Student portal web application

LAb-1

| comp-sci490-002: web/mobile programming| 10-09-2019

Team 11

Manohar class id:07

Pavan kumar chongala class id: 09

Sai varun class id:13

# Introduction:

The student web portal is collection of links to different web pages. It connects many web pages of the activities that student can perform. In this web application students will register first and then student will access different services that are provided in the web application. Students can register the courses and also can perform update and delete function. In student web portal students can also check the calendar and also buy books from the bookstore. Even user can check three of his friends profile.

## Objectives:

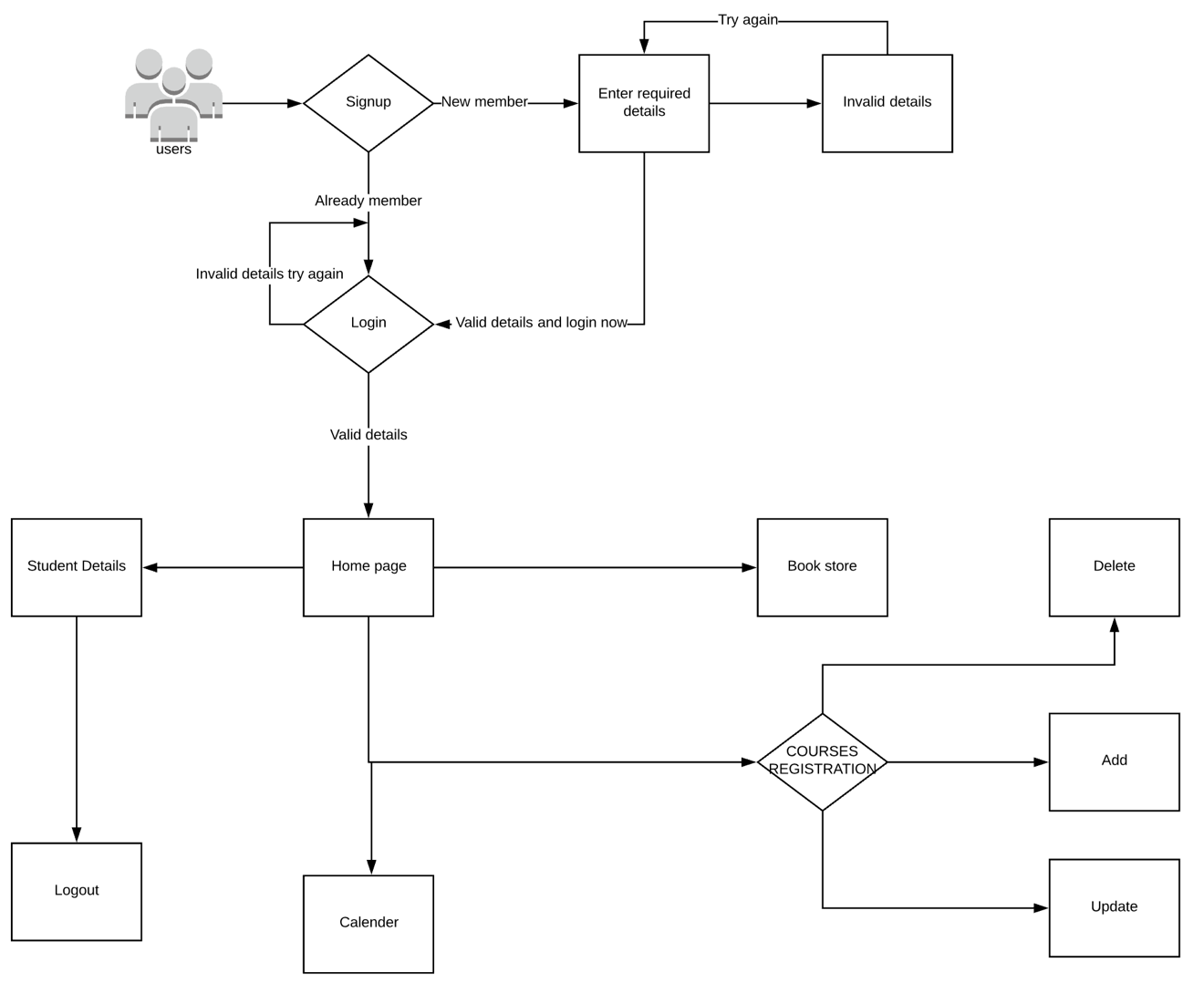
Our main objective is to provide user friendly web portal. The student portal application where student can able to register and login from any place and also can easily access from any screen resolution. We made this web portal mainly as single page application using MEAN stack and Angular CLI.

# Approach:

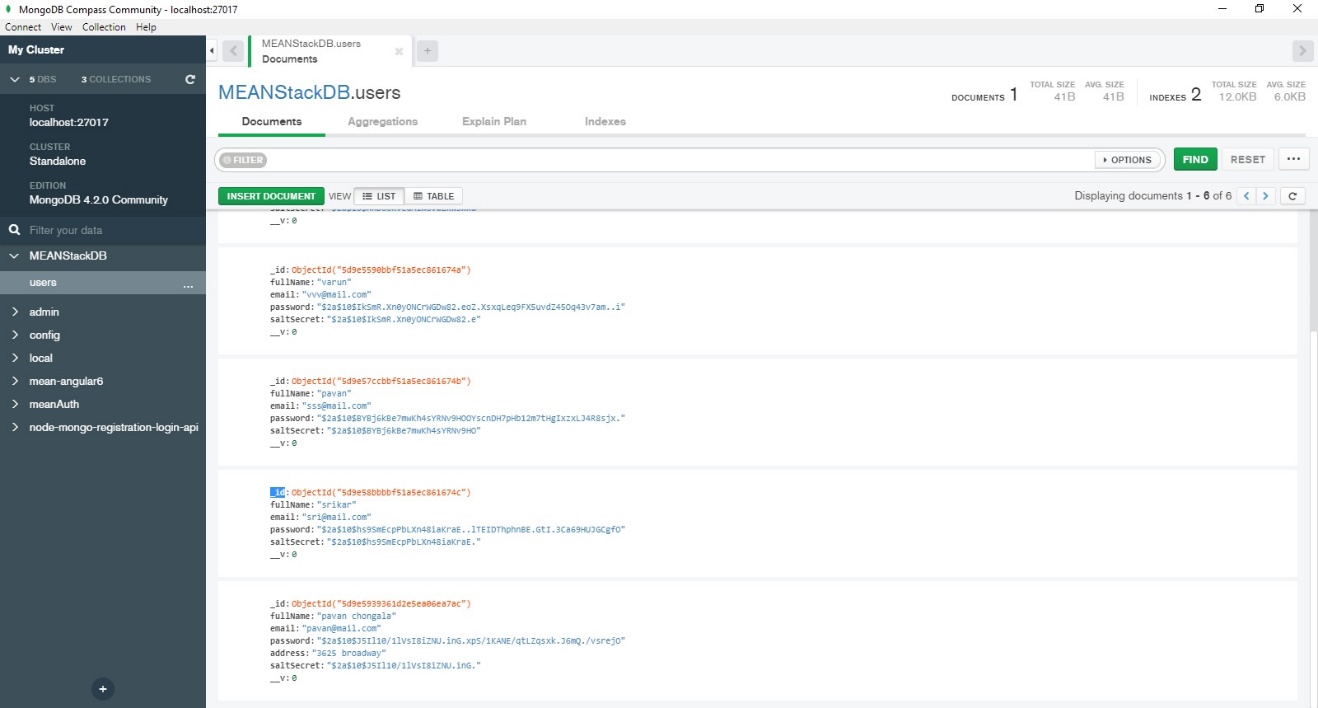
We were inspired from the UMKC pathway for the services to provide in our online student portal.

We spent about 15days to make this portal after learning the MEAN and angular cli because we dont have any prior knowledge about the MEAN stack.

# Workflow:

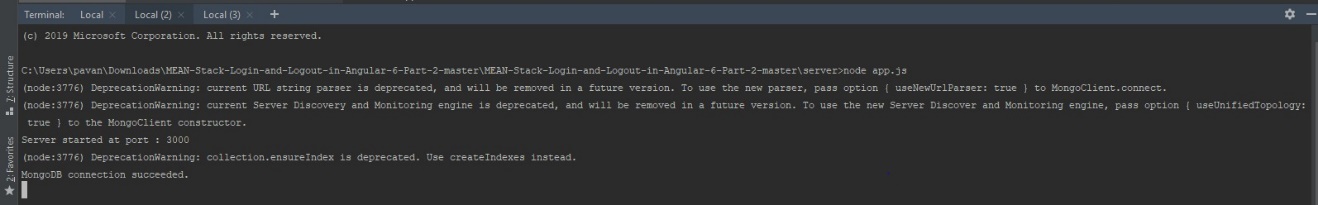


## Database and connection establish:



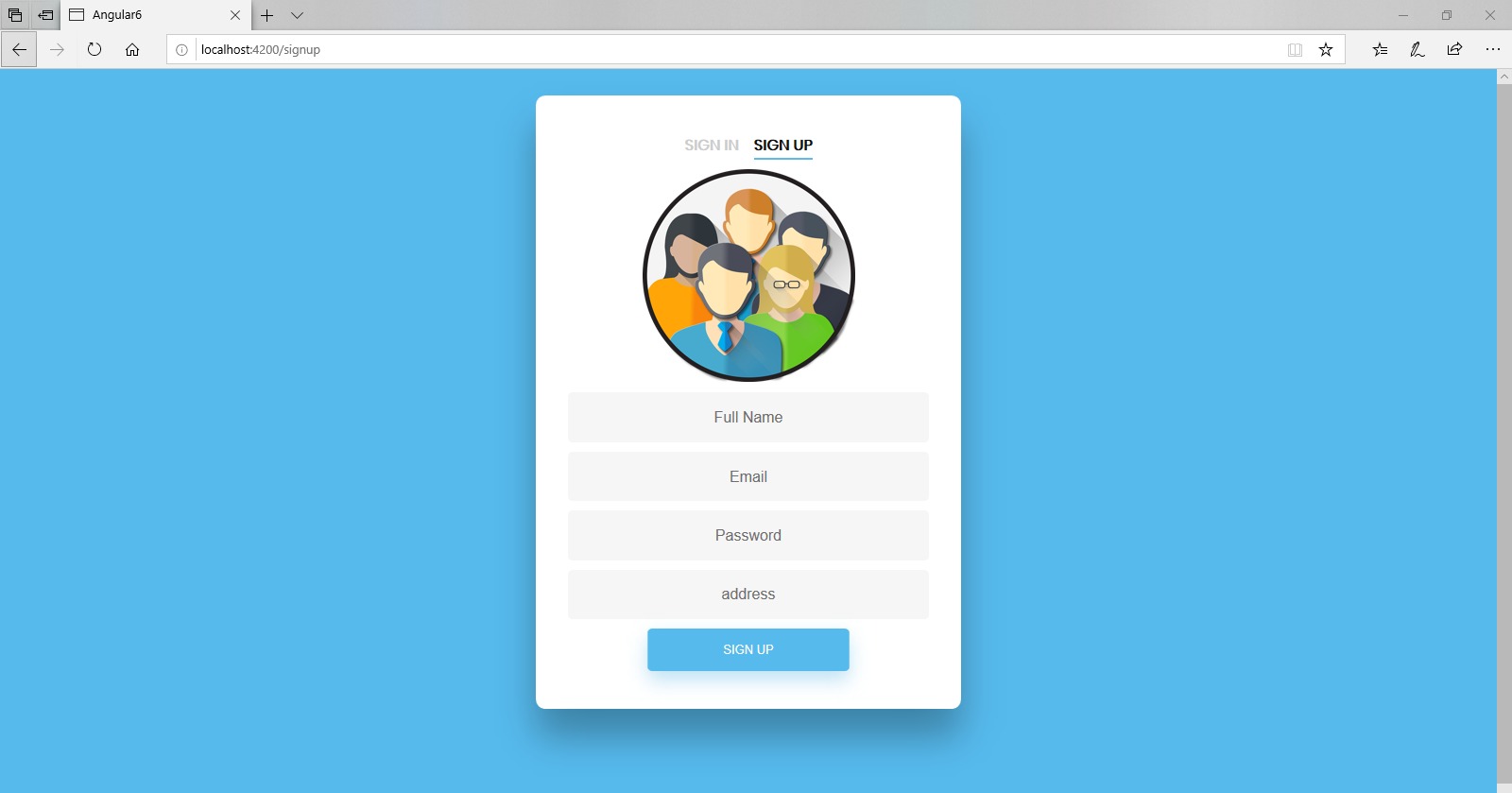
For this student web portal we are using MONGODB as database to store the User registration and login details and also for the update, delete and edit function of Courses.

MONGODB is document type database which does not requires any relational database and also it creates its own unique id. This database is also called NOSQL database.



After installing the MONGODB we need to provide some commands to establish the connection and to start the MONGO server. To establish the project connection with database we are using the NODE.JS for the validation.

## Signup:



Our student portal first page is sign up page were we will need details of the user like

Full name, Email, Password, Address.

Here database manager also cant see the actual password of the users because we are encrypting the password using bcryptjs. Password will be merged with the random string before encryption because it is more secured then encryption.

**Signup html code:**

<div class="wrapper">

<div id="formContent">

<h2 class="underlineHover" routerLink="/login" routerLinkActive="active"> Sign In </h2>

<h2 class="underlineHover" routerLink="/signup" routerLinkActive="active">Sign Up </h2>

<router-outlet></router-outlet>

</div>

</div>

**Sign up ts:**

mport { Component, OnInit } from '@angular/core';

@Component({

selector: 'app-user',

templateUrl: './user.component.html',

styleUrls: ['./user.component.css']

})

export class UserComponent implements OnInit {

constructor() { }

ngOnInit() {

}

}

**Password Encryption:**

// check env.

var env = process.env.NODE\_ENV || 'development';

// fetch env. config

var config = require('./config.json');

var envConfig = config[env];

// add env. config values to process.env

Object.keys(envConfig).forEach(key => process.env[key] = envConfig[key]);

**User controller:**

This is used to save the new User details into MONGODB using moongose

const mongoose = require('mongoose');

const User = mongoose.model('User');

module.exports.register = (req, res, next) => {

    var user = new User();

    user.fullName = req.body.fullName;

    user.email = req.body.email;

    user.password = req.body.password;

    user.save((err, doc) => {

        if (!err)

            res.send(doc);

        else {

            //error handling

        }

    });

}

**Route file:**

This is to create the route for the app.js

require('./config/config');

require('./models/db');

const express = require('express');

const bodyParser = require('body-parser');

const cors = require('cors');

var app = express();

// middleware

app.use(bodyParser.json());

app.use(cors());

// start server

app.listen(process.env.PORT, () => console.log(`Server started at port : ${process.env.PORT}`)

**Implementing route:**

const express = require('express');

const router = express.Router();

const ctrlUser = require('../controllers/user.controller');

router.post('/register', ctrlUser.register);

module.exports = router;

**Error handling:**

To the unexpected errors we are using the error handles.

...

// error handler

app.use((err, req, res, next) => {

    if (err.name === 'ValidationError') {

        var valErrors = [];

        Object.keys(err.errors).forEach(key => valErrors.push(err.errors[key].message));

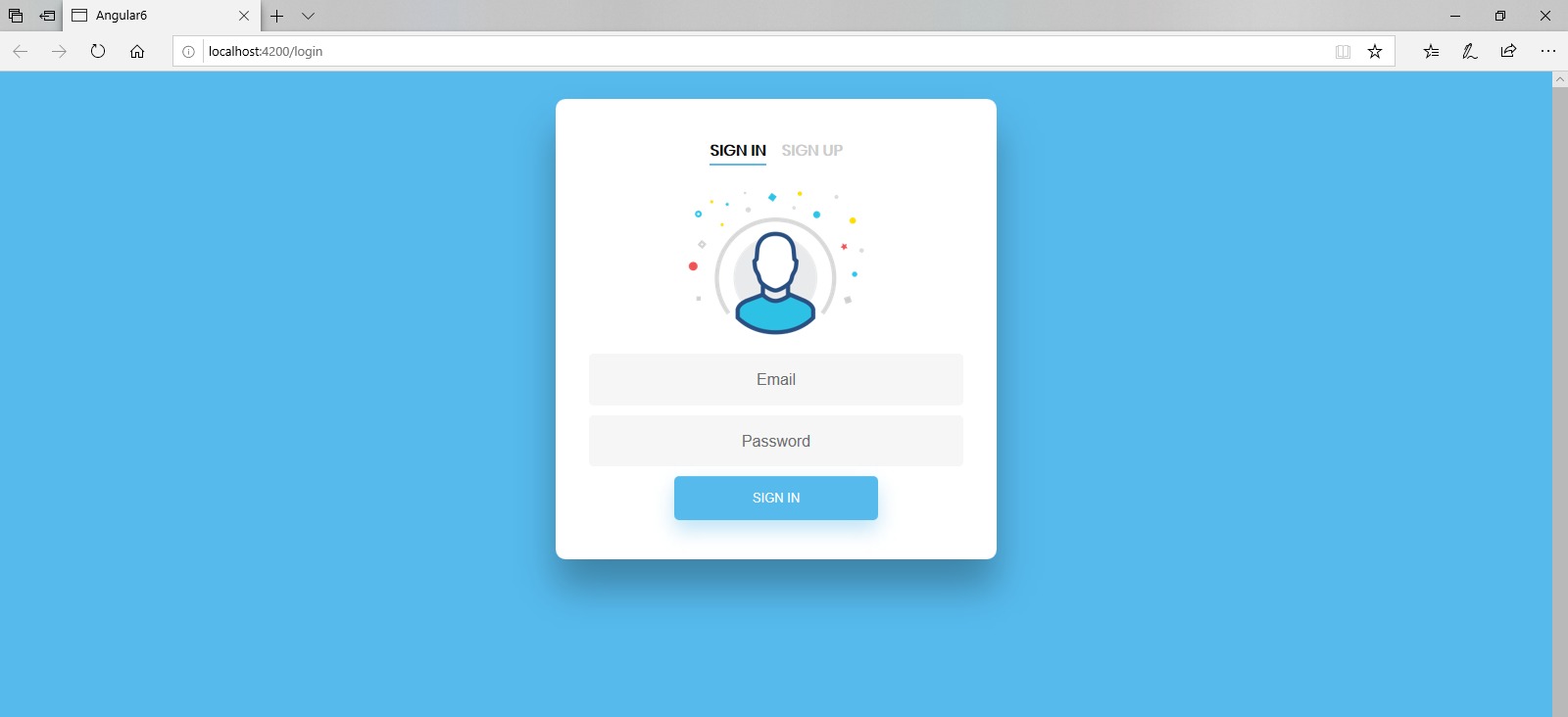
        res.status(422).send(valErrors)

    }

});

...

## Sign in page:

****

**HTML**

<div>

<img src="assets/img/single\_user.png" id="icon" alt="User Icon" />

</div>

<form #signInForm="ngForm" (ngSubmit)="signInForm.valid && onSubmit(signInForm)">

<input type="text" name="email" #email="ngModel" [(ngModel)]="model.email" placeholder="Email" [pattern]="emailRegex" required

[ngClass]="{'invalid-textbox' :signInForm.submitted && !email.valid }">

<div \*ngIf="signInForm.submitted && email.errors?.pattern">

<label class="validation-message">Invalid email address.</label>

</div>

<input type="password" name="password" #password="ngModel" [(ngModel)]="model.password" placeholder="Password" required minlength="4"

[ngClass]="{'invalid-textbox' :signInForm.submitted && !password.valid }">

<div \*ngIf="signInForm.submitted && password.errors?.minlength">

<label class="validation-message">Minimum 4 characters.</label>

</div>

<input type="submit" value="Sign In">

</form>

<!-- Error message -->

<div class="alert" \*ngIf="serverErrorMessages">

{{serverErrorMessages}}

</div>

**Component.ts**

import { UserService } from '../../shared/user.service';

@Component({

selector: 'app-sign-in',

templateUrl: './sign-in.component.html',

styleUrls: ['./sign-in.component.css']

})

export class SignInComponent implements OnInit {

constructor(private userService: UserService, private router: Router) { }

model = {

email : '',

password: ''

};

emailRegex = /^(([^<>()\[\]\\.,;:\s@"]+(\.[^<>()\[\]\\.,;:\s@"]+)\*)|(".+"))@((\[[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\])|(([a-zA-Z\-0-9]+\.)+[a-zA-Z]{2,}))$/;

serverErrorMessages: string;

ngOnInit() {

if (this.userService.isLoggedIn()) {

this.router.navigateByUrl('/userprofile');

}

}

onSubmit(form: NgForm) {

this.userService.login(form.value).subscribe(

res => {

this.userService.setToken(res['token']);

this.router.navigateByUrl('/userprofile');

},

err => {

this.serverErrorMessages = err.error.message;

}

);

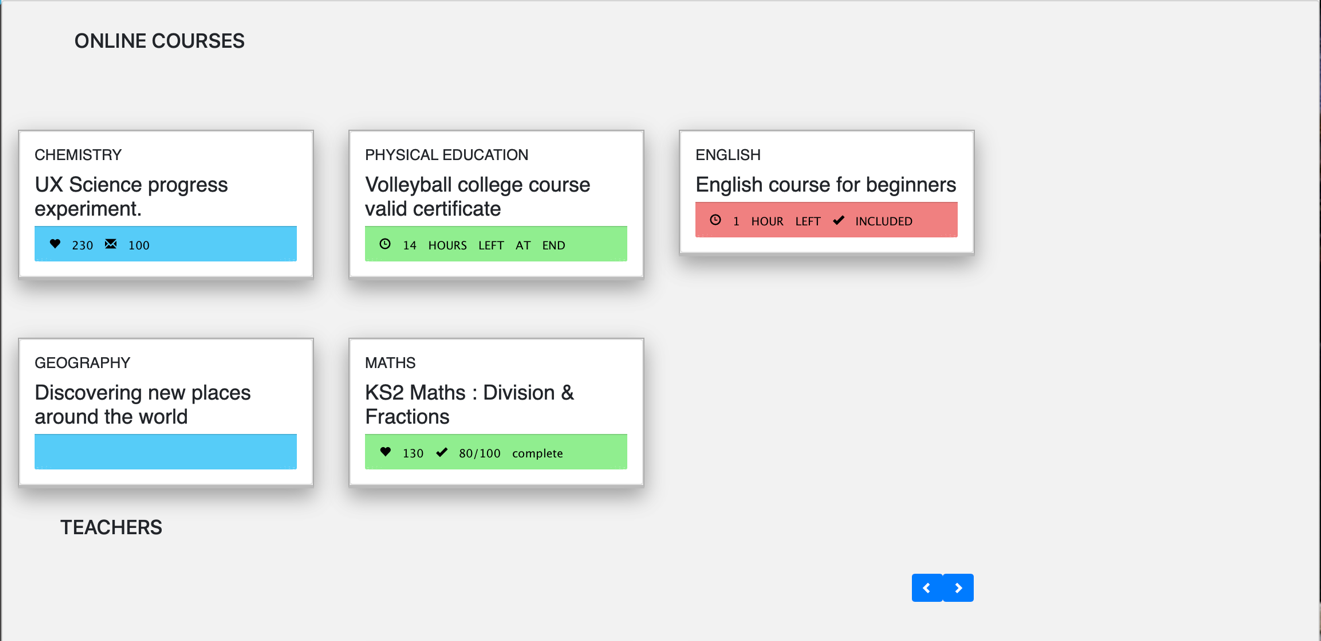
}

}

## Homepage:

### Main content:

In this it will display the courses and timings. We used bootstrap for styling and Glyphicon for the widget icons and Angular for client side verification.



HTML

<div class="card" style="padding-top: 0.99%; background: #f2f2f2" >

<div class="card-body">

<div class="row-12">

<div class="col-3" style="padding-left: 4.5%;">

<h3 class="card-title">ONLINE COURSES</h3>

</div >

<div class="list-item" style="max-height: 50%">

<div class="row">

<div class="col-3" style="padding-top: 0.85%;text-align: end; width: 0%">

<li \*ngFor="let item of coursesDetails" class="cards-list-item">

<div class="card h-100 w-100 text-{{item.color}} bg-{{item.background}}" style="padding-left: 0%;padding-right: 0%">

<div class="card-body" style="padding-left: 5%;padding-right: 5%">

<h5 class="card-title text-left">{{item.header}}</h5>

<h3 class="card-text text-left">{{item.content}}</h3>

<div class="card-footer main\_footer1 text-left">

<span class="glyphicon glyphicon-{{item.likes\_id}}" style="padding-right: 5%;color: black"> {{item.likes}} </span> <span class="glyphicon glyphicon-{{item.comments\_id}}" style="padding-right: 5%;color: black"> {{item.comments}} </span>

</div>

</div>

</div>

</li>

</div>

<div class="col-3" style="padding-top: 0.85%;text-align: end; width: 0%">

<li \*ngFor="let item of subjectCourses" class="cards-list-item">

<div class="card h-100 w-100 text-{{item.color}} bg-{{item.background}}" style="padding-left: 0%;padding-right: 0%">

<div class="card-body" style="padding-left: 5%;padding-right: 5%">

<h5 class="card-title text-left">{{item.header}}</h5>

<h3 class="card-text text-left">{{item.content}}</h3>

<div class="card-footer main\_footer2 text-left">

<span class="glyphicon glyphicon-{{item.likes\_id}}" style="padding-right: 5%;color: black"> {{item.likes}} </span> <span class="glyphicon glyphicon-{{item.comments\_id}}" style="padding-right: 5%;color: black"> {{item.comments}} </span>

</div>

</div>

</div>

</li>

</div>

<div class="col-3" style="padding-top: 0.85%;text-align: end; width: 0%">

<li \*ngFor="let item of courses" class="cards-list-item">

<div class="card h-100 w-100 text-{{item.color}} bg-{{item.background}}" style="padding-left: 0%;padding-right: 0%">

<div class="card-body" style="padding-left: 5%;padding-right: 5%">

<h5 class="card-title text-left">{{item.header}}</h5>

<h3 class="card-text text-left">{{item.content}}</h3>

<div class="card-footer main\_footer3 text-left">

<span class="glyphicon glyphicon-{{item.likes\_id}}" style="padding-right: 5%;color: black"> {{item.likes}} </span> <span class="glyphicon glyphicon-{{item.comments\_id}}" style="padding-right: 5%;color: black"> {{item.comments}} </span>

</div>

</div>

</div>

</li>

</div>

</div>

<div class="row">

<div class="col-3" style="padding-left: 4.5%; padding-top: 2%">

<h3 class="card-title">TEACHERS</h3>

</div >

<div class="col-9 text-right" style="padding-left: 4.5%; padding-top: 2%">

<button type="button" class="btn btn-primary"><span class="glyphicon glyphicon-chevron-left"></span></button><button type="button" class="btn btn-primary"><span class="glyphicon glyphicon-chevron-right"></span></button>

</div>

</div>

<div class="row-3">

<div class="col col-md-3" style="padding-bottom: 2.4%">

<li \*ngFor="let item of teachers" class="cards-list-item">

<div class="card h-100 w-100 text-{{item.color}} bg-{{item.background}}" style="padding-left: 0%;padding-right: 0%">

<div class="card-body" style="padding-left: 5%;padding-right: 5%">

<div class="row">

<div class="col-2">

<img src="../../../../assets/{{item.image}}" class="rounded-circle" width="20" height="20">

</div>

<div class="col-10 text-left">

<h3 class="card-title text-left">{{item.name}}</h3>

<h5 class="card-text text-left">{{item.profession}}</h5>

</div>

</div>

</div>

</div>

</li>

</div>

<div class="col col-md-3">

<li \*ngFor="let item of chemistryTeacher" class="cards-list-item">

<div class="card h-100 w-100 text-{{item.color}} bg-{{item.background}}" style="padding-left: 0%;padding-right: 0%">

<div class="card-body" style="padding-left: 5%;padding-right: 5%">

<div class="row">

<div class="col-2">

<img src="../../../../assets/{{item.image}}" class="rounded-circle" width="20" height="20">

</div>

<div class="col-10 text-left">

<h3 class="card-title text-left">{{item.name}}</h3>

<h5 class="card-text text-left">{{item.profession}}</h5>

</div>

</div>

</div>

</div>

</li>

</div>

<div class="col col-md-3">

<li \*ngFor="let item of mathmatics" class="cards-list-item">

<div class="card h-100 w-100 text-{{item.color}} bg-{{item.background}}" style="padding-left: 0%;padding-right: 0%">

<div class="card-body" style="padding-left: 5%;padding-right: 5%">

<div class="row">

<div class="col-2">

<img src="../../../../assets/{{item.image}}" class="rounded-circle" width="20" height="20">

</div>

<div class="col-10 text-left">

<h3 class="card-title text-left">{{item.name}}</h3>

<h5 class="card-text text-left">{{item.profession}}</h5>

</div>

</div>

</div>

</div>

</li>

</div>

</div>

</div>

</div>

Component.ts:

import { Component, OnInit } from '@angular/core';

@Component({

selector: 'app-content',

templateUrl: './content.component.html',

styleUrls: ['./content.component.scss']

})

export class ContentComponent implements OnInit {

coursesDetails = [

{

background: "white",

color:"black",

header: "CHEMISTRY",

content: "UX Science progress experiment.",

likes\_id: "heart",

likes: 230,

comments\_id: "envelope",

comments: 100,

footer\_background: "pink",

footer\_color:"white"

},

{

background: "white",

color:"black",

header: "GEOGRAPHY",

content: "Discovering new places around the world",

add : "add to your courses",

footer\_background: "rgb(86, 204, 247);",

footer\_color:"white"

}

]

subjectCourses = [

{

background: "white",

color:"black",

header: "PHYSICAL EDUCATION",

content: "Volleyball college course valid certificate",

likes\_id: "time",

comments: "14 HOURS LEFT AT END",

footer\_background: "green",

footer\_color:"white"

},

{

background: "white",

color:"black",

header: "MATHS",

content: "KS2 Maths : Division & Fractions",

likes\_id: "heart",

likes: 130,

comments\_id: "ok",

comments: "80/100 complete",

footer\_background: "violet",

footer\_color:"white"

}

]

courses = [

{

background: "white",

color:"black",

header: "ENGLISH",

content: "English course for beginners",

likes\_id: "time",

likes: "1 HOUR LEFT",

comments\_id: "ok",

comments: "INCLUDED",

footer\_background: "red",

footer\_color:"white"

}

]

teachers = [

{

image:"Jonathan Smithers.png",

name: "Jonathan Smithers",

profession: "English teacher"

}

]

chemistryTeacher= [

{

image:"sree.jpg",

name: "James Howell",

profession: "Chemistry"

}

]

mathmatics=[

{

image:"Jane Austen.png",

name: "Jane Austen",

profession: "Mathematics"

}

]

constructor() { }

ngOnInit() {

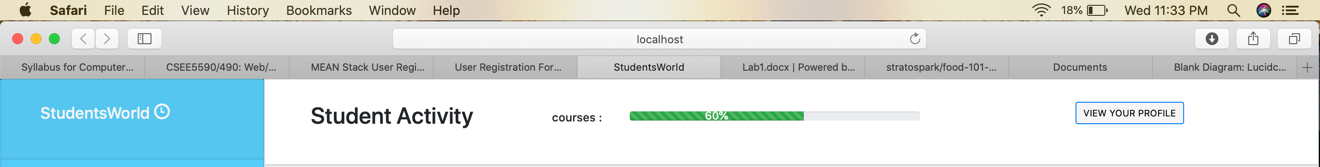
}

}

### Header:

We used bootstrap for styling and here we used Javascript to update the courses percentage based on the user courses registered

Onselect button for view profile to see the user profile and also can update the profile. For which we are extracting the data from MongoDB.



Html:

<div class="card" style="padding-top: 0.99%; background: white" >

<div class="card-body">

<div class="row">

<div class="col-3" style="padding-left: 4.5%;">

<h1 class="card-title">Student Activity</h1>

</div>

<div class="col-1" style="padding-top: 0.85%;text-align: end">

<h5 class="card-title">courses : </h5>

</div>

<div class="col-5" style="padding-top: 0.9%; max-width: 30%;">

<div class="progress">

<div class="progress-bar bg-success progress-bar-striped progress-bar-animated" style="width:60%">60%</div>

</div><br>

</div>

<div class="col-3 text-right">

<a class="btn btn-primary" (click)="onSelect()" style="background: whitesmoke;color: black;">VIEW YOUR PROFILE</a>

</div>

</div>

</div>

</div>

Component.ts:

import { Component, OnInit } from '@angular/core';

import {UserInfoComponent} from '../../user-info/user-info.component'

import { Router } from '@angular/router';

@Component({

selector: 'app-header',

templateUrl: './header.component.html',

styleUrls: ['./header.component.scss']

})

export class HeaderComponent implements OnInit {

constructor(private router: Router) { }

ngOnInit() {

}

onSelect= function (){

this.router.navigate(['/userInfo']);

}

}

SIDE BAR:

Friends:

<div class="mainstudents">

<span class="sidebar-menu-text">{{friendsHeading.name}}</span>

</div>

<li \*ngFor="let item of friendsList" class="sidebar-list-item">

<a class="sidebar-menu-link"[routerLink]="[item.route]" routerLinkActive="active">

<div class="row">

<div class="col-1">

<img src="../../../../assets/{{item.id}}" class="rounded-circle" width="20" height="20">

</div>

<div class="col-7">

<span class="sidebar-menu-text" style="color: white; font-size: 80%">{{item.name}}</span>

</div>

<div class="col-2">

<span class="glyphicon glyphicon-{{item.message\_icon\_id}}" style="padding-top: 5%;color: white; position: absolute;"></span>

</div>

<div class="col-1">

<span class="sidebar-menu-text" style="color: white; position: absolute;">{{item.value}}</span>

</div>

</div>

</a>

</li>

<div class="mainstudents">

<button type="button" class="btn btn-primary" style="margin-left: 30%; background: rgb(60, 60, 61); border-block-color: rgb(60, 60, 61);"> <span class="glyphicon glyphicon-plus-sign"></span></button>

</div>

Component.ts:

import { Component, OnInit } from '@angular/core';

@Component({

selector: 'app-friends',

templateUrl: './friends.component.html',

styleUrls: ['./friends.component.scss']

})

export class FriendsComponent implements OnInit {

friendsList = [

{

id: "manohar.jpg",

name: "Manohar Bollampally",

message\_icon\_id : "comment",

value: 3

},

{

id: "harsha.jpg",

name: "S.l.a.s.h",

},

{

id: "ranga.jpg",

name: "Sree",

}

]

friendsHeading = {

id: "Friends",

name: "FRIENDS",

}

constructor() { }

ngOnInit() {

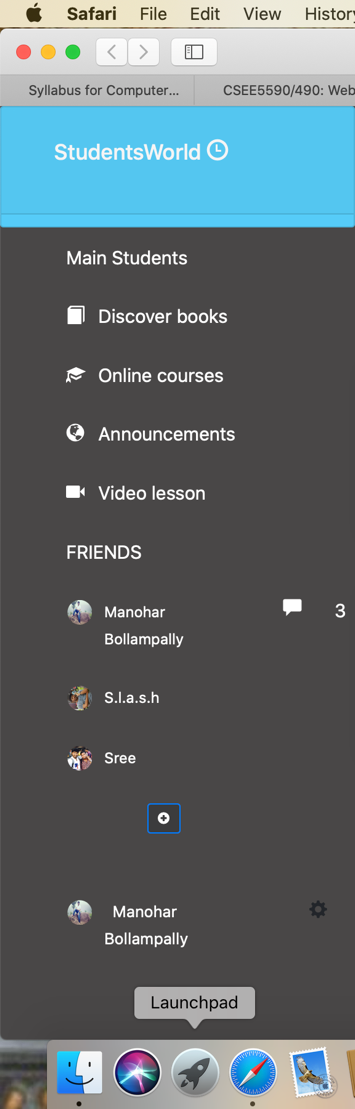
}

}

### Side Bar Main student:

User can go to UMKCbookstore by clicking the dicover books.

And Friends list to see friends profile.



**HTML:**

<div class="student">

<div class="mainstudents">

<span class="sidebar-menu-text">{{sideHeading.name}}</span>

</div>

<ul class="list-group referencematerial">

<li \*ngFor="let item of iconInfo" class="sidebar-list-item">

<a class="sidebar-menu-link"[routerLink]="[item.route]" routerLinkActive="active">

<span class="glyphicon glyphicon-{{item.id}}" style="padding-right: 5%;color: white"></span> <span class="sidebar-menu-text" style="color: white">{{item.name}}</span>

</a>

</li>

<app-friends></app-friends>

<div class="footer">

<div class="row">

<div class="col-1">

<img src="../../../../assets/{{userData.id}}" class="rounded-circle" width="20" height="20">

</div>

<div class="col-8">

<a class="name" (click)="onSelect()" style="font-size: 85% ;color: white;padding-left: 5%;padding-right: 10%;">{{userData.name}}</a>

</div>

<div class="col-2">

<span class="glyphicon glyphicon-{{userData.settingId}}"></span>

</div>

</div>

</div>

</ul>

</div>

Component.ts:

import { Component, OnInit } from '@angular/core';

import {UserInfoComponent} from '../../user-info/user-info.component'

import { Router } from '@angular/router';

@Component({

selector: 'app-main-students',

templateUrl: './main-students.component.html',

styleUrls: ['./main-students.component.scss']

})

export class MainStudentsComponent implements OnInit {

sideHeading = {

id: "MainStudent",

name: "Main Students",

}

iconInfo = [

{

id: "book",

name: "Discover books",

},

{

id: "education",

route: "MainContentComponent",

name: "Online courses",

},

{

id: "globe",

name: "Announcements",

},

{

id: "facetime-video",

name: "Video lesson",

}

]

userData = {

id: "manohar.jpg",

name: "Manohar Bollampally",

settingId: "cog"

}

constructor(private router: Router) { }

ngOnInit() {

}

onSelect= function (){

this.router.navigate(['/userInfo']);

}

}

### User info:

HTML:

<div class="container-fuild" style="padding: 1rem;">

<div class="contentHeader\_\_headline">

<h1>User Information</h1>

</div>

<div class ="userinfo" \*ngIf="userDetails">

<p \*ngIf="userDetails.email">User Email : <i class ="text-muted">{{userDetails.email}}</i></p>

<p \*ngIf="userDetails.name">User name : <i class ="text-muted">{{userDetails.name}}</i></p>

<p \*ngIf="userDetails.password">User password : <i class ="text-muted">{{userDetails.password}}</i></p>

</div>

</div>

Component.ts:

import { Component, OnInit } from '@angular/core';

@Component({

selector: 'app-user-info',

templateUrl: './user-info.component.html',

styleUrls: ['./user-info.component.scss']

})

export class UserInfoComponent implements OnInit {

userDetails = {

name : "manohar bollampally",

email : "manoharbollampally@gmail.com",

password : "mounika"

}

constructor() { }

ngOnInit() {

}

# Parameters:

MEAN stack-

MongoDB for database,

Express JS for client-side verification,

Angular JS client-side verification,

Node JS Backend verification.

JavaScript Client-side verification,

Styling:

CSS

Bootstrap

HTML page

NPM for node js installation

Packages used:

"bcryptjs": "^2.4.3",

"body-parser": "^1.19.0",

"cors": "^2.8.5",

"express": "^4.17.1",

"jsonwebtoken": "^8.5.1",

"lodash": "^4.17.15",

"mongoose": "^5.7.4",

"pass": "^0.2.0",

"passport": "^0.4.0",

"passport-local": "^1.0.0"

# Conclusion:

This portal is used for the educational institutions and this portal is secured because the Password of the user is encrypted even Database admin can’t get the login credentials.

We can further enhance the security and we can increase the services with the chat applications, announcements, Marks, Group discussions etc…