Online Collaboration System

**SOURCE CODE**

1. **Frontend -**
   1. **LOGIN PAGE –**
2. **login.component.html**

<div class="container top-con">

<div class="row justify-content-center">

<div class="col-md-5">

<mat-card class="login-card"> <mat-card-content class="card-body text-center">

<img src="assets/lock.png" class="img-fluid" style="height: 10rem"> <form [formGroup]="loginform" (submit)="formSubmit(loginform.value)" class="mt-5">

<mat-form-field class="col-md">

<mat-label>Username</mat-label>

<input matInput placeholder="Mr. Anderson"

formControlName="username">

<fa-icon class="float" [icon] ="userIcon" matSuffix></fa-icon> <mat-error \*ngIf="getControls().username.errors">Username

required</mat-error>

</mat-form-field>

<mat-form-field class="col-md mt-3">

<mat-label>Password</mat-label> <input matInput placeholder="\*\*\*\*\*\*\*\*\*\*" formControlName="password" type="password">

<fa-icon class="float" [icon] ="lock" matSuffix></fa-icon> <mat-error \*ngIf="getControls().password.errors">Password

required</mat-error>

</mat-form-field>

<button class="btn-block mt-5" mat-raised-button color="primary">Login</button>

</form>

</mat-card-content>

<mat-card-actions>

<hr>

<div class="row" style="padding-left: 8px;padding-right: 8px;"> <div class="col-md">

<p [routerLink]="['/resetpw']" routerLinkActive="router-link-active"

class="text- muted">Forgot password?</p>

Online Collaboration System

</div>

<div class="col-md">

<p class="col" [routerLink]="['/register']" routerLinkActive="router-link-active" class="text-muted float-right">Not registered yet?</p>

</div>

</div>

</mat-card-actions>

</mat-card>

</div>

</div>

</div>

1. **login.component.ts**

import { Component, OnInit } from '@angular/core'; import { FormBuilder, Validators } from '@angular/forms'; import { Users } from '../users';

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

import {faUser, faLock} from '@fortawesome/free-solid-svg-icons'; import Swal from 'sweetalert2';

@Component({ selector: 'app-login',

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

styleUrls: ['./login.component.css'] })

export class LoginComponent implements OnInit { userIcon = faUser;

lock = faLock; loginform; logged\_user; user: string;

constructor(private formbuilder: FormBuilder, private authservice: AuthService, private router: Router ) { }

ngOnInit() { document.body.classList.add('bg-lg') this.initForm();

}

initForm() {

this.loginform = this.formbuilder.group({ username : ['', Validators.required],

Online Collaboration System

password : ['', Validators.required]

} );

}

formSubmit(formdata) {

if ( this.loginform.invalid ) {

return;

}

this.authservice.getUser(formdata.username).subscribe(data => {

const user: Users = data;

if (user) {

if (user.password === formdata.password ) { sessionStorage.setItem( "user" , JSON.stringify(user)); this.authservice.setLogin();

if (user['admin']) {

sessionStorage.setItem( "admin" , JSON.stringify(user)); this.router.navigate(['/admindash'])

} else { this.router.navigate(['/userdash'])

}

}else{

console.log('wrong password!!')

Swal.fire({

icon : 'error',

title : 'Error!',

text : 'Username or password is invalid'

})

}

}else{

console.log('user not found!!')

Swal.fire({

icon : 'error',

title : 'Error!',

text : 'Username or password is invalid'

})

} });

}

getControls(){

return this.loginform.controls;

}

ngOnDestroy(){

Online Collaboration System

document.body.classList.remove('bg-lg');

}}

1. **REGISTER PAGE**

**a. register.component.html**

<div class="container top-con">

<p class="display-4 text-center">New Registration</p> <mat-card>

<mat-card-content >

<div class="row">

<div class="col-md-6">

<img style="width: 500px; height: 430px;" src="./assets/image 6.jpg" > </div>

<div class="col-md-6">

<mat-horizontal-stepper [linear]="true">

<mat-step [stepControl]="registerForm">

<!-- Step 1 -->

<ng-template>Fill your details</ng-template>

<form [formGroup]="registerForm">

<mat-form-field appearance="outline" style="font-size: small;"

class="col-md-6">

<mat-label>First Name</mat-label> <input matInput placeholder="Mr. Anderson"

formControlName="first">

<mat-error \*ngIf="returnControls().first.errors">First name is Required</mat-error>

</mat-form-field>

<mat-form-field appearance="outline" style="font-size: small;"

class="col-md-6">

<mat-label>Last Name</mat-label>

<input matInput placeholder="Kennedy"

formControlName="last">

<mat-error \*ngIf="returnControls().last.errors">Last Name is Required</mat-error>

</mat-form-field>

<mat-form-field appearance="outline" style="font-size: small;"

class="col-md">

<mat-label>Username</mat-label>

Online Collaboration System

<input matInput placeholder="Mr. Anderson"

formControlName="username">

<mat-error \*ngIf="returnControls().username.errors">Username is Required</mat-error>

</mat-form-field>

<mat-form-field appearance="outline" style="font-size: small;"

class="col-md-6">

<mat-label>Password</mat-label> <input matInput placeholder="Mr. Anderson"

formControlName="password" [type]="hide ? 'password' : 'text'"> <button mat-icon-button matSuffix (click)="hide = !hide"

[attr.aria-label]="'Hide password'" [attr.aria-pressed]="hide"

type="button">

<mat-icon>{{hide ? 'visibility\_off' : 'visibility'}}</mat-icon> </button>

<mat-error \*ngIf="returnControls().password.errors">Password is Required</mat-error>

</mat-form-field>

<mat-form-field appearance="outline" style="font-size: small;"

class="col-md-6">

<mat-label>Confirm</mat-label> <input matInput placeholder="Mr. Anderson" formControlName="confirm" [type]="hide ? 'password' : 'text'">

<button mat-icon-button matSuffix (click)="hide = !hide" [attr.aria-label]="'Hide password'" [attr.aria-pressed]="hide" type="button">

<mat-icon>{{hide ? 'visibility\_off' : 'visibility'}}</mat-icon> </button>

<mat-error \*ngIf="returnControls().confirm.errors">Passwords must match</mat-error>

</mat-form-field>

<mat-form-field appearance="outline" style="font-size: small;"

class="col-md">

<mat-label>Email</mat-label>

<input matInput placeholder="Mr. Anderson"

formControlName="email">

<mat-error \*ngIf="returnControls().email.errors">Email is Required</mat-error>

</mat-form-field>

<div class="col-md-10">

Online Collaboration System

<button mat-raised-button color="warn" class="ml-auto mt-3" matStepperNext>Next</button>

</div>

</form>

</mat-step>

<mat-step>

<!-- Step 2 -->

<div class="row sec-row">

<div class="col-md-6">

<div file="Click or drag file" >

<input class="myinput" type="file" (change) = "onfileUpload($event)">

</div>

<button color="accent" mat-raised-button class="btn-block" (click)="userSubmit()">Sign Up</button>

</div>

<div class="col-md-6">

<img [src]="imgURL" \*ngIf="imgURL" class="img-fluid" style="width: 100%;">

</div>

</div>

</mat-step>

</mat-horizontal-stepper>

</div>

</div>

</mat-card-content>

</mat-card>

<p class="text-center h5 mt-3"><a [routerLink]="['/login']"

routerLinkActive="router-link-active" >

Login Instead

</a></p>

</div>

**b. register.component.ts –**

import { Component, OnInit } from '@angular/core'; import { FormBuilder, Validators } from '@angular/forms'; import { UserService } from '../user.service'; import Swal from 'sweetalert2';

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

@Component({

Online Collaboration System

selector: 'app-register',

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

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

})

export class RegisterComponent implements OnInit { registerForm;

selectedFile;

avatarName;

message;

imgURL;

hide = true;

constructor(private formBuilder: FormBuilder, private userservice: UserService, private

router: Router) {

}

ngOnInit() {

this.initForm();

}

initForm() {

this.registerForm = this.formBuilder.group({

first : ['', Validators.required],

last : ['', Validators.required],

username : ['', Validators.required],

email : ['', [Validators.required, Validators.email]],

password : ['', Validators.required],

confirm : ['', Validators.required],

admin : false,

created : new Date()

});

}

getControl(name) {

return this.registerForm.controls[name];

}

userSubmit() {

let formdata = this.registerForm.value;

if(!this.registerForm.valid){

alert('invalid form');

return;

}

this.userservice.getUserByUsername(formdata.username).subscribe(data => { if(!data){

Online Collaboration System

formdata.avatar = this.avatarName; this.userservice.addUser(formdata).subscribe(data => {

console.log(data['message']);

Swal.fire({

icon : 'success',

title : 'Success',

text : 'You have registered successfully, Now login to continue' })

.then(() => {

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

})

});

}else{

Swal.fire({

icon : 'error',

title : 'Error Occured',

text : 'Username already exists!'

})}

})}

onfileUpload(event){

let formdata = new FormData();

this.selectedFile = event.target.files[0];

this.avatarName = this.selectedFile.name;

this.preview(event.target.files);

formdata.append('avatar', this. selectedFile, this.selectedFile.name); this.userservice.uploadImage(formdata).subscribe(response => {

console.log(response.message);

console.log(this.avatarName);

})}

preview(files) {

if (files.length === 0)

return;

var mimeType = files[0].type;

if (mimeType.match(/image\/\*/) == null) { this.message = "Only images are supported."; return;

}

var reader = new FileReader();

reader.readAsDataURL(files[0]);

reader.onload = (\_event) => {

Online Collaboration System

this.imgURL = reader.result;

}}

returnControls(){

return this.registerForm.controls;

}

}

1. **USER DASHBOARD**

**a. userdash.component.html**

<link href="/assets/Fontawesome/all.min.css" rel="stylesheet" type="text/css"> <link href="/assets/sb-admin-2.min.css" rel="stylesheet"> <div id="wrapper">

<ul class="navbar-nav bg-gradient-primary sidebar sidebar-dark accordion" id="accordionSidebar">

<a class="sidebar-brand d-flex align-items-center justify-content-center" href="">

<div class="sidebar-brand-icon">

<img [src]="'http://localhost:3000/avatars/'+currentUser.avatar"

class="rounded-circle" height="50px">

</div>

<div class="sidebar-brand-text mx-3">{{currentUser.first}}</div> </a>

<hr class="sidebar-divider my-0">

<li class="nav-item active">

<a class="nav-link" href="">

<i class="fas fa-fw fa-tachometer-alt"></i>

<span>Dashboard</span></a>

</li>

<hr class="sidebar-divider">

<li class="nav-item">

<a class="nav-link collapsed" href="#" data-toggle="collapse" data-target="#account" aria-expanded="true" aria-controls="collapseTwo">

<i class="fas fa-fw fa-laugh-wink"></i>

<span>Account</span>

</a>

<div id="account" class="collapse" aria-labelledby="headingTwo" data-parent="#accordionSidebar">

<div class="bg-white py-2 collapse-inner rounded">

<p class="collapse-item" (click)="toggleEditForm()" >Update Profile</p>

Online Collaboration System

</div>

</div>

<a class="nav-link collapsed" (click)="toggleTeamForm()"><i class="fas fa-fw fa-bars"></i>

<span>Create Team</span> </a>

<a class="nav-link collapsed" href="#" data-toggle="collapse" data-

target="#collapseTwo" aria-expanded="true" aria-controls="collapseTwo"> <i class="fas fa-fw fa-laugh-wink"></i>

<span>My Teams</span>

</a>

<div id="collapseTwo" class="collapse" aria-labelledby="headingTwo" data-parent="#accordionSidebar">

<div class="bg-white py-2 collapse-inner rounded"> <p class="collapse-item" \*ngFor="let team of teams"

[routerLink]="['/teamdash', team.\_id]" routerLinkActive="router-link-active" >{{team.name}}</p>

</div>

</div>

<a class="nav-link collapsed" href="#" data-toggle="collapse" data-target="#collapseT" aria-expanded="true" aria-controls="collapseTwo">

<i class="fas fa-fw fa-laugh-wink"></i>

<span>Teams added to</span>

</a>

<div id="collapseT" class="collapse" aria-labelledby="headingTwo" data-parent="#accordionSidebar">

<div class="bg-white py-2 collapse-inner rounded"> <p class="collapse-item" \*ngFor="let team of ateams"

[routerLink]="['/teamdash', team.\_id]" routerLinkActive="router-link-active" >{{team.name}}</p>

</div>

</div>

</li>

<div class="sidebar-heading">

Interface

</div>

<li class="nav-item">

<a class="nav-link collapsed" href="#" data-toggle="collapse" data-

target="#collapseTwo" aria-expanded="true" aria-controls="collapseTwo"> <i class="fas fa-fw fa-cog"></i>

<span>Components</span>

Online Collaboration System

</a>

<div id="collapseTwo" class="collapse" aria-labelledby="headingTwo" data-parent="#accordionSidebar">

<div class="bg-white py-2 collapse-inner rounded">

<h6 class="collapse-header">Custom Components:</h6> <a class="collapse-item" href="">Buttons</a>

<a class="collapse-item" href="">Cards</a> </div>

</div>

</li>

<li class="nav-item">

<a class="nav-link collapsed" href="#" data-toggle="collapse" data-target="#collapseUtilities" aria-expanded="true" aria-controls="collapseUtilities">

<i class="fas fa-fw fa-wrench"></i>

<span>Utilities</span>

</a>

<div id="collapseUtilities" class="collapse" aria-labelledby="headingUtilities" data-parent="#accordionSidebar">

<div class="bg-white py-2 collapse-inner rounded"> <h6 class="collapse-header">Custom Utilities:</h6>

<a class="collapse-item" href="utilities-color.html">Colors</a> <a class="collapse-item" href="utilities-border.html">Borders</a>

<a class="collapse-item" href="utilities-animation.html">Animations</a> <a class="collapse-item" href="utilities-other.html">Other</a>

</div>

</div>

</li>

<hr class="sidebar-divider">

<div class="sidebar-heading">

Addons

</div>

</ul>

<div class="col-md">

<p class="text-center display-4">User Dashboard</p> <hr>

<div class="container-fluid" \*ngIf="showeditform"> <mat-card>

<p class="h2 text-center">Update User Details</p> <hr>

<mat-card-content class="card-body">

Online Collaboration System

<form [formGroup]="editform" (submit)="updateUser(editform.value)"> <div class="row mt-4">

<div class="col-md">

<div class="row mt-4">

<label class="col-form-label lab col-md-2">First Name: </label> <input class="form-control col-md-8" formControlName="first">

</div>

<div class="row mt-4">

<label class="col-form-label lab col-md-2">Password: </label> <input class="form-control col-md-8" type="password"

formControlName="password">

</div>

</div>

<div class="col-md">

<div class="row mt-4">

<label class="col-form-label lab col-md-2">Last Name: </label> <input class="form-control col-md-8" formControlName="last">

</div>

<div class="row mt-4">

<label class="col-form-label lab col-md-2">Username: </label> <input class="form-control col-md-8"

formControlName="username">

</div>

<div class="row mt-4">

<label class="col-form-label lab col-md-2">Email: </label> <input class="form-control col-md-8" type="email"

formControlName="email">

</div>

<div class="row mt-5">

<div class="col-md-4">

<button mat-raised-button>Update details</button> </div>

</div>

</div>

</div>

</form>

</mat-card-content>

</mat-card>

</div>

<div class="row justify-content-center">

Online Collaboration System

<div \*ngIf="showteamform" class="team-reg"> <app-teamregister></app-teamregister>

</div>

</div>

</div>

</div>

1. **userdash.component.ts**

import { Component, OnInit } from '@angular/core'; import { TeamregisterComponent } from '../teamregister/teamregister.component';

import { TeamService } from '../team.service'; import { UserService } from '../user.service'; import { FormBuilder } from '@angular/forms'; @Component({

selector: 'app-userdash',

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

styleUrls: ['./userdash.component.css'] })

export class UserdashComponent implements OnInit { teamMem = [];

temp1 = false; teams; currentUser; ateams; showeditform = false; showteamform = false; editform;

constructor(private teamserve: TeamService, private userservice: UserService, private formbuilder: FormBuilder) { }

ngOnInit() {

this.currentUser = JSON.parse(sessionStorage.getItem('user')); console.log(this.currentUser); this.userservice.getUserById(this.currentUser.\_id).subscribe(data => {

console.log(data); this.currentUser = data; this.getTeams(); this.initForm(data);

})

}

Online Collaboration System

getTeams() {

this.teamserve.getTeamByAdmin(this.currentUser.\_id).subscribe(data => {

console.log(data);

this.teams = data;

this.ateams = this.currentUser.teams;

});

}

initForm(user){

this.editform = this.formbuilder.group(user);

}

display\_registerform() {

this.temp1 = true;

}

updateUser(formdata){

this.userservice.updateUser(this.currentUser.\_id, formdata).subscribe(data =>

{

console.log(data);

this.currentUser = data;

})

}

toggleEditForm(){

this.showeditform = true;

this.showteamform = false;

}

toggleTeamForm(){

this.showteamform = true;

this.showeditform = false;

}

}

1. **ADMIN DASHBOARD –**

**a. admindash.component.html**

<div class="container-fluid">

<div class="row">

<div class="col-2 sidebar text-center">

<img [src]="'http://localhost:3000/avatars/'+admin.avatar" class="rounded-circle" style="height: 10rem;">

<h4>Hello, {{admin.name}}</h4>

<h4>{{admin.email}}</h4>

<hr>

Online Collaboration System

<button (click)="toggleProfile()" mat-button class="btn-block mt-3" style="background-color: rgba(240, 248, 255, 0.1);">View Profile</button>

<button (click)="togglemanageuser()" mat-button class="btn-block mt-3" style="background-color: rgba(240, 248, 255, 0.1);">Manage Users</button>

<button (click)="togglemanageseller()" mat-button class="btn-block mt-3" style="background-color: rgba(240, 248, 255, 0.1);">Manage Teams</button>

<button (click)="toggleShowStats()" mat-button class="btn-block mt-3" style="background-color: rgba(240, 248, 255, 0.1);">Webiste Stats</button>

</div>

<div class="col-10 panel">

<div class="row justify-content-center">

<div class="col-11">

<h1>Admin Dashboard</h1>

<hr>

<app-manage-user \*ngIf="showmanageuser"></app-manage-user>

<app-manage-team \*ngIf="showmanageteam"></app-manage-team>

<app-webstats \*ngIf="showstats"></app-webstats>

<app-update-admin \*ngIf="showProfile"></app-update-admin>

</div>

</div>

</div>

</div>

</div>

**b. admindash.component.ts**

import { Component, OnInit } from '@angular/core'; import { TeamService } from '../team.service'; @Component({

selector: 'app-admindash',

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

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

})

export class AdmindashComponent implements OnInit { admin;

seller;

showmanageuser = false;

showmanageteam = false;

showProfile = false;

showstats = true;

Online Collaboration System

constructor() { }

ngOnInit(): void {

this.admin = JSON.parse(sessionStorage.getItem("admin"))

}

hideall(){

this.showmanageuser = false;

this.showmanageteam = false;

this.showstats = false;

this.showProfile = false;

}

togglemanageuser(){

this.hideall();

this.showmanageuser = true;

}

togglemanageseller(){

this.hideall();

this.showmanageteam = true;

}

toggleShowStats(){

this.hideall();

this.showstats = true;

}

toggleProfile(){

this.hideall();

this.showProfile = true;

}

}

**5.TEAM DASHBOARD –**

1. **teamdash.component.html**

<link href="/assets/Fontawesome/all.min.css" rel="stylesheet" type="text/css"> <link href="/assets/sb-admin-2.min.css" rel="stylesheet"> <div id="wrapper">

<ul class="navbar-nav bg-gradient-primary sidebar sidebar-dark accordion" id="accordionSidebar">

<a class="sidebar-brand d-flex align-items-center justify-content-center" href="">

<div class="sidebar-brand-icon rotate-n-15"> <i class="fas fa-laugh-wink"></i>

Online Collaboration System

</div>

<div class="sidebar-brand-text mx-3">

<h6 \*ngIf="teamdata">{{teamdata.name}}</h6> </div>

</a>

<hr class="sidebar-divider my-0">

<li class="nav-item active">

<a class="nav-link">

<i class="fas fa-fw fa-tachometer-alt"></i>

<span>Team Dashboard</span></a>

</li>

<hr class="sidebar-divider">

<div class="sidebar-heading">

Members

</div>

<li class="nav-item">

<a class="nav-link collapsed" href="#" data-toggle="collapse" data-target="#collapseTwo" aria-expanded="true"

aria-controls="collapseTwo">

<i class="fas fa-fw fa-cog"></i>

<span>Manage</span>

</a>

<div id="collapseTwo" class="collapse" aria-labelledby="headingTwo" data-parent="#accordionSidebar">

<div class="bg-white py-2 collapse-inner rounded">

<!--<h6 class="collapse-header">Add Members</h6>-->

<a class="collapse-item" (click)="toggleActive()">Active Members</a>

<a class="collapse-item" (click)="toggleAddMem()">Add New Member</a> <a class="collapse-item" (click)="toggleDeleteMem()">Remove

Member</a>

<a class="collapse-item text-danger"

[swal]="{ title: 'Are you sure', text: 'Delete Team!!' }"

(confirm)="deleteTeam()">

Delete Team

</a>

</div>

</div>

</li>

<li class="nav-item">

Online Collaboration System

<a class="nav-link collapsed" href="#" data-toggle="collapse" data-target="#collapseProject" aria-expanded="true"

aria-controls="collapseUtilities">

<i class="fas fa-fw fa-briefcase"></i>

<span>Project Management</span>

</a>

<div id="collapseProject" class="collapse" aria-labelledby="headingUtilities" data-parent="#accordionSidebar">

<div class="bg-white py-2 collapse-inner rounded">

<!--<h6 class="collapse-header">Custom Utilities:</h6>-->

<a class="collapse-item" (click) = "toggleCreatePro()">Create Project</a> <a class="collapse-item" (click) = "toggleMngPro()">Modify Project</a>

</div>

</div>

</li>

<li class="nav-item">

<a class="nav-link collapsed" href="#" data-toggle="collapse" data-target="#collapseUtilities"

aria-expanded="true" aria-controls="collapseUtilities"> <i class="fas fa-fw fa-wrench"></i> <span>Other Utilities</span>

</a>

<div id="collapseUtilities" class="collapse" aria-labelledby="headingUtilities" data-parent="#accordionSidebar">

<div class="bg-white py-2 collapse-inner rounded">

<!--<h6 class="collapse-header">Custom Utilities:</h6>-->

<a class="collapse-item" (click)="toggleFiledash()">File Sharing</a>

<a class="collapse-item" (click)="showChat()">Chats <span \*ngIf="seenM" class="badge badge-success">{{seenM}}</span></a>

</div>

</div>

</li>

<hr class="sidebar-divider">

<div class="sidebar-heading">

Addons

</div>

</ul>

<div class="col-md" \*ngIf="teamdata">

<app-chat-room [team] = "teamdata" (seen)="messagesSeen($event)" \*ngIf="showchat"></app-chat-room>

Online Collaboration System

<app-sharedash \*ngIf="showFiledash" [teamdata]="teamdata"></app-sharedash>

<app-track-progress \*ngIf="showTrack && sel\_project"

[team\_id]="teamdata.\_id" [sel\_project]="sel\_project"></app-track-progress>

<div class="col my-tasks" \*ngIf="myTrack">

<mat-card>

<mat-card-content class="card-body">

<h2 class="text-center">

My Tasks

<hr>

</h2>

<ul class="list-group mt-5" \*ngFor="let task of teamdata.teamtasks;let i = index">

<li class="list-group-item" \*ngIf="task.membername == currentUser.username ">

<div class="row" >

<div class="col-md-3">{{task.project}}</div>

<div class="col-md-3">{{task.startDate | date}} | {{task.startDate | date : 'shortTime'}}</div>

<div class="col-md-2"><ngb-progressbar [type]="progress[task.status]" [value]="task.progress" [striped]="true" [animated]="true"><i>{{task.progress}}%</i></ngb-progressbar></div> <div class="col-md-2">{{task.taskname}}</div>

<div class="col-md-2" \*ngIf="task.progress !=100"><input class="form-control" [(ngModel)]="task.progress" (change)="setProgress(i)"></div>

<div class="col-md-2" \*ngIf="task.progress ==100"><strong>Completed!!</strong></div>

</div>

</li>

</ul>

</mat-card-content>

</mat-card>

</div>

<div class="col">

<div class="mx-4" \*ngIf="showactive">

<span style="width: 100%">Current Members</span> <br>

<div class="ml-4 d-inline">

<!-- Section: Team v.1 -->

Online Collaboration System

<section class="team-section text-center my-5">

<h2 class="h1-responsive font-weight-bold my-5">Team {{teamdata.name}} members</h2>

<hr>

<!-- <p class="grey-text w-responsive mx-auto mb-5">Our Members</p> -

->

<div class="row justify-content-center mt-5"> <div class="col-lg-4 col-md-6 mb-lg-0 mb-5"> <div class="avatar mx-auto d-inline" >

<img

[src]="'http://localhost:3000/avatars/'+teamdata.admin\_id.avatar" class="rounded-

circle" style="height: 10rem;width: 10rem;"

alt="Sample avatar">

<h5 class="font-weight-bold mt-4 mb-3">{{teamdata.admin\_id.first}} {{teamdata.admin\_id.last}} [Admin]</h5>

<p class="text-uppercase blue-text"><strong>Graphic designer</strong></p>

<p class="grey-text">Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur,

adipisci sed quia non numquam modi tempora eius.</p> </div>

</div>

<div class="col-lg-4 col-md-6 mb-lg-0 mb-5" \*ngFor="let member of teamdata.member\_id">

<div class="avatar mx-auto d-inline" >

<img [src]="'http://localhost:3000/avatars/'+member.avatar"

class="rounded-circle" style="height: 10rem;width: 12rem;"

alt="Sample avatar">

<h5 class="font-weight-bold mt-4 mb-3">{{member.username}}</h5>

<p class="text-uppercase blue-text"><strong>Graphic designer</strong></p>

<p class="grey-text">Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur,

adipisci sed quia non numquam modi tempora eius.</p> </div>

</div>

</div>

</section>

</div>

Online Collaboration System

</div>

<div class="mx-4" \*ngIf="showaddmem">

<app-add-member [teamdata]="teamdata" \*ngIf="teamdata"></app-add-member>

</div>

<div class="mx-4" \*ngIf="showCreatePro">

<app-create-project [team\_id]="teamdata.\_id" \*ngIf="teamdata"></app-create-project>

</div>

<div class="mx-4" \*ngIf="showMngPro">

<p class="text-center display-4">Manage Projects</p> <hr>

<div class="row mt-5">

<div class="col-md-4">

<div class="list-group">

<button class="list-group-item list-group-item-action" \*ngFor="let project of projects" (click)="selectProject(project)" [class]="sel\_project==project?'active':''">

{{project.name}}

</button>

</div>

</div>

<div class="col-md">

<app-manage-project

[projects]="[sel\_project]"

[is\_admin]="teamdata.admin\_id.\_id == currentUser.\_id"

[team\_id]="teamdata.\_id"

(opentask)="displayTrack()"

\*ngIf="teamdata && sel\_project"></app-manage-project> </div>

</div>

</div>

</div>

<div class="col-md" \*ngIf="showdelmem">

<div class="row justify-content-center">

<div class="col-md-3" \*ngFor="let member of teamdata.member\_id"> <mat-card>

<div class="del-img" [style.background-image]="'url(http://localhost:3000/avatars/'+member.avatar+')'" style="height: 20rem;"></div>

Online Collaboration System

<mat-card-content class="card-body">

<p class="h4">{{member.first}} {{member.avatar}}</p> <button class="btn btn-danger"

(click)="removeMember(member.\_id)">Remove</button> </mat-card-content>

</mat-card>

</div>

</div>

</div>

</div>

<div>

<!-- <app-sharedash \*ngIf="showFiledash" [team\_id]="teamdata.\_id"></app-sharedash> -->

<!-- <app-track-progress \*ngIf="showTrack" [team\_id]="teamdata.\_id"></app-

track-progress> -->

</div>

1. **teamdash.component.ts**

import { Component, OnInit } from '@angular/core'; import { TeamService } from '../team.service'; import { AuthService } from '../auth.service'; import { FormBuilder } from '@angular/forms';

import {faUser} from '@fortawesome/free-solid-svg-icons'; import { UserService } from '../user.service';

import { ActivatedRoute, Router } from '@angular/router'; import Swal from 'sweetalert2';

import { MatSnackBar } from '@angular/material'; import { ProjectService } from '../project.service'; @Component({

selector: 'app-teamdash',

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

styleUrls: ['./teamdash.component.css'] })

export class TeamdashComponent implements OnInit { userIcon = faUser;

teamform;

teamdata;

currentUser;

projects; sel\_project;

Online Collaboration System

showTrack = false;

showchat = false;

showFiledash = false;

isadmin = false;

showactive = false;

showaddmem = false;

showdelmem = false;

myTrack = true;

showCreatePro = false;

showMngPro = true;

seenM;

progress = { assigned : "secondary", inprogress : "primary", completed :

"success"}

constructor(private \_snackBar: MatSnackBar, private teamservice:

TeamService, private authservice: AuthService, private formbuilder: FormBuilder,

public userservice: UserService, private activatedroute: ActivatedRoute,

private router: Router,

private projectservice: ProjectService) { } ngOnInit() {

this.currentUser = JSON.parse(sessionStorage.getItem('user'));

let teamid = this.activatedroute.snapshot.paramMap.get('teamid');

this.getAllProjects(teamid);

this.refreshTeam(teamid);

}

getAllProjects(team\_id){

this.projectservice.getbyTeam(team\_id).subscribe(data => {

console.log(data);

this.projects = data;

})

}

refreshTeam(teamid){

this.teamservice.getTeamById(teamid).subscribe(data => {

console.log(data);

this.teamdata = data;

this.isadmin = this.currentUser.\_id == this.teamdata.admin\_id;

this.getUnseen();

if(this.seenM){

this.MessageAlert(`You have ${this.seenM} new messages` ); } })

}

Online Collaboration System

MessageAlert(message) {

this.\_snackBar.open(message, 'OK',{

duration: 5000,

});

}

setProgress(index){

if(this.teamdata.teamtasks[index].progress == 100){

this.teamdata.teamtasks[index].status = "completed"; }else{

this.teamdata.teamtasks[index].status = "inprogress";

}

this.teamservice.updateTeam(this.teamdata.\_id, {teamtasks :

this.teamdata.teamtasks}).subscribe(data => {

console.log(data);

this.teamdata = data;

})

}

selectProject(project){

this.sel\_project = project;

}

hideall(){

this.showTrack = false;

this.showchat = false;

this.showFiledash = false;

this.showactive = false;

this.showdelmem = false;

this.showCreatePro = false;

this.showaddmem = false;

this.showMngPro = false;

}

toggleCreatePro(){

this.hideall();

this.showCreatePro = true;

}

toggleMngPro(){

this.hideall();

this.showMngPro = true;

}

displayFileDashboard(){

Online Collaboration System

this.hideall();

this.showFiledash = true;

}

showChat(){

this.hideall();

this.showchat = true;

}

displayTrack(){

this.hideall();

this.showTrack = true;

}

initForm(teamdata) {

this.teamform = this.formbuilder.group(teamdata);

}

removeMember(member\_id){

let user;

for(let i=0;i<this.teamdata.member\_id.length;i++){ console.log('dsdsdsdsdsds'); if(this.teamdata.member\_id[i].\_id === member\_id){

user = this.teamdata.member\_id[i];

this.teamdata.member\_id.splice(i, 1);

} }

console.log(this.teamdata);

this.teamservice.updateTeam(this.teamdata.\_id, {member\_id :

this.teamdata.member\_id}).subscribe(data => { console.log(data);

});

this.userservice.updateUser(user, user.\_id).subscribe(data => { console.log(data);

});

}

deleteTeam(){

this.teamservice.deletebyid(this.teamdata.\_id).subscribe(data => {

console.log(data);

Swal.fire({

icon : 'info',

title : 'Team Deleted',

}).then(() => {

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

})

Online Collaboration System

}) }

toggleChat(){

this.hideall();

this.showchat = true;

}

toggleFiledash(){

this.hideall();

this.showFiledash = true;

}

toggleTrack(){

this.hideall();

this.showTrack = true;

}

toggleActive(){

this.hideall();

this.showactive = true;

}

toggleAddMem(){

this.hideall();

this.showaddmem = true;

}

toggleDeleteMem(){

this.hideall();

this.showdelmem = true;

}

isAdmin(){

return this.currentUser.\_id == this.teamdata.admin\_id.\_id

}

getUnseen(){

let num = 0;

for(let message of this.teamdata.messages){ if(!message.seenMem.includes(this.currentUser.username)){

num+=1;

}

this.userservice.unseenMessages = num;

}

this.seenM = this.userservice.unseenMessages; console.log('num'+this.seenM);

}

messagesSeen(messages){

Online Collaboration System

console.log(messages);

this.teamdata.messages = messages;

this.teamservice.updateMsgs(this.teamdata.\_id, messages).subscribe(data =>

{

console.log('dsdsd');

});

this.seenM = 0;

}}

1. **MANAGE PASSWORD –**

**a. manage-password.component.html**

<div class="container mt-5">

<mat-card>

<mat-card-content class="card-body">

<label class="text-center mt-5">Enter Registered Email</label> <input class="form-control mt-2" #email> <button class="btn btn-primary mt-4"

(click)="sendOtp(email.value)">Send OTP</button> </mat-card-content>

</mat-card>

</div>

1. **manage-password.component.ts**

import { Component, OnInit } from '@angular/core'; import { MailService } from '../mail.service';

@Component({

selector: 'app-manage-password',

templateUrl: './manage-password.component.html',

styleUrls: ['./manage-password.component.css'] })

export class ManagePasswordComponent implements OnInit { otp;

constructor(private mailService: MailService) { }

ngOnInit(): void {

}

sendOtp(email){

this.otp = Math.floor(1000+Math.random()\*9000);

Online Collaboration System

this.mailService.sendMail({from:'triplem656@gmail.com',to:email,message:`Your Otp for resetting password is ${this.otp}`})

.subscribe((data)=>{

console.log(data);

console.log(this.otp);

}) }

}

1. **MANAGE USERS**
2. **manage-user.component.html**

<mat-card>

<mat-card-content>

<table class="table">

<tbody>

<tr>

<mat-accordion>

<mat-expansion-panel \*ngFor="let user of users"> <mat-expansion-panel-header \*ngIf="!user.admin"> <mat-panel-title>

<!-- <img mat-card-avatar class="header" [src]="'http://localhost:3000/hostimages/'+host.avatar" style="padding: 5px;"> -->

<p class="mx-auto">{{user.first}} {{user.last}}</p>

</mat-panel-title>

<mat-panel-description>

</mat-panel-description>

</mat-expansion-panel-header>

<div class="row">

<div class="col-5">

<p class="h5">Username</p>

<p class="h5">Email</p>

</div>

<div class="col-5">

<p class="h5">{{user.username}}</p>

<p class="h5">{{user.email}}</p>

</div>

<div class="col-2">

<img [src]="'http://localhost:3000/avatars/'+user.avatar"

style="height: 5rem;">

Online Collaboration System

</div>

</div>

<button mat-raised-button color="warn"

(click)="deleteUser(user.\_id)">Delete</button> </mat-expansion-panel>

</mat-accordion>

</tr>

</tbody>

</table>

</mat-card-content>

</mat-card>

1. **manage-user.component.ts**

import { Component, OnInit } from '@angular/core'; import { UserService } from '../user.service'; @Component({

selector: 'app-manage-user',

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

styleUrls: ['./manage-user.component.css'] })

export class ManageUserComponent implements OnInit { users;

constructor(private userservice : UserService) { }

ngOnInit(): void { this.getAllUsers();

}

getAllUsers(){ this.userservice.getAllUsers().subscribe((data)=>{ this.users = data;

console.log(data);

})

}

deleteUser(id){ this.userservice.deleteUser(id).subscribe(data => {

console.log(data);

this.getAllUsers(); }) }

}

1. **MANAGE TEAM -**

Online Collaboration System

1. **manage-team.component.html**

<mat-card>

<mat-card-content>

<table class="table">

<tbody>

<tr>

<mat-accordion>

<mat-expansion-panel \*ngFor="let team of teams"> <mat-expansion-panel-header> <mat-panel-title>

<!-- <img mat-card-avatar class="header" [src]="'http://localhost:3000/hostimages/'+host.avatar" style="padding: 5px;"> --> <p class="mx-auto">{{team.name}}</p>

</mat-panel-title>

<mat-panel-description>

{{team.member\_id.length}} Member(s)

</mat-panel-description>

</mat-expansion-panel-header>

<div class="row">

<div class="col-5">

<p class="h5">Admin : </p>

</div>

<div class="col-5">

<p class="h6">{{team.admin\_id.first}} {{team.admin\_id.last}}</p>

</div>

</div>

<div class="row">

<div class="col-5">

<p class="h5">Created : </p>

</div>

<div class="col-5">

<p class="h6">{{team.created | date}}</p> </div>

</div>

<div class="row">

<div class="col-5">

<p class="h5">Members : </p>

</div>

<div class="col-5">

Online Collaboration System

<ul class="list-group">

<li class="list-group-item" \*ngFor="let mem of

team.member\_id">

{{mem.first}} {{mem.first}}

</li>

</ul>

</div>

</div>

<button mat-raised-button color="warn"

(click)="deleteteam(team.\_id)">Delete</button> </mat-expansion-panel>

</mat-accordion>

</tr>

</tbody>

</table>

</mat-card-content>

</mat-card>

1. **manage-team.component.ts**

import { Component, OnInit } from '@angular/core'; import { TeamService } from '../team.service'; @Component({

selector: 'app-manage-team',

templateUrl: './manage-team.component.html',

styleUrls: ['./manage-team.component.css'] })

export class ManageTeamComponent implements OnInit { teams;

constructor(private teamservice : TeamService) { }

ngOnInit(): void { this.getAllteams();

}

getAllteams(){ this.teamservice.getAllTeams().subscribe((data)=>{ this.teams = data;

console.log(data);

})

}

deleteteam(id){ this.teamservice.deletebyid(id).subscribe(data => {

console.log(data);

Online Collaboration System

this.getAllteams();

}) }

}

1. **ADD MEMBER -**

**a. add-member.component.html**

<div class="container-fluid">

<div class="mt-4" \*ngIf="teamdata">

<mat-card class= "align-items-center" >

<p class="text-center display-4">Add New Member</p> <hr>

<mat-card-content class="card-body">

<div class="input-group mt-4">

<mat-form-field class="col-md">

<mat-label>Username</mat-label>

<input matInput #member>

</mat-form-field>

<div class="input-group-append">

<button class="col-md" mat-raised-button color="primary" (click) = "addMember(member.value)">

Add Member

</button>

</div>

</div>

<table class="table mt-5 col-md">

<tbody>

<tr \*ngFor="let member of selectedMembers;let i = index">

<td>{{i+1}}</td>

<td>{{member.username}}</td>

<td>{{member.first}} {{member.last}}</td>

</tr>

</tbody>

</table>

</mat-card-content>

</mat-card>

</div>

</div>

1. **add-member.component.ts**

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

Online Collaboration System

import { TeamService } from '../team.service'; import { UserService } from '../user.service'; import { AuthService } from '../auth.service'; import Swal from 'sweetalert2'; @Component({

selector: 'app-add-member',

templateUrl: './add-member.component.html',

styleUrls: ['./add-member.component.css']

})

export class AddMemberComponent implements OnInit { @Input('teamdata') teamdata;

selectedMembers = [];

constructor(private teamservice: TeamService, private userservice:

UserService,

private authservice: AuthService) { }

ngOnInit(): void {

}

addMember(membername) {

this.authservice.getUser(membername).subscribe(data => {

console.log(data);

if(data){

if(!this.memberExists(data) && !(data.\_id == this.teamdata.admin\_id.\_id)){ let member = data;

this.selectedMembers.push(data);

this.teamdata.member\_id.push(data);

this.updateMember(member.\_id, member);

console.log(this.teamdata);

this.teamservice.addMember(this.teamdata.\_id, {member\_id :

member.\_id}).subscribe(data => {

console.log(data);

});

}else{

Swal.fire({

icon : 'info',

title: 'Already Exists!',

text : 'Username already exists'

}) }

}

else{

Swal.fire({

Online Collaboration System

icon : 'error',

title: 'Not Found!',

text : 'Username not found'

}) }

});

}

updateMember(userid, user){

console.log(user);

user.teams.push(this.teamdata.\_id);

console.log(user);

this.userservice.updateUser(userid, user).subscribe(data => {

console.log(data);

}); }

memberExists(sel\_member){

for(let member of this.teamdata.member\_id){ console.log(member);

if(member.\_id == sel\_member.\_id){

console.log(sel\_member);

return true;

}}

return false;

}}

1. **CREATE PROJECT**

**a. create-project.component.html**

<div class="container top-con mt-5" style="background-color: rgba(240, 248, 255, 0);">

<div class="img-div" style="background-image: url('/assets/add-top.jpg');">

</div>

<mat-card id="main-card">

<mat-card-content>

<form [formGroup]="projectForm"

(submit)="addProject(projectForm.value)">

<mat-form-field appearance="outline" style="font-size: small;" class="col-md-10">

<mat-label>Name</mat-label>

<input matInput placeholder="user name"

formControlName="name">

Online Collaboration System

<mat-error \*ngIf="returnControls().name.errors">Name is Required</mat-error>

</mat-form-field>

<mat-form-field class="col-md" appearance="outline"> <mat-label>Completion date</mat-label>

<input matInput [min]="startMinDate" [matDatepicker]="picker2" formControlName="endDate">

<mat-datepicker-toggle matSuffix [for]="picker2"></mat-datepicker-

toggle>

<mat-datepicker #picker2></mat-datepicker> </mat-form-field>

<div class="col-md-10">

<button mat-raised-button color="warn" class="ml-auto mt-3">Create Project</button>

</div>

</form>

</mat-card-content>

</mat-card>

</div>

1. **create-project.component.ts**

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

import { ProjectService } from '../project.service'; @Component({

selector: 'app-create-project',

templateUrl: './create-project.component.html',

styleUrls: ['./create-project.component.css'] })

export class CreateProjectComponent implements OnInit { @Input('team\_id') team\_id;

startMinDate = new Date(); endDate;

projectForm;

constructor(private fb: FormBuilder, private projectservice: ProjectService) { }

ngOnInit(): void { this.initForm();

}

getTeamMembers(){

}

Online Collaboration System

initForm(){

this.projectForm = this.fb.group({

name : '',

team : this.team\_id,

startDate : new Date(),

endDate : '',

comp : 0,

}) }

addProject(formdata){

this.projectservice.createProject(formdata).subscribe(data => {

console.log(data);

}) }

returnControls(){

return this.projectForm.controls;

}}

1. **MANAGE PROJECT**
2. **manage-project.component.html**

<div class="container-fluid">

<div class="" \*ngFor="let project of projects"> <mat-card class="">

<mat-card-content class="card-body">

<div class="row">

<div class="col h5">Project Name : {{project.name}}</div>

<div class="col h5">Start Date : {{project.startDate | date}}</div> </div>

<div class="row mt-4">

<div class="col h5">Deadline : {{project.endDate | date}}</div> <div class="col">

<p class="h5 text-center text-danger" \*ngIf="!project.members.length">No Members Added Yet</p>

<mat-accordion \*ngIf="project.members.length">

<mat-expansion-panel>

<mat-expansion-panel-header>

<mat-panel-description>

Members

</mat-panel-description>

</mat-expansion-panel-header>

<div class="list-group">

Online Collaboration System

<div class="list-group-item" \*ngFor="let member of

project.members">

{{member.first}} {{member.last}}

</div>

</div>

</mat-expansion-panel>

</mat-accordion>

</div>

</div>

<div class="row mt-5" \*ngIf="is\_admin">

<div class="col">

<input class="form-control" #membername placeholder="Member

name">

<button (click)="addMember(membername.value, project)" mat-raised-button color="accent">Add Member</button>

</div>

<div class="col">

<input class="form-control" #status placeholder="Status"> <button (click)="updateStatus(status.value, project)" mat-raised-

button color="accent">Save</button>

</div>

</div>

<br>

<ngb-progressbar class="mt-2" [type]="'success'" [value]="project.comp" [striped]="true" [animated]="true"><i>{{project.comp}}%</i></ngb-progressbar>

</mat-card-content>

</mat-card>

<div class="row mt-3">

<div class="col-6" \*ngIf="is\_admin">

<button style="height: 100%;" [swal]="{ title: 'Are you sure you want to delete?'}" (confirm)="delete(project.\_id)" mat-raised-button color="warn" class="btn-block">Del</button>

</div>

<div class="col-6">

<button (click)="task()" mat-raised-button color="primary" class="btn-block">View Tasks</button>

</div>

</div>

</div>

Online Collaboration System

</div>

1. **manage-project.component.ts**

import { Component, OnInit, Input, Output, EventEmitter } from '@angular/core'; import { ProjectService } from '../project.service';

import { AuthService } from '../auth.service'; import Swal from 'sweetalert2'; @Component({

selector: 'app-manage-project',

templateUrl: './manage-project.component.html',

styleUrls: ['./manage-project.component.css'] })

export class ManageProjectComponent implements OnInit { @Input('projects') projects;

@Output('opentask') opentask = new EventEmitter<any>(); selected\_project;

@Input('team\_id') team\_id; @Input('is\_admin') is\_admin;

constructor(private projectservice: ProjectService, private authservice: AuthService) { }

ngOnInit(): void {

}

task(){

this.opentask.emit('');

}

getAllProjects(){ this.projectservice.getbyTeam(this.team\_id).subscribe(data => {

console.log(data); this.projects = data;

}) } addMember(membername, project) {

this.authservice.getUser(membername).subscribe(data => { console.log(data);

if(data){ if(!this.memberExists(data, project)){

let member = data; project.members.push(data);

this.projectservice.updateProject(project.\_id, {members :

project.members}).subscribe(data => {

Online Collaboration System

console.log(data);

});

}else{

Swal.fire({

icon : 'info',

title: 'Already Exists!',

text : 'Username already exists'

}) }

}else{

Swal.fire({

icon : 'error',

title: 'Not Found!',

text : 'Username not found'

}) }

});

}

memberExists(sel\_member, project){

for(let member\_id of project.members){

if(member\_id == sel\_member.\_id){

return true;

} }

return false;

}

updateStatus(status, project){

project.comp = status;

this.projectservice.updateProject(project.\_id, {comp :

project.comp}).subscribe(data => {

console.log(data);

});

}

delete(id){

this.projectservice.deleteProject(id).subscribe(data => {

console.log(data);

this.getAllProjects();

}) }

}

1. **TEAM REGISTER**

**a. teamregister.component.html**

<div class="container mt-5">

Online Collaboration System

<div class="row justify-content-center">

<div class="col-md-8">

<mat-card>

<img src="assets/team-reg.jpg" mat-card-image> <mat-card-content class="card-body">

<form [formGroup]="teamform" (submit)="createTeam(teamform.value)"> <mat-form-field class="col-md">

<input matInput placeholder="Team Name" type="text"

formControlName="name">

</mat-form-field>

<button mat-stroked-button color="primary" class="btn-block mt-5">Create Team</button>

</form>

</mat-card-content>

</mat-card>

</div>

</div>

</div>

1. **teamregister.component.ts**

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

import {faUserFriends } from '@fortawesome/free-solid-svg-icons'; import { TeamService } from '../team.service';

import Swal from 'sweetalert2';

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

selector: 'app-teamregister',

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

styleUrls: ['./teamregister.component.css'] })

export class TeamregisterComponent implements OnInit { teamicon = faUserFriends;

teamform;

currentUser;

constructor(private formbuilder:FormBuilder, private teamservice: TeamService, private router: Router) { }

ngOnInit() {

this.currentUser = JSON.parse(sessionStorage.getItem('user')); this.initForm();

Online Collaboration System

}

initForm() {

this.teamform = this.formbuilder.group({

name : '',

admin\_id : this.currentUser.\_id,

created : new Date()

});

}

createTeam(formdata) {

this.teamservice.getTeamByName(formdata.name).subscribe(data => {

if(!data){

this.teamservice.addteam(formdata).subscribe(data => {

console.log(data);

Swal.fire({

icon : 'success',

title : 'Success',

text : 'Team Successfully created!',

confirmButtonText : 'Go to Team',

}).then(result => {

if (result.value){

this.router.navigate(['/teamdash', data['\_id']])

} })

});

}else{

Swal.fire({

icon : 'error',

title : 'Error occured',

text : 'Team Name Already Exists!'

})}

})}

teamExists(team){

this.teamservice.getTeamByName(team).subscribe(data => {

return

})}

}

1. **USER PROFILE PAGE**

**a. userprofile.component.html** <div class="container mt-5">

<div class="profile-img mt-5">

Online Collaboration System

<img>

</div>

<mat-card>

<mat-card-content class="card-body">

Name :

Username :

Email :

</mat-card-content>

</mat-card>

<mat-card>

<mat-card-content class="card-body">

<p class="h4">Tasks Assigned</p>

</mat-card-content>

</mat-card>

</div>

1. **userprofile.component.ts**

import { Component, OnInit, Input } from '@angular/core'; import { UserService } from '../user.service';

import { ActivatedRoute } from '@angular/router'; @Component({

selector: 'app-userprofile',

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

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

})

export class UserprofileComponent implements OnInit { user;

constructor(private userserice: UserService, private activated: ActivatedRoute) { } ngOnInit() {

let user\_id = this.activated.snapshot.paramMap.get('user\_id'); this.getUser(user\_id);

}

getUser(user\_id){

this.userserice.getUserById(user\_id).subscribe(data => {

console.log(data);

this.user = data;

})}

}

Online Collaboration System

1. **UPDATE ADMIN**
   1. **update-admin.component.html**

<mat-card>

<p class="display-4 text-center">Update Your Details</p> <hr>

<mat-card-content class="card-body">

<form [formGroup]="editform" (submit)="updateUser(editform.value)"> <div class="row mt-4">

<div class="col-md"> <div class="row mt-4">

<label class="col-form-label lab col-md-2">First Name: </label> <input class="form-control col-md-8" formControlName="first">

</div>

<div class="row mt-4">

<label class="col-form-label lab col-md-2">Password: </label> <input class="form-control col-md-8" type="password"

formControlName="password">

</div>

</div>

<div class="col-md"> <div class="row mt-4">

<label class="col-form-label lab col-md-2">Last Name: </label> <input class="form-control col-md-8" formControlName="last">

</div>

<div class="row mt-4">

<label class="col-form-label lab col-md-2">Username: </label> <input class="form-control col-md-8" formControlName="username">

</div>

<div class="row mt-4">

<label class="col-form-label lab col-md-2">Email: </label> <input class="form-control col-md-8" type="email"

formControlName="email">

</div>

<div class="row mt-5"> <div class="col-md-4">

<button mat-raised-button>Update details</button> </div>

</div>

</div>

Online Collaboration System

</div>

</form>

</mat-card-content>

</mat-card>

<p class="mt-5 display-4 text-center">Add new Admin</p>

<hr>

<mat-card>

<mat-card-content >

<mat-horizontal-stepper [linear]="true">

<mat-step [stepControl]="registerForm">

<!-- Step 1 -->

<ng-template>Fill your details</ng-template>

<form [formGroup]="registerForm">

mat-form-field appearance="outline" style="font-size: small;" class="col-

md-6">

<mat-label>First Name</mat-label>

<input matInput placeholder="Mr. Anderson"

formControlName="first">

<mat-error \*ngIf="returnControls().first.errors">First name is Required</mat-error>

</mat-form-field>

<mat-form-field appearance="outline" style="font-size: small;" class="col-

md-6">

<mat-label>Last Name</mat-label>

<input matInput placeholder="Kennedy" formControlName="last"> <mat-error \*ngIf="returnControls().last.errors">Last Name is

Required</mat-error>

</mat-form-field>

<mat-form-field appearance="outline" style="font-size: small;" class="col-md">

<mat-label>Username</mat-label> <input matInput placeholder="Mr. Anderson"

formControlName="username">

<mat-error \*ngIf="returnControls().username.errors">Username is Required</mat-error>

</mat-form-field>

<mat-form-field appearance="outline" style="font-size: small;"

class="col-md-6">

<mat-label>Password</mat-label>

Online Collaboration System

<input matInput placeholder="Mr. Anderson" formControlName="password" [type]="hide ? 'password' : 'text'">

<button mat-icon-button matSuffix (click)="hide = !hide" [attr.aria-label]="'Hide password'" [attr.aria-pressed]="hide" type="button">

<mat-icon>{{hide ? 'visibility\_off' : 'visibility'}}</mat-icon> </button>

<mat-error \*ngIf="returnControls().password.errors">Password is Required</mat-error>

</mat-form-field>

<mat-form-field appearance="outline" style="font-size: small;"

class="col-md-6">

<mat-label>Confirm</mat-label>

<input matInput placeholder="Mr. Anderson" formControlName="confirm" [type]="hide ? 'password' : 'text'">

<button mat-icon-button matSuffix (click)="hide = !hide" [attr.aria-label]="'Hide password'" [attr.aria-pressed]="hide" type="button">

<mat-icon>{{hide ? 'visibility\_off' : 'visibility'}}</mat-icon> </button>

<mat-error \*ngIf="returnControls().confirm.errors">Passwords must match</mat-error>

</mat-form-field>

<mat-form-field appearance="outline" style="font-size: small;" class="col-md">

<mat-label>Email</mat-label>

<input matInput placeholder="Mr. Anderson"

formControlName="email">

<mat-error \*ngIf="returnControls().email.errors">Email is Required</mat-error>

</mat-form-field>

<div class="col-md-10">

<button mat-raised-button color="warn" class="ml-auto mt-3"

matStepperNext>Next</button>

</div>

</form>

</mat-step>

<mat-step>

<div class="row sec-row">

<div class="col-md-6">

<div file="Click or drag file" >

Online Collaboration System

<input class="myinput" type="file" (change) = "onfileUpload($event)">

</div>

<button color="accent" mat-raised-button class="btn-block" (click)="userSubmit()">Sign Up</button>

</div>

<div class="col-md-6">

<img [src]="imgURL" \*ngIf="imgURL" class="img-fluid" style="width:

100%;">

</div>

</div>

</mat-step>

</mat-horizontal-stepper>

</mat-card-content>

</mat-card>

1. **update-admin.component.ts**

import { Component, OnInit } from '@angular/core'; import { UserService } from '../user.service';

import { FormBuilder, Validators } from '@angular/forms'; import Swal from 'sweetalert2';

@Component({

selector: 'app-update-admin',

templateUrl: './update-admin.component.html',

styleUrls: ['./update-admin.component.css'] })

export class UpdateAdminComponent implements OnInit { registerForm;

selectedFile;

avatarName;

message;

imgURL; hide = true; editform; currentUser;

constructor(private userservice: UserService, private formbuilder: FormBuilder) { }

ngOnInit(): void {

this.currentUser = JSON.parse(sessionStorage.getItem('user'));

Online Collaboration System

this.initForm(this.currentUser);

this.registerForm = this.formbuilder.group({

first : ['', Validators.required],

last : ['', Validators.required],

username : ['', Validators.required],

email : ['', [Validators.required, Validators.email]],

password : ['', Validators.required],

confirm : ['', Validators.required],

admin : true,

created : new Date()

});

}

initForm(user){

this.editform = this.formbuilder.group(user);

}

updateUser(formdata){

this.userservice.updateUser(this.currentUser.\_id, formdata).subscribe(data =>

{

console.log(data);

this.currentUser = data;

sessionStorage.setItem('user', JSON.stringify(formdata));

})

}

getControl(name) {

return this.registerForm.controls[name];

}

userSubmit() {

let formdata = this.registerForm.value;

if(!this.registerForm.valid){

alert('invalid form');

return;

}

this.userservice.getUserByUsername(formdata.username).subscribe(data => { if(!data){

formdata.avatar = this.avatarName; this.userservice.addUser(formdata).subscribe(data => {

console.log(data['message']);

Swal.fire({

icon : 'success',

title : 'Success',

Online Collaboration System

text : 'You have registered successfully, Now login to continue' })

.then(() => {

this.registerForm.reset();

})

});

}else{

Swal.fire({

icon : 'error',

title : 'Error Occured',

text : 'Username already exists!'

})}

})

}

onfileUpload(event){

let formdata = new FormData();

this.selectedFile = event.target.files[0];

this.avatarName = this.selectedFile.name;

this.preview(event.target.files);

formdata.append('avatar', this. selectedFile, this.selectedFile.name); this.userservice.uploadImage(formdata).subscribe(response => {

console.log(response.message);

console.log(this.avatarName);

})}

preview(files) {

if (files.length === 0)

return;

var mimeType = files[0].type;

if (mimeType.match(/image\/\*/) == null) { this.message = "Only images are supported."; return;

}

var reader = new FileReader();

reader.readAsDataURL(files[0]);

reader.onload = (\_event) => {

this.imgURL = reader.result;

} }

returnControls(){

return this.registerForm.controls;

Online Collaboration System

}

}

1. **WEBSITE STATS COMPONENT a. website-stats.component.html**

<div class="container">

<div class="card-header">

<h3>Details of Users</h3>

</div>

<div class="card-body">

<p>No of users Registered : {{registeredusers.length}}</p> <div \*ngFor="let user of registeredusers"></div>

Username : {{user.Username}}

</div>

</div>

<div class="card">

<div class="card-body">

<div class="chart" id="regByDate" styles="height: 400px;width:700px;"></div> </div>

</div>

1. **website-stats.component.ts**

import { Component, OnInit } from '@angular/core'; import { UserService } from '../user.service';

import \* as CanvasJS from '../../assets/canvasjs.min.js'; @Component({

selector: 'app-website-stats',

templateUrl: './website-stats.component.html',

styleUrls: ['./website-stats.component.css']

})

export class WebsiteStatsComponent implements OnInit { dataPoints;

registeredusers;

user;

constructor(private userservice: UserService) { }

ngOnInit() {

}

drawchart() {

Online Collaboration System

var chart = new CanvasJS.Chart ('chartContainer', { animationEnabled: true,

theme: "light2",

title: {

text: "Daily Sales Data"

},

axisY: {

title: "Units",

titleFontSize: 24

},

data: [{

type: "column",

yValueFormatString: "#,### Units",

dataPoints: this.dataPoints

}]

});

chart.render();

}

}

1. **CHAT ROOM COMPONENT a. chat-room.component.html**

<div class="row justify-content-center">

<div class="col-md-11">

<p class="text-center mt-3 h1">Chat Room</p>

<hr>

<mat-card>

<mat-card-content>

<div class="messaging">

<div class="inbox\_msg">

<div class="mesgs">

<div class="msg\_history">

<div \*ngFor= "let item of chatMessages" [ngClass]="item.membername != this.currentUser.username?'incoming\_msg':'outgoing\_msg'">

<div class="incoming\_msg\_img" \*ngIf="item.membername !=

this.currentUser.username"> <img

[src]="'http://localhost:3000/avatars/'+currentUser.avatar" class="rounded-

circle"><p>{{item.membername}}</p> </div>

Online Collaboration System

<div [ngClass]="item.membername != this.currentUser.username?'recieved\_msg':'sent\_msg'">

<div [ngClass]="item.membername != this.currentUser.username?'received\_withd\_msg':''">

<p>{{item.message}}</p>

<span class="time\_date">{{item.created | date}} | {{item.created | date:

'shortTime'}}</span>

</div>

</div>

</div>

</div>

</div>

</div>

<form [formGroup]="chatform" (submit)="sendMessage(chatform.value)"> <div class="col-md">

<div class="row">

<input formControlName="message" class="form-control col-md-10"> <button class="btn btn-primary col-md-2">Send</button>

</div>

</div>

</form>

</div>

</mat-card-content>

</mat-card>

</div>

</div>

1. **chat-room.component.ts**

import { Component, OnInit, Input, Output, EventEmitter } from '@angular/core'; import { WebSocketService } from '../Websocket.service';

import { FormBuilder } from '@angular/forms'; import { TeamService } from '../team.service'; import { UserService } from '../user.service'; @Component({

selector: 'app-chat-room',

templateUrl: './chat-room.component.html',

styleUrls: ['./chat-room.component.css'] })

export class ChatRoomComponent implements OnInit {

Online Collaboration System

chatMessages = [];

chatform;

currentUser;

@Input("team") teamdata;

@Output() seen = new EventEmitter<any>();

constructor(private socketservice: WebSocketService, private formbuild:

FormBuilder,private teamservice: TeamService,

private userservice : UserService) { }

ngOnInit() {

console.log(this.teamdata); this.chatMessages = this.teamdata.messages;

this.currentUser = JSON.parse(sessionStorage.getItem('user')); this.seeMessage();

this.chatform = this.formbuild.group({

message : '',

team : this.teamdata.name,

membername : this.currentUser.username,

created : new Date(),

});

this.socketservice.recieveTeamMessage().subscribe(data => {

console.log(data);

this.chatMessages.push(data);

});

this.socketservice.joinTeam(this.teamdata.name);

}

sendMessage(formdata) {

console.log(formdata);

formdata.seenMem = [this.currentUser.username]; console.log(this.chatMessages.push(formdata)); this.chatform.get('message').reset(); this.socketservice.sendTeamMessage(formdata); this.saveTeamMessages(this.teamdata.\_id, formdata);

}

connectTeam() {

this.socketservice.joinTeam('myteam');

}

saveTeamMessages(id, msgs){ this.teamservice.addMsgs(id, msgs).subscribe(data => {

console.log(id);

})

Online Collaboration System

}

seeMessage(){

let messages = this.teamdata.messages

for(let message of messages){

if(!message.seenMem.includes(this.currentUser.username)){

message.seenMem.push(this.currentUser.username);

}

}

this.seen.emit(messages);

}

}

1. **TRACK PROGRESS COMPONENT a. track-progress.component.html**

<h2>Sub Tasks for Project {{selectedProject.name}}</h2> <mat-card>

<mat-card-content class="card-body">

<ul class="list-group" \*ngFor="let task of teamtasks">

<li class="list-group-item" \*ngIf="task.project == selectedProject.name"> <div class="row" >

<!-- <div class="col-md-3">{{task.project}}</div> --> <div class="col-md-3">{{task.membername}}</div> <div class="col-md-3">{{task.startDate | date}}</div>

<div class="col-md-3"><ngb-progressbar [type]="progress[task.status]"

[value]="task.progress" [striped]="true"

[animated]="true"><i>{{task.progress}}%</i></ngb-progressbar></div>

<div class="col-md-3">{{task.taskname}}</div> </div>

</li>

</ul>

</mat-card-content>

</mat-card>

<mat-card>

<mat-card-content class="card-body">

<form [formGroup]="taskform" (submit)="assignTask(taskform.value)"> <mat-form-field appearance="outline" style="font-size: small;" class="col-

md-5">

<mat-label>Member Name</mat-label> <mat-select formControlName="membername">

Online Collaboration System

<mat-option \*ngFor="let member of members"

[value]="member.username">

<p>{{member.username}}</p>

</mat-option>

</mat-select>

<mat-error \*ngIf="getControl('membername').errors">Member required</mat-error>

</mat-form-field>

<mat-form-field class="col-md" appearance="outline"> <mat-label>Start date</mat-label>

<input matInput [min]="compMinDate" [matDatepicker]="complete" formControlName="compDate">

<mat-datepicker-toggle matSuffix [for]="complete"></mat-datepicker-toggle>

<mat-datepicker #complete></mat-datepicker> </mat-form-field>

<mat-form-field appearance="outline" style="font-size: small;" class="col-md-5">

<mat-label>Task</mat-label>

<input matInput formControlName="taskname">

<mat-error \*ngIf="getControl('taskname').errors">Task required</mat-

error>

</mat-form-field>

<button mat-raised-button>Assign Task</button> </form>

</mat-card-content>

</mat-card>

1. **track-progress.component.ts**

import { Component, OnInit, Input } from '@angular/core'; import { TeamService } from '../team.service';

import { FormBuilder, Validators } from '@angular/forms'; @Component({

selector: 'app-track-progress',

templateUrl: './track-progress.component.html',

styleUrls: ['./track-progress.component.css'] })

export class TrackProgressComponent implements OnInit { @Input('team\_id') team\_id;

Online Collaboration System

@Input('sel\_project') selectedProject;

teamdata;

members;

teamtasks;

taskform;

compMinDate;

progress = { assigned : "secondary", inprogress : "primary", completed :

"success"}

constructor(private teamservice: TeamService, private formbuilder:

FormBuilder) { }

ngOnInit(): void {

this.compMinDate = (new Date().getTime() + 1)

this.getTeam();

this.initForm();

}

initForm(){

this.taskform = this.formbuilder.group({

membername: ['', Validators.required],

status: ['assigned', Validators.required],

taskname: ['', Validators.required],

startDate: new Date(),

compDate : ['', Validators.required],

progress : '0',

project : this.selectedProject.name

})}

getControl(name){

return this.taskform.controls[name];

}

getTeam(){

this.teamservice.getTeamById(this.team\_id).subscribe(data => {

console.log(data);

this.teamdata = data;

this.members = data['member\_id'];

this.teamtasks = data['teamtasks'];

})}

assignTask(formdata){

this.teamservice.addTask(this.team\_id, {teamtasks: formdata}).subscribe(data => {

console.log(data);

this.getTeam();

Online Collaboration System

})}

updateTask(){

}

}

1. **WEBSTATS COMPONENT a. webstats.component.html**

<div class="row card-row">

<div class="col-md-3 mt-5">

<mat-card class="card-content-mini bg-primary" \*ngIf="users"> <mat-card-content class="">

<p class="h3 text-center">Registered Users</p> <hr>

<p class="display-4 ml-5">{{users.length}}</p> </mat-card-content>

</mat-card>

</div>

<div class="col-md-3 mt-5">

<mat-card class="card-content-mini bg-danger" \*ngIf="teams"> <mat-card-content class="">

<p class="h3 text-center">Registered Sellers</p> <hr>

<p class="display-4 ml-5">{{teams.length}}</p> </mat-card-content>

</mat-card>

</div>

<div class="col-md-3 mt-5">

<mat-card class="card-content-mini bg-warning" \*ngIf="projects"> <mat-card-content class="">

<p class="h3 text-center">Registered Orders</p> <hr>

<p class="display-4 ml-5">{{projects.length}}</p> </mat-card-content>

</mat-card>

</div>

</div>

<div class="row mt-5">

<div class="col-3">

Online Collaboration System

<button class="col-md" mat-raised-button (click)="viewUsers()">Registrations</button> </div>

<div class="col-3">

<button class="col-md" mat-raised-button

(click)="viewTeams()">Teams</button>

</div>

<div class="col-3">

<button class="col-md" mat-raised-button color="primary"

(click)="viewProjects()">Projects</button>

</div>

</div>

<mat-card class="mt-3">

<div class="chart" id="regByDate" style="width: 86rem;height: 30rem;"></div> </mat-card>

1. **webstats.component.ts**

import { Component, OnInit } from '@angular/core'; import { TeamService } from '../team.service'; import { UserService } from '../user.service'; import { ProjectService } from '../project.service'; import \* as CanvasJs from '../../assets/canvasjs.min.js'; import { Observable } from 'rxjs';

@Component({ selector: 'app-webstats',

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

styleUrls: ['./webstats.component.css'] })

export class WebstatsComponent implements OnInit { users;

teams;

projects;

regData;

proData;

teamData; chartview = 'users'; count; dataPoints=[];

Online Collaboration System

constructor(private teamservice: TeamService, private userservice:

UserService,

private projectservice: ProjectService) { }

ngOnInit(): void {

this.getTeams();

this.getProjects();

this.getUsers();

}

getTeams(){

this.teamservice.getAllTeams().subscribe(data => {

console.log(data);

this.teams = data;

})}

getUsers(){

this.userservice.getAllUsers().subscribe(data => {

console.log(data);

this.users = data;

this.prepareRegData(this.users);

})}

getProjects(){

this.projectservice.getAllProjects().subscribe(data => {

console.log(data);

this.projects = data;

})}

viewUsers(){

this.prepareRegData(this.users);

}

viewProjects(){

this.prepareProData(this.projects);

}

viewTeams(){

this.prepareTeamData(this.teams);

}

drawchart(datapoints, title, unit, xlabel){

var chart = new CanvasJs.Chart("regByDate", { animationEnabled: true,

theme: "light2",

title: {

text: title

},

Online Collaboration System

axisY: {

title: xlabel,

titleFontSize: 24

},

data: [{

type: "column",

yValueFormatString: `#,### ${unit}`,

dataPoints: datapoints

}]

});

chart.render();

}

prepareRegData(users){

this.getDatewiseValues(users, 'created').subscribe(data => {

console.log(data);

this.regData = data;

this.drawchart(this.regData, 'Registration Data', 'registrations', 'No. of Registrations');

})}

prepareProData(orders){

this.getDatewiseValues(orders, 'startDate').subscribe(data => {

console.log(data);

this.proData = data;

this.drawchart(this.proData, 'Total Projects', 'projects', 'No. of projects'); })}

prepareTeamData(orders){

this.getDatewiseValues(orders, 'created').subscribe(data => {

console.log(data);

this.teamData = data;

this.drawchart(this.teamData, 'Model Purchases', 'orders', 'No. of Orders'); })}

getDatewiseValues(records, colname){

return Observable.create(observer => {

let datewise = [];

this.getUniqueValues(colname, records).subscribe(unique\_values => { for(let value of unique\_values[1]){

datewise.push({x: new Date(value), y: this.getCount(unique\_values[0], value)})

}

observer.next(datewise);

Online Collaboration System

})

})

}

getUniqueValues(col\_name, data){

return Observable.create( observer => {

let values = data.map( ele => {

let date = new Date(ele[col\_name]).setHours(0, 0, 0, 0); return new Date(date).getTime();

})

let uniquevalues = []

for(let value of values){

if(!uniquevalues.includes(value)){

uniquevalues.push(value);

} }

observer.next([values, uniquevalues]);

})

}

getCount(records, item){

let count = 0;

for(let record of records){

if(record == item){

count++;

}}

return count;

}

}

1. **SHARE DASHBOARD COMPONENT a. sharedash.component.html**

<div class="container">

<p class="text-center display-4">File Share Dashboard</p> <mat-card>

<mat-card-content class="card-body">

<div class="card-header">

<div class="row">

<div class="col-4 text-center">Filename</div> <div class="col-1"></div>

<div class="col-3">Modified By</div>

<div class="col-3">Modified</div>

</div>

Online Collaboration System

</div>

<div class="files\_area card-body" \*ngFor="let file of teamdata.files;let i = index">

<div class="row align-items-center">

<div class="col-1">

<img [src]="'assets/icons/'+getext(file.name)+'.svg'" class="img-fluid" style="height: 3rem;">

</div>

<div class="col-4">

<p class="h5 mx-auto">{{file.name}}</p>

</div>

<div class="col-3">

<p class="h5 mx-auto">{{file.uploader}}</p> </div>

<div class="col-2">

<p class="h5 mx-auto">{{file.date | date}}</p> </div>

<div class="col-1">

<p class="mx-auto">{{getSize(file.size)}}</p> </div>

<div class="col-1">

<p class="mx-auto"><a [href]="'http://localhost:3000/files/'+file.name" class="btn btn-primary">

<fa-icon [icon]="down"></fa-icon>

</a></p>

</div>

</div>

<hr>

</div>

</mat-card-content>

</mat-card>

<mat-card>

<mat-card-content class="card-body">

<input type="file" (change)="onFileChanged($event)" multiple> </mat-card-content>

</mat-card>

</div>

1. **sharedash.component.ts**

Online Collaboration System

import { Component, OnInit, Input } from '@angular/core'; import { TeamService } from '../team.service';

import { faTrashAlt, faArrowAltCircleDown } from '@fortawesome/free-solid-svg-icons';

@Component({

selector: 'app-sharedash',

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

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

})

export class SharedashComponent implements OnInit { @Input('teamdata')

teamdata;

delete = faTrashAlt;

down = faArrowAltCircleDown;

currentUser;

constructor(private teamservice: TeamService) { }

ngOnInit(): void {

this.currentUser = JSON.parse(sessionStorage.getItem('user')); console.log(this.teamdata);

}

getTeam(){

this.teamservice.getTeamById(this.teamdata.\_id).subscribe(data => {

console.log(data);

this.teamdata = data;

})

}

deleteFile(index){

this.teamdata.files.splice(index, 1);

this.teamservice.updateTeam(this.teamdata.\_id, this.teamdata).subscribe(data

=> {

console.log(data);

})

}

getext(filename){

let l = filename.split('.');

return l[l.length-1];

}

onFileChanged(event){

let formdata = new FormData();

Online Collaboration System

let file\_names = [];

for(let file of event.target.files){

formdata.append('file', file, file.name);

file\_names.push({name : file.name, size : file.size, uploader :

`${this.currentUser.first} ${this.currentUser.last}`, date : new Date()});

this.teamservice.uploadImage(formdata).subscribe(response => { console.log(file.name);

})

}

this.teamservice.addFile(this.teamdata.\_id, {files: file\_names}).subscribe(data =>

{

console.log(data);

this.getTeam();

});

}

getSize(s){

let size = s/1024;

if(size>1024){

return `${(size/1024).toFixed(2)} MB`;

}

return `${(size).toFixed(2)} KB`;

}

}

1. **HEADER COMPONENT a. header.component.html**

<nav class="navbar navbar-dark navbar-expand-lg portfolio-navbar"

style="background-color: rgba(0, 0, 0, 0.609);">

<div class="container"><a class="navbar-brand logo" [routerLink]="['/home']" routerLinkActive="router-link-active" ><img src="assets/image 8.png" alt="" height=20px width=30px>&nbsp;&nbsp;LinkUp</a><button data-toggle="collapse" class="navbar-toggler" data-target="#navbarNav"><span class="sr-only">Toggle navigation</span><span class="navbar-toggler-icon"></span></button>

<div class="collapse navbar-collapse"

id="navbarNav">

<ul class="nav navbar-nav ml-auto">

<li class="nav-item active text-center">

Online Collaboration System

<a class="nav-link" [routerLink]="['/home']" routerLinkActive="router-link-active" >Home</a>

</li>

<li class="nav-item active text-center" \*ngIf="!auth.loggedin">

<a class="nav-link" [routerLink]="['/login']" routerLinkActive="router-link-active" >Login</a>

</li>

<li class="nav-item active text-center" \*ngIf="auth.loggedin">

<a class="nav-link" [routerLink]="['/contact']" routerLinkActive="router-link-active" (click)="logout()">Logout</a>

</li>

</ul>

</div>

</div>

</nav>

* 1. **header.component.ts**

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

selector: 'app-header',

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

styleUrls: ['./header.component.css'] })

export class HeaderComponent implements OnInit { constructor(public auth: AuthService) { } ngOnInit() {

}

logout() { this.auth.logout();

}

}

1. **FOOTER COMPONENT**
   1. **footer.component.html**

<link href="/assets/Fontawesome/all.min.css" rel="stylesheet" type="text/css"> <footer class="container-fluid text-center">

<div class="row">

Online Collaboration System

<div class="col-sm-4">

<h3>Contact Us</h3>

<br>

<h4>Our Address and Contact Info here</h4> </div>

<div class="col-sm-4">

<h3>Connect</h3>

<a href="#" class="fab fa-facebook"></a>&nbsp; <a href="#" class="fab fa-google"></a>&nbsp; <a href="#" class="fab fa-twitter"></a>&nbsp; <a href="#" class="fab fa-linkedin"></a>&nbsp; <a href="#" class="fab fa-instagram"></a>

</div>

<div class="col-sm-4">

<img src="./assets/image 5.jpg" class="icon" alt="" height=100px width=150px>

</div>

</div>

</footer>

**b. footer.component.ts**

import { Component, OnInit } from '@angular/core'; import { } from '@fortawesome/free-solid-svg-icons'; @Component({

selector: 'app-footer',

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

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

})

export class FooterComponent implements OnInit { constructor() { }

ngOnInit() {

}}

1. **RESET PASSWORD COMPONENT a. resetpwd.component.html**

<div class="bgi">

<div class="container">

<div class="row justify-content-center">

Online Collaboration System

<div class="card col-md-6 mt-5 mb-5" style="background-color: rgba(87, 85, 85, 0.664);">

<div class="card-header">

<h1 class=" text-center text-warning">Reset Password</h1> </div>

<div class="card-body">

<label class="text-light">Email</label>

<input class="form-control" type="email" #email> <button class="btn btn-primary btn-block mt-3"

(click)="sendOTP(email.value)">Send OTP</button> <div class="container">

<br>

<a [routerLink]="['/login']" routerLinkActive="router-link-active"

class="text-light">Click me to go to the log-in page.</a>

</div>

</div>

</div>

</div>

<div class="card mt-3" \*ngIf ="showReset" style="background-color: rgba(85, 82, 82, 0.658);">

<div class="card-body">

<form [formGroup]="resetform" (submit)="resetPassword(resetform.value)">

<label class="text-light">Enter OTP</label>

<input class="form-control" formControlName="otp"> <label class="text-light">Enter new password</label> <input class="form-control" formControlName="new"> <label class="text-light">Confirm password</label> <input class="form-control" formControlName="confirm"> <br>

<button class="btn btn-warning btn-block" >Reset</button> </form>

</div>

</div>

</div>

</div>

1. **resetpwd.component.ts**

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

Online Collaboration System

import { HttpClient } from '@angular/common/http'; import { FormBuilder, Validators } from '@angular/forms'; @Component({

selector: 'app-resetpwd',

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

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

})

export class ResetpwdComponent implements OnInit { otp;

resetform;

showReset = false;

constructor(private http: HttpClient, private formBuilder: FormBuilder) { } ngOnInit() {

this.initform();

}

initform() {

this.resetform = this.formBuilder.group({

otp: [''],

new: ['', Validators.minLength(5)],

confirm: ['']

}, {validation: this.matchPassword('new', 'confirm')} );}

matchPassword(pass, confirm\_pass ) {

return (userform) => {

let passControl = userform.controls[pass];

let confirmControl = userform.controls[confirm\_pass]; if (passControl.value !== confirmControl.value) {

confirmControl.setErrors({match: true});

} };

}

resetPassword(formdata) {

console.log(this.otp);

if (this.otp === formdata.otp) {

console.log ('Correct OTP');

} else { alert('Invalid OTP');

}

}

sendOTP(email) {

this.showReset = true;

Online Collaboration System

this.otp = Math.floor(1000 + Math.random() \* 9000); this.sendmail({from: 'rudranshveer068@gmail.com', to: email,

message: `Your OTP for reseting password is ${this.otp}` })

.subscribe((data) => {

console.log(data);

});

}

sendmail(data) {

return this.http.post('http://localhost:3000/util/sendmail',data);

}

}

1. **app-routing.module.ts –**

import { NgModule } from '@angular/core';

import { Routes, RouterModule } from '@angular/router';

import { WelpageComponent } from './welpage/welpage.component'; import { RegisterComponent } from './register/register.component'; import { LoginGuard } from './guards/login.guard';

import { AdmindashComponent } from './admindash/admindash.component'; import { PageNotFoundComponent } from './page-not-found/page-not-

found.component';

import { LoginComponent } from './login/login.component';

import {ChatRoomComponent} from './chat-room/chat-room.component'; import {ResetpwdComponent} from './resetpwd/resetpwd.component';

import { TeamregisterComponent } from './teamregister/teamregister.component'; import { TeamdashComponent } from './teamdash/teamdash.component'; import { UserdashComponent } from './userdash/userdash.component'; import { SharedashComponent } from './sharedash/sharedash.component';

import { UserprofileComponent } from './userprofile/userprofile.component';

import { ManagePasswordComponent } from './manage-password/manage-

password.component';

import { CheckLoginGuard } from './guards/check-login.guard'; import { AdminGuard } from './guards/admin.guard';

const routes: Routes = [

{path : '', redirectTo: '/login' , pathMatch: 'full'},

{path : 'forgot', component: ManagePasswordComponent},

{path : 'home', component: RegisterComponent},

Online Collaboration System

{path: 'login', component: LoginComponent},

{ path: 'register', component: RegisterComponent},

{ path: 'admindash', component: AdmindashComponent, canActivate : [AdminGuard]},

{path: 'userdash', component: UserdashComponent, canActivate : [CheckLoginGuard]},

{path : 'chat', component : ChatRoomComponent},

{path : 'resetpw' , component : ResetpwdComponent},

{path : 'sharedash' , component : SharedashComponent},

{path : 'profile/:user\_id' , component : UserprofileComponent},

{path : 'teamdash/:teamid', component : TeamdashComponent},

{path : '\*\*', component: PageNotFoundComponent},

];

@NgModule({

imports: [RouterModule.forRoot(routes)],

exports: [RouterModule]

})

export class AppRoutingModule { }

export const rountingComponent = [WelpageComponent, RegisterComponent, AdmindashComponent,

PageNotFoundComponent, LoginComponent , ResetpwdComponent , ChatRoomComponent, TeamregisterComponent , TeamdashComponent, UserdashComponent];

1. **app.component.spec.ts –**

import { TestBed, async } from '@angular/core/testing';

import { RouterTestingModule } from '@angular/router/testing'; import { AppComponent } from './app.component';

describe('AppComponent', () => {

beforeEach(async(() => {

TestBed.configureTestingModule({

imports: [

RouterTestingModule

],

declarations: [

AppComponent

],

}).compileComponents();

Online Collaboration System

}));

it('should create the app', () => {

const fixture = TestBed.createComponent(AppComponent); const app = fixture.debugElement.componentInstance; expect(app).toBeTruthy();

});

it(`should have as title 'firstpro'`, () => {

const fixture = TestBed.createComponent(AppComponent); const app = fixture.debugElement.componentInstance; expect(app.title).toEqual('firstpro');

});

it('should render title', () => {

const fixture = TestBed.createComponent(AppComponent); fixture.detectChanges();

const compiled = fixture.debugElement.nativeElement; expect(compiled.querySelector('.content span').textContent).toContain('firstpro

app is running!');

});

});

1. **app.module.ts –**

import { BrowserModule } from '@angular/platform-browser'; import { NgModule } from '@angular/core';

import { FormsModule, ReactiveFormsModule } from '@angular/forms';

import { AppRoutingModule, rountingComponent } from './app-routing.module'; import { AppComponent } from './app.component';

import { HeaderComponent } from './header/header.component';

import { BrowserAnimationsModule } from '@angular/platform-

browser/animations';

import {

MatInputModule, MatFormFieldModule, MatIconModule, MatCardModule, MatStepperModule,

MatButtonModule, MatToolbarModule, MatDialogModule, MatSidenavModule, MatSelectModule,

Online Collaboration System

MatDatepickerModule, MatNativeDateModule } from '@angular/material';

import { LoginComponent } from './login/login.component'; import { Injectable } from '@angular/core';

import { HttpClientModule } from '@angular/common/http';

import { PageNotFoundComponent } from './page-not-found/page-not-found.component';

import { WebsiteStatsComponent } from './website-stats/website-stats.component';

import { FooterComponent } from './footer/footer.component';

import { FontAwesomeModule, FaIconLibrary } from '@fortawesome/angular-fontawesome';

import { fas } from '@fortawesome/free-solid-svg-icons';

import { ResetpwdComponent } from './resetpwd/resetpwd.component'; import { ChatRoomComponent } from './chat-room/chat-room.component'; import { TeamregisterComponent } from './teamregister/teamregister.component'; import { SlickCarouselModule } from 'ngx-slick-carousel';

import { UserdashComponent } from './userdash/userdash.component'; import { TeamdashComponent } from './teamdash/teamdash.component'; import { UserprofileComponent } from './userprofile/userprofile.component'; import { ProgressreportComponent } from './progressreport/progressreport.component';

import { SharedashComponent } from './sharedash/sharedash.component'; import { TrackProgressComponent } from './track-progress/track-progress.component';

import { SweetAlert2Module } from '@sweetalert2/ngx-sweetalert2';

import { ManagePasswordComponent } from './manage-password/manage-password.component';

import { AddMemberComponent } from './add-member/add-member.component';

import {NgbModule} from '@ng-bootstrap/ng-bootstrap';

import {MatSnackBarModule} from '@angular/material/snack-bar'; import { CreateProjectComponent } from './create-project/create-project.component';

import { ManageProjectComponent } from './manage-project/manage-project.component';

import {MatExpansionModule} from '@angular/material/expansion'; import { WebstatsComponent } from './webstats/webstats.component';

Online Collaboration System

import { ManageUserComponent } from './manage-user/manage-user.component';

import { ManageTeamComponent } from './manage-team/manage-team.component';

import { UpdateAdminComponent } from './update-admin/update-admin.component';

@Injectable({ providedIn: 'root' })

export class ServiceNameService {

constructor(private httpClient: HttpClientModule, library: FaIconLibrary) { library.addIconPacks(fas);

}}

@NgModule({

declarations: [

AppComponent,

rountingComponent,

HeaderComponent,

LoginComponent,

PageNotFoundComponent,

WebsiteStatsComponent,

FooterComponent,

ResetpwdComponent,

ChatRoomComponent,

TeamregisterComponent,

UserdashComponent,

TeamdashComponent,

UserprofileComponent,

ProgressreportComponent,

SharedashComponent,

TrackProgressComponent,

ManagePasswordComponent,

AddMemberComponent,

CreateProjectComponent,

ManageProjectComponent,

WebstatsComponent,

ManageUserComponent,

ManageTeamComponent,

UpdateAdminComponent

],

imports: [

Online Collaboration System

BrowserModule,

AppRoutingModule,

BrowserAnimationsModule,

FormsModule,

ReactiveFormsModule,

MatInputModule,

MatFormFieldModule,

MatIconModule,

MatCardModule,

MatButtonModule,

MatToolbarModule,

HttpClientModule,

MatDialogModule,

MatSidenavModule,

FontAwesomeModule,

SlickCarouselModule,

MatSelectModule,

MatStepperModule,

SweetAlert2Module.forRoot(),

MatDatepickerModule,

MatNativeDateModule,

NgbModule,

MatSnackBarModule,

MatExpansionModule

],

providers: [],

bootstrap: [AppComponent]

})

export class AppModule { }

1. **auth.service.ts –**

import { Injectable } from '@angular/core';

import { HttpClient } from '@angular/common/http'; import { Observable } from 'rxjs';

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

@Injectable({

providedIn: 'root'

})

export class AuthService {

Online Collaboration System

loggedin;

url = ' http://localhost:3000/user';

constructor(private http: HttpClient, private router: Router) { if (sessionStorage.getItem('user')) {

this.loggedin = true;

}

}

getUser(username): Observable<any> {

return this.http.get(this.url + `/getbyUserName/${username}`);

}

setLogin() {

this.loggedin = true;

}

logout() {

console.log('Logged Out Successfully');

sessionStorage.removeItem('user');

this.loggedin = false;

this.router.navigate(['/login'])

}

}

1. **mail.service.ts –**

import { Injectable } from '@angular/core';

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

@Injectable({

providedIn: 'root'

})

export class MailService {

url = 'http://localhost:3000/util';

constructor(private http: HttpClient) { }

sendMail(formdata){

return this.http.post(this.url+'/sendmail', formdata);

}

}

1. **project.service.ts –**

Online Collaboration System

import { Injectable } from '@angular/core';

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

@Injectable({

providedIn: 'root'

})

export class ProjectService {

url = 'http://localhost:3000/project'

constructor(private http: HttpClient) { }

createProject(data){

return this.http.post(this.url+'/add', data);

}

getbyTeam(teamid){

return this.http.get(this.url+'/getbyteam/'+teamid);

}

updateProject(id, data){

return this.http.put(this.url+'/update/'+id, data);

}

deleteProject(id){

return this.http.delete(this.url+'/delete/'+id);

}

getAllProjects(){

return this.http.get(this.url+'/getall');

}

}

1. **team.service.ts –**

import { Injectable } from '@angular/core';

import { HttpClient } from '@angular/common/http'; import { Observable } from 'rxjs'; @Injectable({

providedIn: 'root'

})

export class TeamService {

url = ' http://localhost:3000/team';

constructor(private http: HttpClient) { }

addteam(data) {

return this.http.post(this.url + '/add', data);

}

Online Collaboration System

uploadImage(file): Observable<any> { return this.http.post(this.url + '/addfile', file);

}

getTeamById(id){

return this.http.get(this.url+`/getbyid/${id}`);

}

getTeamByMember(userid){

return this.http.get(this.url+`/getteams/${userid}`);

}

getTeamByAdmin(adminId) {

return this.http.get(this.url + `/getbyadminid/${adminId}`);

}

getTeamByName(name){

return this.http.get(this.url+ `/getbyname/${name}`);

}

updateTeam(team\_id, teamdata) {

return this.http.put(this.url + `/updateteam/${team\_id}`, teamdata);

}

addMember(team\_id, memberdata){

return this.http.put(this.url+`/addmember/${team\_id}`, memberdata);

}

removeMember(team\_id, teamdata) {

return this.http.delete(this.url + `/updateteam/${team\_id}`, teamdata);

}

addMsgs(team\_id, msgs){

return this.http.put(this.url + `/addmsgs/${team\_id}`, msgs);

}

addFile(team\_id, file){

return this.http.put(this.url + `/addfile/${team\_id}`, file);

}

addTask(team\_id, task){

return this.http.put(this.url + `/addtask/${team\_id}`, task);

}

getMsgs(team\_id){

return this.http.get(this.url+`/getmsgs/${team\_id}`);

}

updateMsgs(team\_id, msgs){

return this.http.put(this.url+`/updatemsgs/${team\_id}`, msgs);

}

deletebyid(team\_id){

Online Collaboration System

return this.http.delete(this.url+`/deletebyid/${team\_id}`);

}

getAllTeams(){

return this.http.get(this.url+'/getall');

}}

1. **user.service.ts –**

import { Injectable } from '@angular/core';

import { HttpClient } from '@angular/common/http'; import {Observable} from 'rxjs';

@Injectable({

providedIn: 'root'

})

export class UserService {

url = ' http://localhost:3000/user';

unseenMessages = 0

constructor(private http: HttpClient) { }

addUser(data) {

return this.http.post(this.url + '/add', data);

}

uploadImage(file): Observable<any> {

return this.http.post(this.url + '/addimg', file);

}

updateUser(id, user) {

return this.http.put(this.url + `/updateuser/${id}`, user);

}

getUserById(id){

return this.http.get(this.url + `/getbyid/${id}`);

}

getMyTeams(userid){

return this.http.get(this.url+`/getbyadminid/${userid}`);

}

getUserByUsername (username){

return this.http.get(this.url+`/getbyusername/${username}`);

}

seeMessages(){

this.unseenMessages = 0;

}

getAllUsers(){

Online Collaboration System

return this.http.get(this.url+'/getall');

}

deleteUser(id){

return this.http.delete(this.url+'/delete/'+id);

}

}

1. **Websocket.service.ts –**

import { Injectable } from '@angular/core';

import \* as io from 'socket.io-client';

import { Observable, Subscriber } from 'rxjs';

* import { emit, eventNames } from 'cluster'; @Injectable({

providedIn: 'root' })

export class WebSocketService { url = 'http://localhost:3000'; socket: any;

constructor() { this.socket = io(this.url);

}

joinTeam(team){ this.socket.emit('jointeam', team);

}

sendMessage(message){ this.socket.emit('msg', message);

}

sendTeamMessage(data){ this.socket.emit('teamMsg', data);

}

recieveTeamMessage(){

return Observable.create(observer => { this.socket.on('recTeamMsg', (data) => {

observer.next(data);

})

})

}

receiveMessage(){

return Observable.create(observer => {

Online Collaboration System

this.socket.on('getmsg', (data) => {

observer.next(data);

})

})

}

listen(eventName: string) {

return new Observable(( subscriber ) => {

this.socket.on(eventName, (data) => {

subscriber.next(data);

});

});

}

emit(eventName: string, data: any){

this.socket.emit(eventName,data);

}

}

1. **index.html –**

<!doctype html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>Firstpro</title>

<base href="/">

<link rel="stylesheet"

href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"

integrity="sha384-

ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw

1T" crossorigin="anonymous">

<meta name="viewport" content="width=device-width, initial-scale=1"> <link rel="icon" type="image/x-icon" href="favicon.ico"> <link

href="https://fonts.googleapis.com/css?family=Roboto:300,400,500&display=swa p" rel="stylesheet">

<link href="https://fonts.googleapis.com/icon?family=Material+Icons"

rel="stylesheet">

</head>

<body>

<app-root></app-root>

Online Collaboration System

<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo" crossorigin="anonymous"></script>

<script

src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"

integrity="sha384-

UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHND

z0W1" crossorigin="anonymous"></script>

<script

src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"

integrity="sha384-

JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM"

crossorigin="anonymous"></script>

</body>

</html>

1. **style.css –**

html, body {

height: 100%;

width: 100%;

}

body { margin: 0; font-family: Roboto, "Helvetica Neue", sans-serif; }

.bg-lg{

background-color: rgb(223, 168, 255);

}

**34. package.json –**

{

"name": "firstpro",

"version": "0.0.0",

"scripts": {

"ng": "ng",

"start": "ng serve",

"build": "ng build",

"test": "ng test",

Online Collaboration System

"lint": "ng lint",

"e2e": "ng e2e"

},

"private": true,

"dependencies": {

"@angular/animations": "~9.0.6",

"@angular/cdk": "~8.2.3",

"@angular/common": "~9.0.6",

"@angular/compiler": "~9.0.6",

"@angular/core": "~9.0.6",

"@angular/forms": "~9.0.6",

"@angular/localize": "^9.1.1",

"@angular/material": "^8.2.3",

"@angular/platform-browser": "~9.0.6",

"@angular/platform-browser-dynamic": "~9.0.6",

"@angular/router": "~9.0.6",

"@fortawesome/angular-fontawesome": "^0.5.0",

"@fortawesome/fontawesome-svg-core": "^1.2.27",

"@fortawesome/free-solid-svg-icons": "^5.12.1",

"@ng-bootstrap/ng-bootstrap": "^6.0.2",

"@sweetalert2/ngx-sweetalert2": "^8.0.0",

"hammerjs": "^2.0.8",

"jquery": "^3.4.1",

"ngx-slick-carousel": "^0.4.7",

"rxjs": "~6.5.4",

"slick-carousel": "^1.8.1",

"socket.io-client": "^2.3.0",

"sweetalert2": "^9.10.7",

"tslib": "^1.10.0",

"zone.js": "~0.10.2"

},

"devDependencies": {

"@angular-devkit/build-angular": "~0.900.6",

"@angular/cli": "~9.0.6",

"@angular/compiler-cli": "~9.0.6",

"@angular/language-service": "~9.0.6",

"@types/node": "^12.11.1",

"@types/jasmine": "~3.3.8",

"@types/jasminewd2": "~2.0.3",

"codelyzer": "^5.1.2",

Online Collaboration System

"jasmine-core": "~3.4.0",

"jasmine-spec-reporter": "~4.2.1",

"karma": "~4.1.0",

"karma-chrome-launcher": "~2.2.0",

"karma-coverage-istanbul-reporter": "~2.0.1",

"karma-jasmine": "~2.0.1",

"karma-jasmine-html-reporter": "^1.4.0",

"protractor": "~5.4.0",

"ts-node": "~7.0.0",

"tslint": "~5.15.0",

"typescript": "~3.7.5"

}

}

**B. Backend –**

1. **projectModel.js –**

const mongoose = require('../../connection');

const Schema = mongoose.Schema; const projectSchema = new Schema({

name : String,

team : {type: mongoose.Types.ObjectId, ref : 'teams'},

members: [{type: mongoose.Types.ObjectId, ref : 'users'}],

startDate : Date,

endDate : Date,

comp : Number,

})

const model = mongoose.model("projects", projectSchema);

module.exports = model;

1. **TeamModel.js –**

const mongoose = require('../../connection');

const Schema = mongoose.Schema;

const teamSchema = new Schema({

name : String,

Online Collaboration System

admin\_id : {type: mongoose.Types.ObjectId, ref : 'users'},

member\_id: [{type: mongoose.Types.ObjectId, ref : 'users'}],

messages: {type: Array},

files: {type: Array},

teamtasks: {type: Array},

created : Date

})

const model = mongoose.model("teams", teamSchema);

module.exports = model;

1. **userModel.js –**

const mongoose = require('../../connection');

const Schema = mongoose.Schema;

const userschema = new Schema({

first: String,

last : String,

username : String,

email: String,

password: String,

avatar : String,

admin : Boolean,

created : Date,

teams : [{type: mongoose.Types.ObjectId, ref : 'teams'}],

})

const model = mongoose.model("users", userschema);

module.exports = model;

1. **projectManager.js –**

const express = require('express')

const router = express.Router();

const Model = require('../Models/projectModel'); router.post('/add', (req, res)=> {

data = req.body;

model = new Model(data)

Online Collaboration System

model.save()

.then(data => {

res.status(200).json(data);

})

.catch(err => {

res.status(500).json(err);

})

})

router.delete('/deletebyid/:teamid', (req, res) => { teamid = req.params.teamid Model.findByIdAndDelete(teamid)

.then(data => {

res.status(200).json(data);

})

.catch(err => {

res.status(500).json(err);

})

})

router.get('/getbyid/:id', (req, res) => {

id = req.params.id

Model.findById(id)

.populate('members')

.then(data => {

res.status(200).json(data);

})

.catch(err => {

res.status(500).json(err);

})

})

router.delete('/delete/:id', (req, res) => {

id = req.params.id

Model.findByIdAndDelete(id)

.then(data => {

res.status(200).json(data);

})

.catch(err => {

Online Collaboration System

res.status(500).json(err);

})

})

router.get('/getbyteam/:teamid', (req, res) => { id = req.params.teamid Model.find({team : id})

.populate('members')

.then(data => {

res.status(200).json(data);

})

.catch(err => {

res.status(500).json(err);

})

})

router.get('/getall', (req, res) => {

Model.find({})

.then(data => {

res.status(200).json(data);

})

.catch(err => {

res.status(500).json(err);

})

})

router.put('/update/:id', (req, res) => {

id = req.params.id;

data = req.body;

Model.findByIdAndUpdate(id, data)

.then(data => {

res.status(200).json(data);

})

.catch(err => {

res.status(500).json(err);

})

})

module.exports = router

Online Collaboration System

1. **TeamManager.js** –

const express = require('express')

const router = express.Router();

const Model = require('../Models/TeamModel'); const multer =require('multer');

const storage = multer.diskStorage({

destination : (req, file, cb) => {

cb(null, './uploads/files');

},

filename : (req, file, cb) => {

cb(null, file.originalname);

}

});

const upload=multer({storage : storage})

router.post('/addfile',upload.single('file'), (req,res)=>{ res.json({message : "file upload success"})

})

router.post('/add', (req, res)=> {

data = req.body;

model = new Model(data)

model.save()

.then(data => {

res.status(200).json(data);

})

.catch(err => {

res.status(500).json(err);

})

})

router.get('/getbyadminid/:adminid', (req, res) => { admin\_id = req.params.adminid Model.find({admin\_id : admin\_id})

Online Collaboration System

.populate('member\_id')

.then(data => {

res.status(200).json(data);

})

.catch(err => {

res.status(500).json(err);

})

})

router.get('/getbyname/:name', (req, res) => { let name = req.params.name Model.findOne({name : name})

.populate('member\_id')

.then(data => {

res.status(200).json(data);

})

.catch(err => {

res.status(500).json(err);

})

})

router.delete('/deletebyid/:teamid', (req, res) => { teamid = req.params.teamid Model.findByIdAndDelete(teamid)

.then(data => {

res.status(200).json(data);

})

.catch(err => {

res.status(500).json(err);

})

})

router.get('/getbyid/:id', (req, res) => {

team\_id = req.params.id

Model.findById(team\_id)

.populate('member\_id')

.populate('admin\_id')

.then(data => {

res.status(200).json(data);

})

.catch(err => {

Online Collaboration System

res.status(500).json(err);

})

})

router.get('/getall', (req, res) => {

Model.find({})

.populate('admin\_id')

.populate('member\_id')

.then(data => {

res.status(200).json(data);

})

.catch(err => {

res.status(500).json(err);

})

})

router.put('/updateteam/:id', (req,res) => {

team\_id = req.params.id;

teamdata = req.body;

Model.findOneAndUpdate(team\_id, teamdata)

.then(data => {

res.status(200).json(data);

})

.catch(err => {

res.status(500).json(err);

})

})

router.delete('/updateteam/:id'),(req,res)=>{

team\_id=req.params.id;

teamdata=req.body;

console.log(teamdata);

Model.findOneAndRemove(team\_id, teamdata)

.then(data => {

res.status(200).json({"message": "success"});

})

.catch(err => {

res.status(500).json(err);

Online Collaboration System

})

}

router.put('/addmsgs/:id', (req, res) => {

msgs = req.body;

teamid = req.params.id;

console.log(teamid);

Model.findOneAndUpdate({\_id : teamid}, {$push: {messages : msgs}})

.then(data => {

res.status(200).json(data);

})

.catch(err => {

res.status(500).json(err);

})

})

router.put('/updatemsgs/:id', (req, res) => {

msgs = req.body;

teamid = req.params.id;

console.log(teamid);

Model.findOneAndUpdate({\_id : teamid}, {messages : msgs})

.then(data => {

res.status(200).json(data);

})

.catch(err => {

res.status(500).json(err);

})

})

router.put('/addmember/:id', (req, res) => {

memberdata = req.body;

teamid = req.params.id;

console.log(teamid);

Model.findOneAndUpdate({\_id : teamid}, {$push: memberdata})

.then(data => {

res.status(200).json(data);

})

.catch(err => {

res.status(500).json(err);

})

Online Collaboration System

})

router.put('/addfile/:id', (req, res) => {

file = req.body;

teamid = req.params.id;

console.log(file);

Model.findOneAndUpdate({\_id : teamid}, {$push: file})

.then(data => {

res.status(200).json(data);

})

.catch(err => {

res.status(500).json(err);

})

})

router.put('/addtask/:id', (req, res) => {

file = req.body;

teamid = req.params.id;

console.log(file);

Model.findOneAndUpdate({\_id : teamid}, {$push: file})

.then(data => {

res.status(200).json(data);

})

.catch(err => {

res.status(500).json(err);

})

})

module.exports = router

1. **UserManager.js –**

const express = require('express')

const router = express.Router();

const Model = require('../Models/userModel'); const multer =require('multer');

const path = require('path');

const storage = multer.diskStorage({

destination : (req, file, cb) => {

Online Collaboration System

cb(null, './uploads/avatars');

},

filename : (req, file, cb) => {

cb(null, file.originalname);

}

});

const upload=multer({storage : storage})

router.post('/addimg',upload.single('avatar'), (req,res)=>{ res.json({message : "file upload success"})

})

router.get('/getImg/:name', (req,res) =>{

const imgpath = pathg.join(\_dirname, '../../uploads' , req.params.name);

console.log(imgpath);

res.sendFile(imgpath);

})

router.post('/add', (req, res)=> {

data = req.body;

model = new Model(data)

model.save()

.then(data => {

res.status(200).json({"message": "success"});

})

.catch(err => {

res.status(500).json(err);

})

})

router.put('/updateuser/:id', (req,res) => {

team\_id = req.params.id;

teamdata = req.body;

console.log(`user id : ${teamdata.\_id}`); Model.findByIdAndUpdate(team\_id, teamdata) .then(data => {

res.status(200).json(data);

})

.catch(err => {

res.status(500).json(err);

Online Collaboration System

})

})

router.get('/getbyUserName/:username', (req, res)=> { let usern = req.params.username

Model.findOne({username : usern})

.populate('teams')

.then(data => {

res.status(200).json(data);

})

.catch(err => {

res.status(500).json(err);

})

})

router.get('/getbyid/:id', (req, res)=> {

let id = req.params.id

Model.findById(id)

.populate('teams')

.then(data => {

console.log(`fetched ${data}`)

res.status(200).json(data);

})

.catch(err => {

res.status(500).json(err);

})

})

router.delete('/delete/:id', (req, res)=> {

let id = req.params.id

Model.findByIdAndDelete(id)

.then(data => {

console.log(`fetched ${data}`)

res.status(200).json(data);

})

.catch(err => {

Online Collaboration System

res.status(500).json(err);

})

})

router.get('/getall', (req, res) => {

Model.find({})

.then(data => {

res.status(200).json(data);

})

.catch(err => {

res.status(500).json(err);

})

})

module.exports = router

**7**. **socket-io-server.js –**

var server=require('http').server(app);

var io=require('socket.io')(server);

io.on('connection', function (socket){

console.log(socket);

console.log("A user connected");

socket.emit('test event','here is some data');

socket.on('çreateteam', (data) => {

console.log(socket);

socket.emit('teamcreated', socket);

})

});

server.listen(3000,()=>{

console.log('socket.io is listening on port 3000');

});

1. **index.js –**

Online Collaboration System

const express=require ('express');

const app=express();

const port=3000;

const parser = require('body-parser')

const userRouter = require('./api/routes/UserManager') const teamrouter = require('./api/routes/TeamManager') const projectrouter = require('./api/routes/projectManager') const cors=require('cors');

//socket config

var http = require('http').createServer(app);

var io = require('socket.io')(http);

io.on('connection', function(socket){

console.log('a user connected');

socket.on('jointeam', team => {

socket.join(team);

console.log(`connected to team : ${team}`);

})

socket.on('disconnect', function(){

console.log('user disconnected');

});

socket.on('teamMsg', data => {

console.log(`a team message ${data}`);

// data.sent = false;

socket.to(data.team).emit('recTeamMsg', data);

})

});

app.use(cors())

app.use(parser.json());

app.use('/user', userRouter);

app.use('/team', teamrouter);

app.use('/project', projectrouter);

app.use(express.static('uploads'))

app.get('/', (req, res) => {

res.send('get request accepted');

Online Collaboration System

})

app.get('/add', (req, res) => {

res.send('index add ')

})

app.post('/home', (req, res) => {

res.send('get request accepted');

})

http.listen(port, () => {

console.log('server is running');

})

1. **connection.js –**

const mongoose=require("mongoose")

const con\_url="mongodb://localhost:27017/teamColab" mongoose.connect(con\_url,{useNewUrlParser:true, useFindAndModify : false,

useUnifiedTopology: true})

.then(()=>

{console.log("successfully connected to the database");

})

.catch((err)=>

{console.log(err);

})

module.exports=mongoose