

Event Organisation Portal using MEAN Stack

1st Rashmee Wadagave
Department of Technology
Shivaji University
Kolhapur,Maharashtra,India

2nd Srushti Karoshi
Department of Technology
Shivaji University
Kolhapur,Maharashtra,India

3rd Pranali Ravan
Department of Technology
Shivaji University
Kolhapur,Maharashtra,India

4th Rutuja Santikar
Department of Technology
Shivaji University
Kolhapur,Maharashtra,India

5th Rashmi Deshmukh
Department of Technology
Shivaji University
Kolhapur,Maharashtra,India

Abstract—With the tremendous use of the internet nowadays makes web development more demanding and also makes challenging at the same time for developers. These challenges have been overcome by the latest technologies. One of these technologies is the MEAN stack. This paper examines the usage of MEAN stack components (MongoDB, Express Js, Angular and NodeJs) and develops a web application. Prominently this has presented end-to-end use of MEAN stack components in a web application that is from implementation to its working. MEAN stack with its efficient features provides fast, scalable and notable results for developing systematic mobile and web application. Using MEAN stack, portal "Event Organisation" is created as an institute level portal to maintain connection between students and institute.

Index Terms—MEAN stack, web application, MongoDB, Express Js, Angular, Node Js, Auth Guard, JWT(Json Web Token)

I. INTRODUCTION

The last few years have seen a tremendous growth in Internet usage, which makes web development in demand and also makes it challenging year by year. These challenges have been managed by upcoming new technologies. One of the technologies is the MEAN stack, this makes web development fast and easy to handle. Due to the pandemic, everything was online including schools, colleges and universities. Moreover, after the pandemic things have remained as it is with no change. That makes technologies more in demand over the internet.

In this paper, present a web application called "Event Organisation", which is an Event organizing site that allows the student to interact with institutes. This web application is a MEAN stack development which makes the application fast and easy to develop.

In this application, many other features can help students gain knowledge. Only college-level details are restricted to that institute student and can be accessed after undergoing login credential authentication which is provided by Angular Auth guard and JWT at the backend.

II. RELATED WORK

When the pandemic started, the situation got upside down where there was a loss of communication between students and

the institute and also loss in the current affairs happening in the institute. Allowing students to stay in touch, get to know about current affairs and gain knowledge about the projects done by other students and teachers was done using modern technology MEAN stack [1].

Dynamic web application using a modern tool known as MEAN is a powerful combination of four technologies that make both client and server-side components interactive in a web application [2]. The major advantage of using this technology together is that, all rely on JavaScript on the server-side Typescript on the client-side of the application [3].

A. Angular

Angular is the next mean stack framework used for web development. It is maintained by Google as an open-source web application framework. Emendation of Angular into a HTML page with a JavaScript tag. Additional to it extends HTML attribute with directives and after it binds data with HTML Expression. Angular makes possible of two-way binding between front-end and back-end. This makes the developer develop template in HTML easy and also in terms of timeline. An angular system can take care of the routing, which means it is easy in Angular to switch from one view to another. Dependency injection or design pattern and Data binding are two main structures of Angular which increase flexibility and modularity in application [4].

B. Node Js

Node Js is the next cross-platform, open-source runtime environment used for the development of networking and server-side applications [5]. Applications in Node Js are written using JavaScript language and can be run within the Node Js runtime environment. Node Js uses V8 engine of google chrome for providing runtime environment to execute JavaScript. Node is a major part of backend API services.

C. Express Js

Express Js is one of the latest and widely used web frameworks in the Node.js environment. Express Js is a web server built on Node.js [6]. Express Js allows you to handle

various networking devices like Routes, Server, and I/O stuff very easily. The simple command to install Express in Linux is `npm install express`

D. MongoDB

MongoDB is simple and is used for storing databases. MongoDB is an open-source document database and a NoSQL database. It is written in C++. MongoDB provides support for rich queries to retrieve data from the database. It supports the JavaScript i.e., server-side execution allowing any developer to use an only one programming language for both client/server-side code [7].

Sr.No	Open Source	Scalability, Efficiency	Compared to other similar Systems/ Framework
Angular	✓	Enables maintainable and extensible code.	Angular is better than React because of data binding, material design contained in it.
Node Js	✓	Provide load balancing that makes it scalable and efficient.	Compared to python, node Js is best as it has asynchronous programming
Express Js	✓	Lightweight and leveraging to build large-scale application.	Build- middleware is only in express Js and not in tinyhttp.

TABLE I
MEAN STACK

III. SOFTWARE REQUIREMENT

The main goal of our web application is to make use of full-stack technologies and show how powerful it can be when used together in terms of performance and cost [2]. MEAN stack has been the most popular web technology for building a highly responsive web application.

Using these technologies, developing an educational web application that is very worthwhile, beneficial and provides a great platform for students as well as for teachers. The Windows NT platform or Linux is required with an Intel 32 bit processor. This uses Visual Studio Code as an Integrated Development Environment (IDE). Other installations for the project are Npm (node package manager) which is used for managing modules i.e., to install all development dependencies of the project. After creating a folder for the server, installed mongoose, JSON web token, and body-parser, handlebars. All the information about these dependencies is written in the package.json file. AngularCLI (Command-line-interface) tool is used to initialize, develop and maintain the Angular application directly from the command shell. Installation of Authguard for authentication explicitly in this project. For the attractive GUI (graphical user interface) bootstrap is used to design Templates.

IV. PROPOSED SYSTEM

A. Implementation

This website development is RESTful API, where every implementation is done independently between client side and server side with each having different port number and address as shown in "Fig.1". For implementation specification used are as follows:

- Angular version 12
- Node Js version 14.17.3
- Express Js version 4.17.12
- Mongoose version 6.1.4
- MongoDB compass version 1.30.1

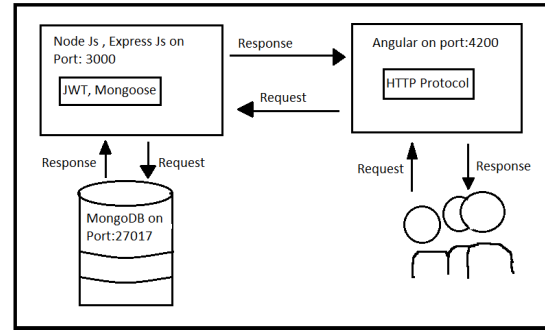


Fig. 1. Architecture

1) *Implementation of client side* : For the implementation of client side Angular is used. It is the latest version which uses typescript as its language. It's a single page application, implemented using the combination of Angular and bootstrap as a web design. This application provides all the details regarding each route, these routes are in components that are:

- about us
- events
- special events
- project
- login

To this more additional functions are used are Auth guard to recognize authorized persons and auth services which is a feature to the application that performs some specific function. Auth service is implemented in Auth guard component typescript file using dependency injection. After login Auth guard recognizes the person and decides where to displays the special event, project tab "Fig 2.". If the authentication is true, it displays these tabs if not it again comes to the login page.

For authentication auth service is used which identifies the token, this token is generated by JWT at back-end. In this service back-end is connect through URL.

2) *Implementation of server side* : Server side consist of API and MongoDB. According to the request node Js fetches the data from mongoDB which is done with the help of mongoose. [8] Here in an application, MongoDB helps to perform CRUD—Create, Read, Update, and Delete operation. This CRUD operation allows data to modify according to the

```
import { Router } from '@angular/router';
import { AuthService } from '../auth.service';

@Injectable({
  providedIn: 'root'
})
export class AuthGuard implements CanActivate {
  constructor(private _authService: AuthService,
    private _router: Router) { }

  canActivate(): boolean {
    if (this._authService.loggedIn()) {
      console.log('true')
      return true
    }
    else {
      console.log('false')
      this._router.navigate(['/login'])
      return false
    }
  }
}
```

Fig. 2. Auth guard

client-side which can be added or deleted events, log in ids, and projects.

a) *Implementation of MongoDB:* The schema is created in express Js application, which contains two schemas one for login and other for events. For login schema, logical schema contains prn and password “Fig 3”.

```
const mongoose = require('mongoose');
const Schema = mongoose.Schema;
const userSchema = new Schema({

  prn:{
    type: String,
    required: true,
  },
  password:{
    type: String,
    required: true,
  }
});
```

Fig. 3. Login Schema

For event schema, logical schema contains name, description, teacher “Fig 4”. After creating the schema, its exported to the main Js application [9]. Mongoose as Object Data

```
const sch = new Schema({
  name: {
    type: String,
  },
  description: {
    type: String,
  },
  Teacher: {
    type: String,
  }
});
const eve = mongoose.model('events',sch,'events');
const spe = mongoose.model('special',sch,'special');
const log = mongoose.model('login', userSchema,'login');
module.exports={
  eve,spe,log
}
```

Fig. 4. Event Schema and exporting the schemas

Modeling (ODM) a library of mongoDB which helps node Js to manage database according to the schema. According to the schema declared data is stored in mongoDB compass in “Fig 5”, “Fig 6”.

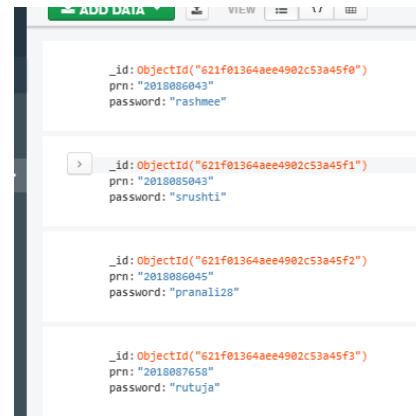


Fig. 5. Login database according to schema

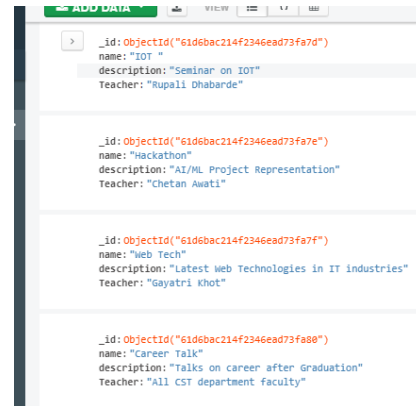


Fig. 6. Event database according to schema

b) *Implementation of API:* Application Programming Interface (API) acts as an intermediate between two applications. Here intermediate is node Js and express Js. Routes are present in express Js application which performs the required task of fetching data from mongoDB as requested from client- side. Extracting data can be possible after successful connection between mongoDB and express as shown in “fig 7”. Now

```
const db = "mongodb://localhost:27017/Events";
mongoose.connect(db,{useNewUrlParser:true,useUnifiedTopology: true },
function(err){
  if(err){
    console.error('Error! ' + err)
  } else {
    console.log('Connected to mongodb')
  }
});
```

Fig. 7. Connection of MongoDB with Express Js

routes can access the data from mongoDB after writing the queries for the specific data fetch. These queries are in “fig

8". When the client side sent request to route is events, events get the request and give the response according to the query

```
router.get('/events', function(req, res, next) {
  eve.find({}).then(function(events) {
    res.send(events);
  }).catch(next);
});
```

Fig. 8. Route Events

In express Js application, JWT is also present as a middleware for the authentication purpose. JWT in there is main used for login credential. JWT successful generates the token shown in “fig 9” after authenticating the requested body i.e., data entered by used and the database data, this token is verified for respective routes which are special events, projects. Verified token are send to auth service for further process.

```
const jwt = require('jsonwebtoken')
router.post('/login', (req, res) => {
  let user = req.body.prn
  let pwd = req.body.password
  const data = log.findOne({prn:user, password:pwd})
  if(data) {
    let payload = {subject: 1}
    let token = jwt.sign(payload, 'secretKey')
    res.status(200).send({token})
  }
  else {
    res.status(401).send('Invalid Password')
  }
});
```

Fig. 9. Route Events

B. Working

Web application "Event Organisation" is an educational site and its work is shown in "Fig.10". On top of the homepage, the header contains a global event, a local event, projects and login opinion. The global event shows students events that will occur in the institute, for this login is not needed as its open for all other college students. In local events students need to log in to access it, this page also shows events that are going to occur in college but it has a limitation that these events are only up to that institute student. Both global and local events contain further sections that describe the department. So, it makes it easy for students to find events according to their perspective and can attend that.

To display content in the project tab, students must be logged to the page as it allows them to show to that respected institute. Here the contents have sections that help students to add projects according to the department. In this tab, students can only add projects that have done so that other fellow members can get knowledge. But students don't have the authority to delete or edit any other student's project. Only institute teachers have the authority to delete and modify content. The login option on the header allows students to

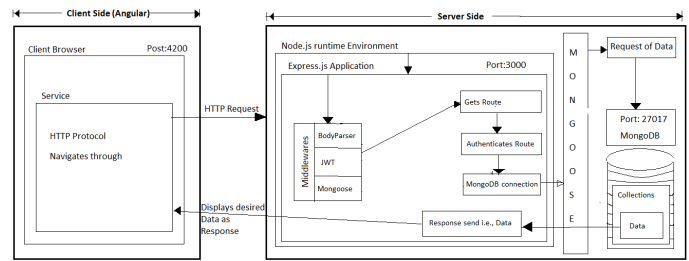


Fig. 10. Workflow of web application

directly log in to the page directly apply login to local event as well as Project tab.

CONCLUSION

This work has shown that the web application "Event Organisation" is useful for every institute to remain in contact with students in various ways. With the use of the latest technology MEAN stack development of this web application made it easy. Along with these technologies the upcoming challenges have been overcome in time and made the developer less challenging. The scalable deployment of web-service using MEAN stack also eminence further exploration as the technology matures.

REFERENCES

- [1] M. P. S. Mr. Ninaad Nirgudkar, "The mean stack," *International Research Journal of Engineering and Technology (IRJET)*, vol. 4, 2017.
- [2] . Ramappa and D. Bein, "Musiqglobe.fm using mean stack," 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC), pp. 661–664, 2018.
- [3] A. J. Poulter, S. J. Johnston, and S. J. Cox, "Using the mean stack to implement a restful service for an internet of things application," 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT), pp. 280–285, 2015.
- [4] "Angular documentation." [Online]. Available: <https://angular.io/docs>
- [5] "Node js documentation." [Online]. Available: <https://nodejs.org/en/docs/>
- [6] "Express js documentation." [Online]. Available: <https://expressjs.com/>
- [7] "Mongodb documentation." [Online]. Available: <https://docs.mongodb.com/>
- [8] K. K. Logesh M, "Mean stack web development," *International Research Journal of Engineering and Technology (IRJET)*, vol. 7, 2020.
- [9] Munawar, Ghifari. (2018)., "Analisa Penerapan MEAN Stack Dalam Pengembangan Web Berbasis Deklaratif," *9th Industrial Research Workshop and National Seminar*.
- [10] Patel, Viral, Daanyaal Kapadia, Deval Ghevariya, and Shiburaj Pappu. "All India Grievance Redressal App." *Journal of Information Technology and Digital World* 2, no. 2 (2020): 91-99.
- [11] Kumar, Dinesh, and Dr S. Smys. "Enhancing Security Mechanisms for Healthcare Informatics Using Ubiquitous Cloud." *Journal of Ubiquitous Computing and Communication Technologies (UCCT)* 2, no. 01 (2020): 19-28.
- [12] Ityala, Saiteja, Oshin Sharma, and Prasad B. Honnavalli. "Transparent Watermarking QR Code Authentication for Mobile Banking Applications." In *International Conference on Inventive Computation Technologies*, pp. 738-748. Springer, Cham, 2019.