**ICP7**

**Library Management System**

**Mean Stack**

**ICP Group 4**

**Name: Venkata Mahesh Mokkapati**

**Email: vmzwn@umsystem.edu**

**Partner Name: Sailaja Narra**

**Partner email:** [**sntnn@umsystem.edu**](mailto:sntnn@umsystem.edu)

**Partner Repo:** [**https://github.com/UMKC-APL-WebMobileProgramming/ICP7-sailajanarra**](https://github.com/UMKC-APL-WebMobileProgramming/ICP7-sailajanarra)

**My report, video and Source code links:**

**Report:** [**https://drive.google.com/file/d/1U627vxKu3r9UmQp2MGyKo0y9Pdh003XX/view?usp=sharing**](https://drive.google.com/file/d/1U627vxKu3r9UmQp2MGyKo0y9Pdh003XX/view?usp=sharing)

**Video:** [**https://umsystem.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=6e482364-c73c-4d98-8f7d-adbe003a55fb**](https://umsystem.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=6e482364-c73c-4d98-8f7d-adbe003a55fb)

**Source:** [**https://github.com/UMKC-APL-WebMobileProgramming/ICP7-Mahesh68**](https://github.com/UMKC-APL-WebMobileProgramming/ICP7-Mahesh68)

**This task documentation (proper comments), Video, validation and responsiveness has also been handled.**

In this task I learnt about connecting to database and full framed application. There is source code given which is having front end components and some node js configuration.

So here on click of one of the list elements, we can edit and also delete the element and store the state using mongo DB.

**Database connection:**

Installed monogo DB locally in my machine and installed node modules after downloading the content from the source.

Created some db path and used that folder for storing DB logs.

Run mongod.exe file to connect to database. Now open mongodb compass and create a new connection with the database url mentioned in the app.js file

mongodb://localhost/mean-angular6

Here is the connection in the app.js file

Text

Description automatically generated

DB connections in the compass

Table

Description automatically generated

List of dbs are as shown above

Below are the collections in the connected database

Graphical user interface, text, application

Description automatically generated

Here is the list of documents inside a collection

Graphical user interface, text, application, email

Description automatically generated

We can create documents manually here in the compass or else using command prompt. And these will reflect in the front end application.

**Now connecting front end to backend(node js)**

As the controller in the front end will handle any http requests that are triggered by the user, there is also a controller in the backend that handles the client side requests and give response back to the client.

Here is the step by step connection that’s happening when a request is made from the front end:

Appservice.ts

Text

Description automatically generated

App.js file which is having same api url:

Text

Description automatically generated

So the api router has all the CRUD operations which is book.js

Text

Description automatically generated

If the url is “/” and if a get request is reached to server then the server will return this response back to the client.

Here the logical code for the update and delete methods are same as get and save. Only difference is the http calling methods.

Text

Description automatically generated

Here once user clicks on any element URL will also get updated along with the ID of the book that is clicked.

So to do some manipulations for that book we can make use of the params in the URL.

Params is an inbuilt method to both angular and node which gives the values of query parameters in the URL. So navigation and handling URL is the main thing that has to be handled because based on navigation and url change different components are getting loaded and CRUD operations are getting called.

Book create component

Text

Description automatically generated

Text

Description automatically generated

Book edit component

Text

Description automatically generated

Here the list of form fields are created using form builder which is a template driven form.

Using mat module handled the input required field errors which is as shown below:

Text

Description automatically generated

So if any form fields are not specified then there is a mat error displayed to the user

A screenshot of a computer

Description automatically generated with low confidence

List of books

Graphical user interface, text, application

Description automatically generated

When user clicks on one of the element, book details component will get loaded and here is the page:

Graphical user interface, website

Description automatically generated

So here are the two buttons for edit and delete that book from the database:

For editing bookedit component is used which loads the form as explained above

Text

Description automatically generated

Put request in the node server will get called after successful submission of the book edit.

Text

Description automatically generated

Here async and await methods are used to make the javascript ajax call synchronous . Here the response will be a promise so by the time we get response this line of code will get executed and resolved.

Hence using await function this method will wait for the promise to get resolved which means successful updation to DB and fetch the response from the backend.

Here updated the title in the book and saved.

A screenshot of a computer

Description automatically generated with low confidence

On successful update to DB, front end will get navigated to book details page with the updated values

Graphical user interface, website

Description automatically generated

A screenshot of a computer

Description automatically generated with low confidence

Graphical user interface, website

Description automatically generated

Graphical user interface, text, application, chat or text message

Description automatically generated

If user want to create a new book, this add book button will serve the purpose.

On click of add button we will get the details of the book with the form fields editable

Graphical user interface

Description automatically generated with medium confidence

Graphical user interface, application, PowerPoint

Description automatically generated

On click of delete button, delete method will get called and in the DB using findOneAndRemove method will get called and removed from the mongo database.

On successful removal from the DB, page will get navigated to the book component an updated books list will be shown.

Graphical user interface, application

Description automatically generated

Here are responses in the network tab

Graphical user interface, text, application

Description automatically generated

**Responsive page:**

Graphical user interface, website

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence