Assignment – 2 (12%)

COMP 3123 Full Stack Development – I

Submission: Week 13 – Sunday, 04th Dec 2022 23:59 PM

Not submission extension as it might affect other coursework

PART – I/Assignment – I:

Creating NodeJS/Express/MongoDB Application

Refer Assignment-1 and call all api's using axios http client or http client of your choice to make data available to frontend using react. Don't make mongodb connection on frontend side.

Docker container

Create docker compose to create mongodb, frontend and backend container. Submit docker compose file to GitHub repo and provide the instruction to execute the containers.

OR

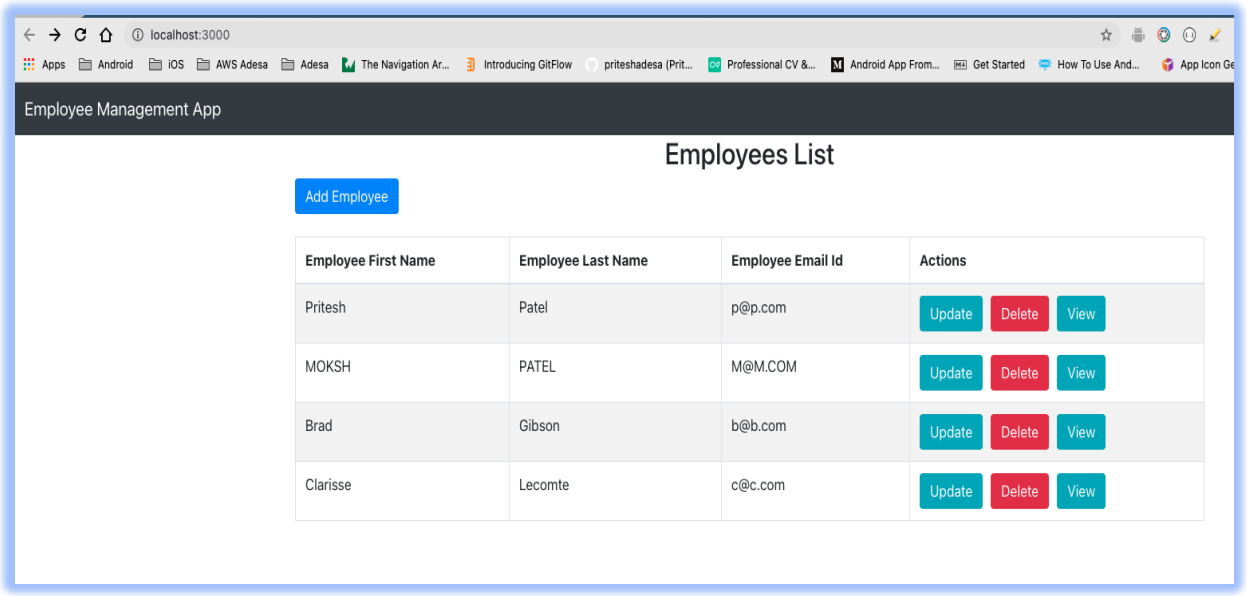
Deploy your backend to Heroku/Vercel or hosting/cloud of your choice. Update backend whenever required to accommodate the requirement of frontend

PART – II: - Creating ReactJS Frontend Application

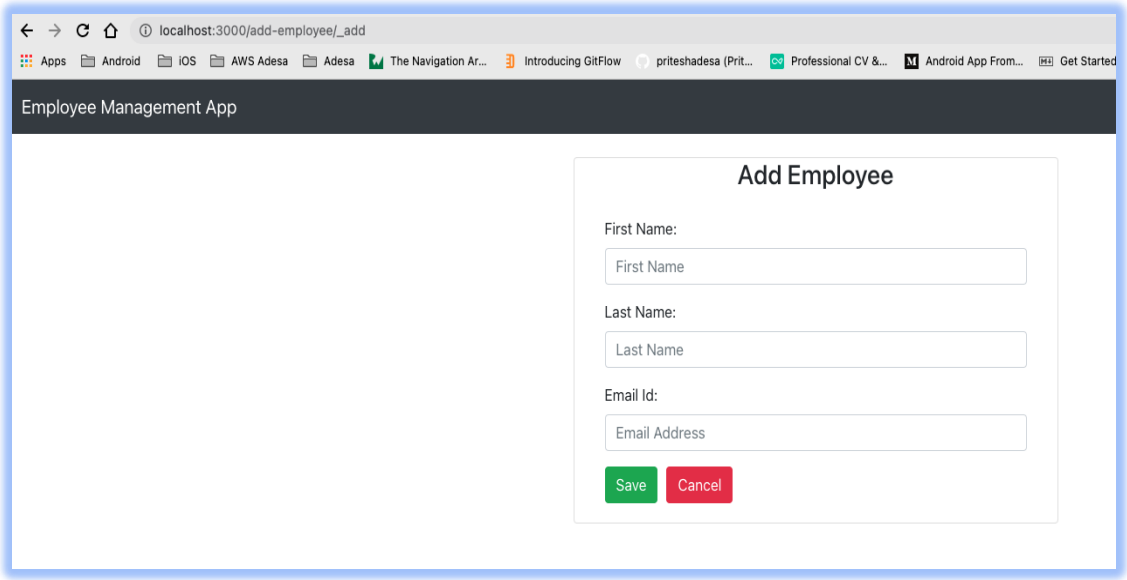
As a newly hired graduate I would like to demonstrate my full stake developer skills. I am creating following application to validate my potential to secure the job of frontend developer.

Application requirements are as listed below:

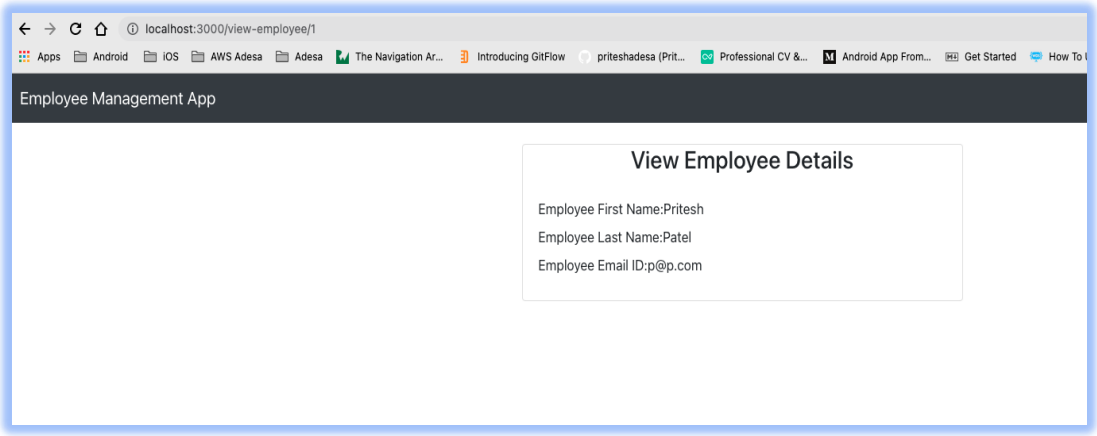
1. Create ReactJS application name **studentID_comp3123_assignment2_reactjs** to perform CRUD operation. Consume REST API exposed by your backend application. Also create **GitHub** repo with same name and frequent commit/push the code with appropriate commit message.
2. Implement appropriate Routes/Navigation/Menu techniques to load pages
3. Create Login and Signup screens (Make design of your choice).
4. Maintain session using html local storage or technology of your choice.
5. After successful login open List of employee screen**. (Sample Design below)**



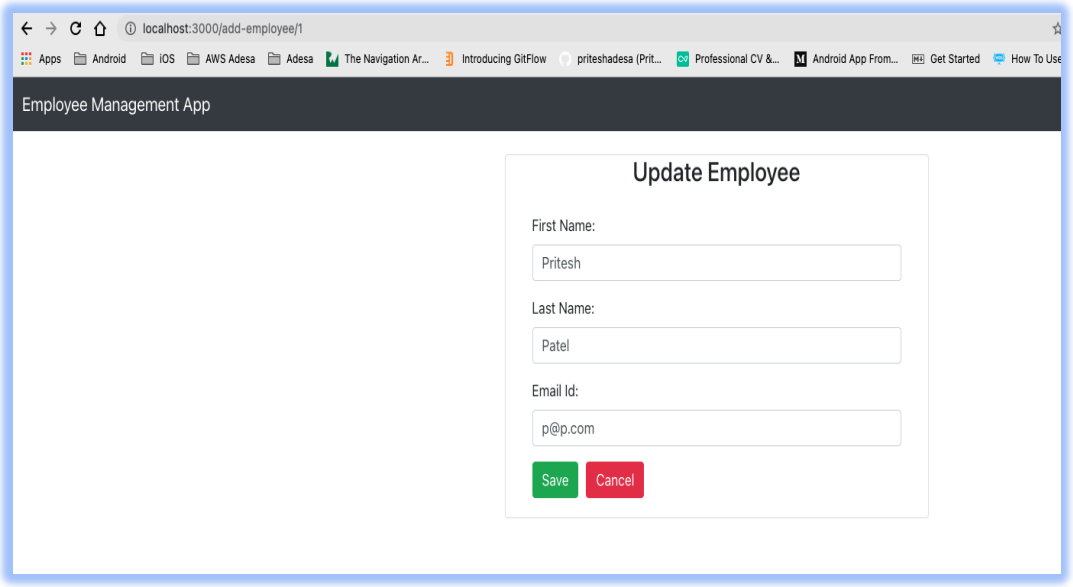
6. On click of "Add Employee" add new employee by calling API **(Sample Design below)**



7. View Employee details when you click on View button **(Sample Design below)**



8. Update employee information when you click Update button **(Sample Design below)**



9. Delete employee record when you press delete button

10. Logout from application and redirect to login page

NOTE:

- **NO extension as it might affect other coursework**
- *All screens design is for reference only. Think of your own UI design.*
- *Please do some research to look your work more professional and acceptable according to industry standards.*
- *Make use of ReactJS concepts like states, props, material design, react-bootstrap, fetch/Axios, react-routes, class component, function components, hooks, etc.*

Evaluation Rubric

- **Part-I:** 10% - Deploy your all APIs to **Heroku/Vercel** app otherwise **BIG ZERO**
- **Part-II:** 80% - Each Correct screen with professional design and for any error or half/wrong implementation **BIG ZERO**
- Correct **GitHub** repository submission and **naming** convention 10%. No GitHub repo and no correct implementation of git then **BIG ZERO**.
- Display error messages for each validation

Sr. No.	Component	Points
1	Deploy backend to Heroku/Vercel or OTHER or creating docker compose	10
2	Working Signup component, Login component and Logout	15
3	List all Employee component with good design and theme	15
4	Add New Employee screen with good design and theme and with all validation messages	15
5	View and Update Employee component with good design and theme and with all validation messages	15
6	Delete Employee	10
7	Accepted UI/UX using material design OR bootstrap, etc.	10
8	GitHub repository with valid commits and readme file. Screenshots submission for validation on BB	10

Submission

- Remove **node_module** folder from both projects
- Upload ZIP file of your Backend NodeJS application
- Upload ZIP file of your Frontend ReactJS application
- Screenshot showing data from your mongodb database (1 screen)
- Screenshot showing data from your REST API test on POSTMAN (5-8 screen)
- Screenshot of all your CRUD operation screens on ReactJS app (5-8 screen)
- GitHub repos links of backend and frontend application

If you have any question, then email at pritesh.patel2@georgebrown.ca or use SLACK