Page Application (SPA) for the backend you developed in Assignment 2.

The SPA app for both students should:

- Use Angular Routing to navigate among the components using a navigation menu.
- Use Angular Service for database operations.
- Communicate with the backend server RESTful API that you have developed in A2 (API/V1) to perform all the database operations.
- Communicate with the backend server via Socket.io to execute Cloud Services

Team Tasks

1. Develop the angular routing module that lists all the Angular routes and components.
2. Add "Page Not Found" Component. This component should be displayed if an invalid URL (does not exist) is provided.
3. Add "Invalid Data" Component. This component should be displayed if invalid data is sent to the backend. In other words, if the server responds with error 400 due to invalid data, Angular should display the "Invalid Data" component.

Group 1 (Student #1)

1. Add the following Angular Components:
    1. **Add Category:** This component adds a new category to the database. It should redirect the client to the list categories component. The component should also redirect the client to the "Invalid Data" component if the server responds with status 400.

    2. **List Categories:** This component should list all categories in a table format. All the categories' names must be in upper case, which should be done using Angular Pipe. In other words, add an Angular Pipe to transform the category's name to all caps.

    3. **Delete Category:** This component takes the category ID and deletes it (and its events) from the backend database. To implement: list all categories and add a button that selects and deletes the required category (see the expected output recording). The component should also redirect the client to the "Invalid Data" component if the server responds with status 400.

4. **Display Category:** This component shows the details of a category. To implement: list all the categories in a table format and add a button for each row that selects and displays the required category. Use the same pipe to transform the category's name to all caps. The category detail page should also contain the list of events of the selected category. *Hint: you can add the button to the list categories component. Sample screenshot added here.*

5. **Update Category**: This component takes the category ID, name and description as input and sends them to the backend API to update. The component should also redirect the client to the "Invalid Data" component if the server responds with status 400.

2. Add a component that:
    1. takes a string in English as input
    2. sends it using Socke.io (not HTTP POST) to the backend that is listening to socket.io requests.
    3. The backend converts the text to speech using Cloud Text-to-Speech API
    4. The backend responds with the MP3 speech output of the string (path to generate output file) to the frontend component.

3. Add a component that:

    1. Shows the statistics of events and categories. It should list the number of events and categories that are currently in the database.

> *Hint: You might need to add an endpoint to the backend to serve the reqruied stats.*

4. Make your Frontend application to be Progressive Web Apps

5. Deploy your application to your VM on your GCP account

6. Update the backend endpoints to respond with an error (HTTP status) 400 if the user provides invalid data.

For Example, if the user sends the string "Categ$ry" as a category name, the backend should respond with status 400.

Hint: Use try{}catch(error){} to catch the errors

Group 2 (Student #2)

1. Add the following Angular Components:
    1. **Add Event:** This component adds a new event to the database. It should redirect the client to the list events component. The component should also redirect the client to the "Invalid Data" component if the server responds with status 400.

    2. **List Events:** This component should list all events in a table format. All the events' duration must be in hours and minutes format, which should be done using Angular Pipe. In other words, add an Angular Pipe to transform the event's duration to hours/minutes.

       For example, if the duration is 110 minutes, the component should display "1 hour(s) 50 minute(s)" in the duration field, and a pipe should do this transformation.

    3. **Delete Event:** This component takes the event ID, deletes it (and removes it from its categories) from the backend database. To implement: list all events and add a button that selects and deletes the required event (see the expected output recording). The component should also redirect the client to the "Invalid Data" component if the server responds with status 400.

    4. **Display Event:** This component shows the details of an event. Again, this can be implemented by listing all the events in a table format and adding a button for each row that selects and displays the required event. Use the Angular Pipe to transform the event's duration to hours/minutes.
       Hint: you can add the button to the list events component.

    5. **Update Event:** This component inputs the event ID, name and capacity and sends them to the backend API to update. The component redirects the client to the list events component. The component should also redirect the client to the "Invalid Data" component if the server responds with status 400. Sample screenshot over here.

1. Add a component that:
    1. takes a string in English as input
    2. sends it using Socke.io to the backend
    3. The backend translates to one of three languages of your choice using Google Translate services.
    4. The backend responds with the translated string to the frontend component.

3. Add a component that:

1. Shows the statistics of operations that happened on the database. It should list the number of create, update and delete operations on the database.

> *Hint: You might need to add an endpoint to the backend to serve the reqruied stats.*

4. Make your Frontend application to be Progressive Web Apps

5. Deploy your application to your VM on your GCP account

6. Update the backend endpoints to respond with an error (HTTP status) 400 if the user provides invalid data.

For Example, if the user sends the value -10 as a duration, the backend should respond with status 400.

Hint: Use try{}catch(error){} to catch the errors

Expected Output

File Structure

Group 1 (Category)

Sample Category Detail Page

Category List Add Category Update Category Delete Category Text To Speech Stats G1

Event List Add Event Update Event Delete Event Translator Stats G2

Object Id

651ce4589b67e997110fe4ca

Category Id

CME-8618

Name

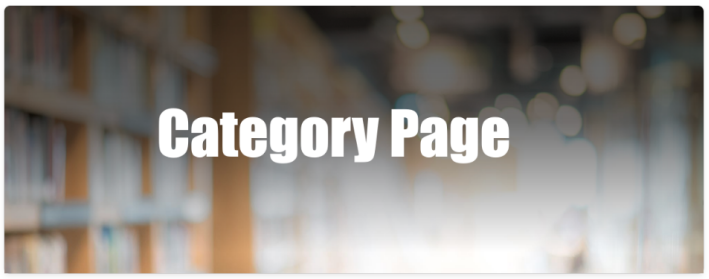
MELBOURNE

Description

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Created Date Time

04/10/2023 3:04 PM



Id	Name	Start	End	Duration	Is Active	Capacity	Tickets Available	Actions
EID-7777	AFL Grand Final	28/06/2023 6:30 PM	28/06/2023 8:30 PM	120	☑	500	500	View
EKR-5412	New Year Eve Celebrations	31/12/2023 11:00 PM	01/01/2024 12:50 AM	110	☑	1500	1500	View
EMX-7371	Music festival	13/12/2023 6:00 PM	13/12/2023 9:00 PM	180	☑	1500	0	View

Developed by Dream Team

Group 2 (Events)

Sample Update Event Page

Event Management App

Category List | Add Category | Update Category | Delete Category | Text To Speech | Stats G1

Event List | Add Event | Update Event | Delete Event | Translator | Stats G2

Select one of the events to update

Name

New Year Eve Celebrations

Capacity

1500

Update

Id	Name	Image	Created Date	Actions
EID-7777	AFL Grand Final	/event.png 		Select
EKR-5412	New Year Eve Celebrations	/event.png 		Select
EMX-7371	Music festival	/event.png 		Select

Marking Rubric

Summary

Team Tasks

Group 1(Student #1)

Group 2(Student #2)

What to submit?

Where?

Every student must submit assignment-3 to two places: **Moodle** and **Gitlab**.

For Moodle, you must ZIP your files into a single file and upload it to the assignment-3 submission link that can be found on Moodle-->Assessment.

For GitLab, you will be provided with a group repository, and you should commit/push your project to it. We are expecting at least three non-trivial commits to your group repo. See the marking rubric for more details.

What??

1. A document in PDF, DOCX, or MD format that contains:
    1. Steps to run your applications
    2. [Optional] unsolved bugs and issues
    3. this document must be placed in the root folder of your project
3. The source code of the project

Late Submission