Angular Dashboard

09 Gaurav Khalapurkar
11 Aakash Gupta
32 Saksham Patil
38 Sanket Pednekar

Problem Statement

We students have to navigate between various platforms when searching for the VAP related materials. Hence, it creates a distraction while we are studying.

Technologies Used

- Angular
- OneDrive API
- Node.js
- Express
- MongoDb
- Heroku

Frontend

Description

With a modern design, we have created a user friendly UI so that the students can learn the enrolled VAP with one click.We have kept all the things at one place so that it becomes easier to navigate between various things such as syllabus, notes, code, videos of the recorded sessions and schedule of the ongoing/upcoming lectures.

Components

We have created multiple components in order to enhance the user experience. Some of the main components include Signup Component, Login Component, Navbar Component and Dashboard Component.

The other components created are the different pages in the dashboard component which are shown in the same using routing.

Services

We have created a global service in which all the global variables and functions for the components to access are declared.

Route Guard

A route guard is created and added on route which checks if the user is already logged in status. This is done by checking if there is any jwtToken stored in the local storage and then sends a get request to the verify token API, which then verifies if the token is valid or not, If the token is valid then the user gets logged, otherwise he/she is redirected to the login page.

Backend

Description

We have hosted our server on Heroku in order to access it everywhere and have created a User Model in MongoDB in which we are storing all of the users' details like name, email and password. Furthermore, the password is stored after encryption to ensure security.

We have also created 3 different Routes for different requests such as:

1. Register Route in order to create a new user,
2. Login Route in order to check if the user exists and return jwtToken in order to log the user in.
3. Verify Token Route in order to check if the jwtToken provided is valid or not.

Model

A User model is created in mongoDB in which we are storing all of the users' details like name, email and password (encrypted).

Server Routes

API Route → https://angular-dashboard-vap.herokuapp.com

Login Route → /api/v1/user/login

Register Route → /api/v1/user/register

Verify Token Route → /api/v1/user/verify