

## Exercise 8 – NodeJS

Note: Performed Activity using MEAN Stack

### Node Code

```
var express = require("express");
var validator = require('validator');
app = express();
app.use(express.json());
bodyparser = require("body-parser");
mongoose = require("mongoose");
var cors = require('cors')
app.use(cors())
mongoose.connect("mongodb://localhost:27017/mydb", {
  useNewUrlParser: true,
  useCreateIndex: true,
  useUnifiedTopology: true
});

const schema = mongoose.Schema({
  email: {
    type: String,
    unique: true,
    required: true,
    trim: true,
    lowercase: true,
    validate(value) {
      if (!validator.isEmail(value)) {
        throw new Error("Email is invalid!");
      }
    },
  },
  password: {
    type: String,
    required: true,
    minlength: 7,
    trim: true,
    validate(value) {
      if (value.toLowerCase().includes("password")) {
        throw new Error('Password cannot contain "password"');
      }
    },
  },
},
```

```
});

const User = mongoose.model("User", schema);

app.post("/login", (req, res) => {
  res.header("Access-Control-Allow-Origin", "*");
  const _id = req.body.email;
  console.log(req.body);
  User
    .findOne({ email: _id })
    .then((user) => {
      console.log(user);
      if (user.password !== req.body.password) {
        return res.status(405).send();
      }
      res.status(200).send(user);
    })
    .catch((e) => {
      res.status(404).send();
    });
});

app.post("/register", (req, res) => {
  res.header("Access-Control-Allow-Origin", "*");
  console.log(req);
  const user = new User(req.body);
  user
    .save(req.body)
    .then(() => {
      res.status(201);
      res.send(user);
    })
    .catch((e) => {
      res.status(400);
      res.send(e);
      // console.log(e)
    });
});

app.listen(3000, "localhost", function () {
  console.log("Hello World!");
});
```

## Auth Service (used for API Calls)

### auth.service.ts

```
import { HttpClient } from '@angular/common/http';
import { Injectable } from '@angular/core';

@Injectable({
  providedIn: 'root'
})
export class AuthService {

  constructor(private http: HttpClient) {

  }

  login(model){
    return this.http.post('http://localhost:3000/login', model);
  }

  register(model){
    return this.http.post('http://localhost:3000/register', model);
  }
}
```

## Login Component

### login.component.html

```
<form #loginForm="ngForm" class="form-signin text-center" (submit)="login()">
  <h1 class="h3 mb-3 font-weight-normal">Please sign in</h1>
  <label for="inputEmail" class="sr-only">Email address</label>
  <input #Email ngModel type="email" name="email" id="inputEmail"
  [(ngModel)]="model.email" class="form-control" placeholder="Email address"
  [pattern]="emailPattern" required>
  <br>
  <label for="inputPassword" class="sr-only">Password</label>
  <input #Password ngModel type="password" name="password" id="inputPassword"
  [(ngModel)]="model.password" class="form-control" placeholder="Password" required>
  <br><br>
  <button [disabled]="!loginForm.valid" class="btn btn-lg btn-success btn-block"
  type="submit">Sign in</button>
  <br><button class="btn btn-danger btn-block" (click)="register()">Not
  Registered?</button>
</form>
```

## login.component.css

```
form{
  display: block;
}

.form-signin{
  width: 100%;
  max-width: 330px;
  padding: 15px;
  margin: auto;
}

.vertically-center{
  margin: 0;
  position: absolute;
  top: 50%;
  -ms-transform: translateY(-50%);
  transform: translateY(-50%);
}

mb-4, .my-4 {
  margin-bottom: 1.5rem!important;
}
```

## Login.component.ts

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

@Component({
  selector: 'app-login',
  templateUrl: './login.component.html',
  styleUrls: ['./login.component.css']
})
export class LoginComponent implements OnInit {
  model: any = {};
  emailPattern = "^[a-z0-9._%+-]+@[a-z0-9.-]+\.[a-z]{2,4}$"
  passwordPattern="^(?=.*[a-z])(?=.*[A-Z])(?=.*\\d)[a-zA-Z\\d\\W]{8,63}$";
  constructor(private route: Router, private authService: AuthService) { }

  ngOnInit(): void {
  }

  login(){
    console.log(this.model);
    this.authService.login(this.model).subscribe((next: any) => {
      console.log(next);
    });
  }
}
```

```

        this.route.navigate(['/welcome']);
    }, error => {
        console.log(error);
        alert('Incorrect Username/Password');
    });
}

register(){
    this.route.navigate(['register']);
}
}

```

## Register Component

### register.component.html

```

<form #loginForm="ngForm" class="form-signin text-center" (submit)="register()">
  <h1 class="h3 mb-3 font-weight-normal">Please Register</h1>
  <label for="inputEmail" class="sr-only">Email address</label>
  <input #Email ngModel type="email" name="email" id="inputEmail"
  [(ngModel)]="model.email" class="form-control" placeholder="Email address"
  [pattern]="emailPattern" required>
  <br>
  <label for="inputPassword" class="sr-only">Password</label>
  <input #Password ngModel type="password" name="password" id="inputPassword"
  [(ngModel)]="model.password" class="form-control" placeholder="Password" required>
  <br><br>
  <button [disabled]="!loginForm.valid" class="btn btn-lg btn-success btn-block"
  type="submit">Register</button>
  <br><button class="btn btn-danger btn-block" (click)="login()">Already
  Registered?</button>
</form>

```

### register.component.css

```

form{
    display: block;
}

.form-signin{
    width: 100%;
    max-width: 330px;
    padding: 15px;
    margin: auto;
}

.vertically-center{
    margin: 0;
}

```

```

    position: absolute;
    top: 50%;
    -ms-transform: translateY(-50%);
    transform: translateY(-50%);
}

mb-4, .my-4 {
    margin-bottom: 1.5rem!important;
}

```

## register.component.ts

```

import { Component, OnInit } from '@angular/core';
import { Router } from '@angular/router';
import { AuthService } from '../auth.service';

@Component({
  selector: 'app-register',
  templateUrl: './register.component.html',
  styleUrls: ['./register.component.css']
})
export class RegisterComponent implements OnInit {

  model: any = {};
  emailPattern = '^[a-z0-9._%+-]+@[a-z0-9.-]+\.[a-z]{2,4}$';
  constructor(private route: Router, private authService: AuthService) { }

  login(){
    this.route.navigate(['login']);
  }

  register(){
    this.authService.register(this.model).subscribe((next: any) => {
      console.log(next);
      this.route.navigate(['/welcome']);
    }, error => {
      console.log(error);
      alert('Could not register. ');
    });
  }

  ngOnInit(): void {
  }
}

```

# Welcome Component

## welcome.component.html

```
<h1 style="text-align: center;">Welcome User!</h1>
```

## POSTMAN Screenshots

### Login testing should give status code 200 OK

A screenshot of the Postman application showing a POST request to `localhost:3000/login`. The request body is a JSON object with `email` and `password` fields. The response status is 200 OK, and the response body is a JSON object containing `_id`, `email`, `password`, and `__v` fields.

POST localhost:3000/login

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "email": "rohanmitta01@gmail.com",
3   "password": "rohan123"
4 }
```

Body Cookies Headers (7) Test Results

Status: 200 OK Time: 18 ms Size: 342 B Save Responses

Pretty Raw Preview Visualize JSON

```
1 {
2   "_id": "5f7e9f3c6d10b1080fb655a7",
3   "email": "rohanmitta01@gmail.com",
4   "password": "rohan123",
5   "__v": 0
6 }
```

### Register Testing should give status code 201 Created

A screenshot of the Postman application showing a POST request to `localhost:3000/register`. The request body is a JSON object with `email` and `password` fields. The response status is 201 Created, and the response body is a JSON object containing `_id`, `email`, `password`, and `__v` fields.

POST localhost:3000/register

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "email": "rohan@gmail.com",
3   "password": "rohan123"
4 }
```

Body Cookies Headers (7) Test Results

Status: 201 Created Time: 17 ms Size: 339 B Save Responses

Pretty Raw Preview Visualize JSON

```
1 {
2   "_id": "5f7ec2e6233d9823e56e1963",
3   "email": "rohan@gmail.com",
4   "password": "rohan123",
5   "__v": 0
6 }
```

## Output Screenshots:

### Initial Database State

_id	email	password	__v
1	rohanmittal01@gmail.com	rohan123	0

### Register Page

Please Register

Email address

Password

Register

Already Registered?

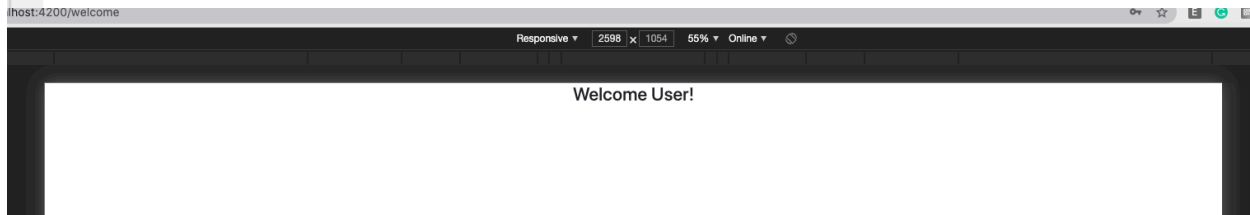
### Database State and Next Page after registering

email: [rohan.mittal2018@vitstudent.ac.in](mailto:rohan.mittal2018@vitstudent.ac.in)

password: rohan123



_id	email	password	__v
1	ObjectId(...) rohanmittal01@gmail.com	rohan123	0
2	ObjectId(...) rohan.mittal2018@vitstudent.ac.in	rohan123	0



## Login Page

Please sign in



Sign in

Not Registered?

## After Logging In:

