**Wingsl Tl Mini-project - Full-Stack Application**

Open **Project** folder(you can find in desktop) in VS Code application

**Backend Nodejs Part:**

Type below code in "/Desktop/Projects/wings.../Nodejs/routers/users.js" file: const express= require("express");



canst Course= require("../mongoose/models/courses");

//setting up the student router

const usersRouter = new express.Router();

//write your code here usersRouter.get('/courses/get', async (req, res)=> {

const data= await Course.find() res.status(200).json(data)

})

usersRouter.post('/courses/enroll/:id', async (req, res)=> { const id= req.params.id;

const getData = await Course.findByid(id) if (!getData.isApplied) {

po= {

isApplied: true

}

await Course.findByidAndUpdate(id, po) res.status(200).json({ message: "Sucessfully" })

} else {

res.status(403).json({ message: "not"})

}

})

usersRouter.patch('/courses/rating/:id', async (req, res)=> { const id= req.params.id;

const getData = await Course.findByid(id) if (!getData.isRated && getData.isApplied) {

const body= req.body;

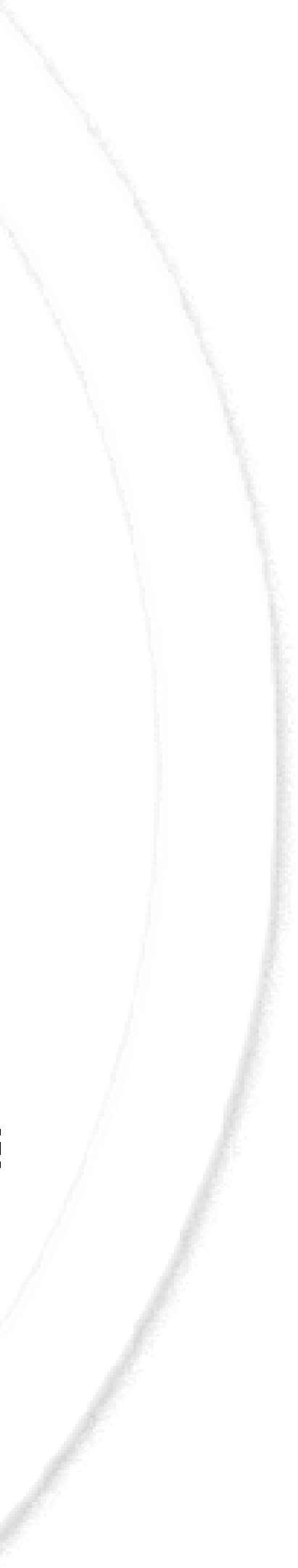
const noOfRatings = getData.noOfRatings + 1;

const finalRate = (((getData.noOfRatings \* getData.rating) + body.rating)

/ noOfRatings).toFixed(l)

pa= {

isRated: true, rating: finalRate,



noOfRatings: noOfRatings

}

await Course.findByidAndUpdate(id, pa) res.status(200).json({ message: "Sucessfully" })

} else {

res.status(403).json({ message: "Not"})

}

})

usersRouter.delete('/courses/drop/:id', async (req, res)=> { const id= req.params.id;

const getData = await Course.findByid(id) if (getData.isApplied) {

de= {

isApplied: false

}

await Course.findByidAndUpdate(id, de) res.status(200).json({ message: "Sucessfully" })

} else {

res.status(403).json({ message: "Not"})

}

})

module.exports= usersRouter;

**Frontend ReactJs Part:**

Type below code in "/Desktop/Projects/wings.../ReactJs/src/App.js" file:

import React, {Component} from 'react'; import "./App.css"

class Home extends Component {

url = "http://localhost:8001/courses/" state= {

show: false, data: [], rating: 1,

}

componentDidMount = () => { this.handleGetData()

}

handleGetData = () => {

fetch(this.url + "get")

.then((res) => res.json())

.then((json) => {

this.setState({ data: json })

})

}

handleApply = async (id)=> { const requestOption = {



method: 'post',

headers: { 'Content-Type': 'application/json' }

};

fetch(this.url + 'enroll/' + id, requestOption)

.then(() => {

this.handleGetData()

})

};

handleRating = (e) => {

this.setState({ rating: e.target.value })

}

handleAddRating = async (id)=> { const requestOption = {

method: 'PATCH',

headers: { 'Content-Type': 'application/json' }, body: JSON.stringify({ rating: this.state.rating})

}

fetch(this.url +"rating/"+ id, requestOption)

.then(() => {

this.handleGetData()

})

}

handleDrop = async (id)=> {

const requestOption

= {

method: 'DELETE',

headers: { 'Content-Type': 'application/json' },

};

fetch(this.url +"drop/"+ id, requestOption)

.then(() => {

this.handleGetData()

})

}

render() {

return (

<div className="home">

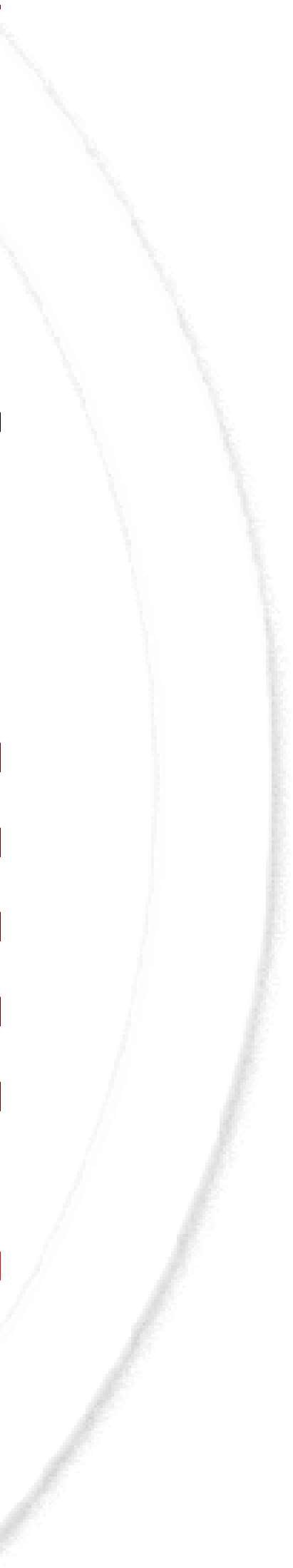
<header>

<h2>ABC Learning</h2>

</header>

<div className="cardContainer">

{this.state.data.map(courses => { return (



<div className="card">

**<ul>**

<div className="header">

<li>{courses.courseName}</li>

<li>{courses.courseDept}</li>

<li>{courses.description}</li>

{courses.isApplied &&

<li>

{!courses.isRated 8&

<li>Rate:

<select className="rating" name="rating" onChange={this.handleRating}>

<option>l</option>

<option>2</option>

<option>3</option>

<option>4</option>

<option>S</option>

</select>

<button className="rate" onClick={() => this.handleAddRating(courses.\_id)}>Add</button>

</li>

}

{courses.isApplied &&

<button className="drop" onClick={() => this.handleDrop(courses.\_id)}>Drop Course</button>

}

</li>

}

{!courses.isApplied &&

<li><button className="btn" onClick={() => this.handleApply(courses.\_id)}>Apply</button></li>

}

</div>

<div className="footer">

Join our channel if you haven't joined yet https://t.me/fresco milestone( @fresco\_milestone )

<li>{courses.duration} hrs .

{courses.noOfRatings} Ratings . {courses.rating}/5</li>

</div>

</ul>

</div>

)

})}

</div>

</div>

) ;

}

}

export default Home

Install project dependencies by using "sh install.sh" command in the terminal and to run the application by using "sh run.sh"

To test the application "sh test.sh"