Assignment – 2 (12%)

COMP 3123 Full Stack Development – I

Submission: Week 13 – Sunday, 05th Dec 2021 23:59 PM

Not submission extension as it might affect other coursework

PART – I: Creating NodeJS/Express/MongoDB Application

1. Create NodeJS + express + mongodb/mongoos application having name **studentID_assignment2_backend**. Also create same **GitHub** repo.
2. Create database name **studentid_assignment2** on mongodb atlas containing **employee** collection.
3. Database collection fields to consider are as follow:

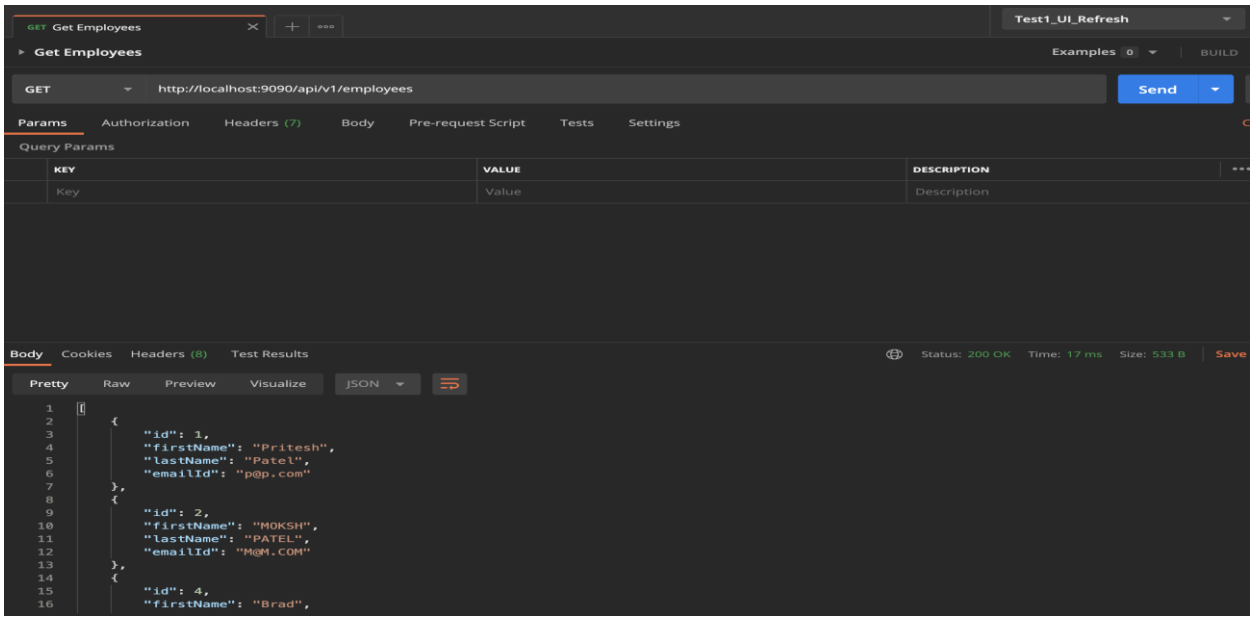
   Id, firstname, lastname and emailid
4. Implement **mongoos** for data modeling.
5. Validation to perform are as follow: (**Refer** previous classes work and exercises)
   a. Id, firstname and lastname are mandatory fields, if values are not provided for any of the field, then return appropriate error message
   b. Emailid is mandatory and must be in valid format, otherwise return appropriate error message
6. Develop following five REST APIs (**JSON**) using express to perform CRUD operations.

Sr. No.	API Name	HTTP Method	Path	Status Code	Description
(1)	GET Employees	GET	`/api/v1/employees`	200 (OK)	All Employee resources are fetched.
(2)	POST Employee	POST	`/api/v1/employees`	201 (Created)	A new Employee resource is created.
(3)	GET Employee	GET	`/api/v1/employees/{id}`	200 (OK)	One Employee resource is fetched.
(4)	PUT Employee	PUT	`/api/v1/employees/{id}`	200 (OK)	Employee resource is updated.
(5)	DELETE Employee	DELETE	`/api/v1/employees/{id}`	204 (No Content)	Employee resource is deleted.

7. **Optionally** implement Login feature if you wish (JWT token) and host backend app to Heroku platform for free.

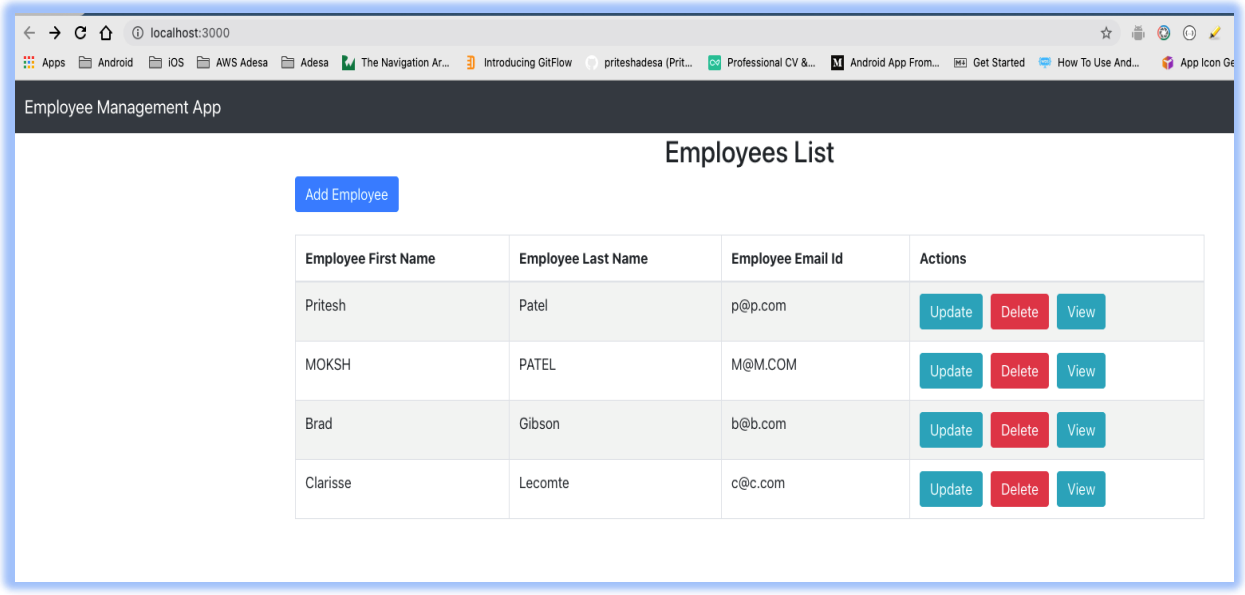
8. **Testing**: Import given Postman API file from assignment folder (**Assignment2_Employee_API.postman_collection.json**) to refer/test the REST API. Here I am using http://localhost:9090 base URL to call all the endpoints after executing backend application. Please update POST according to your configuration.

For example

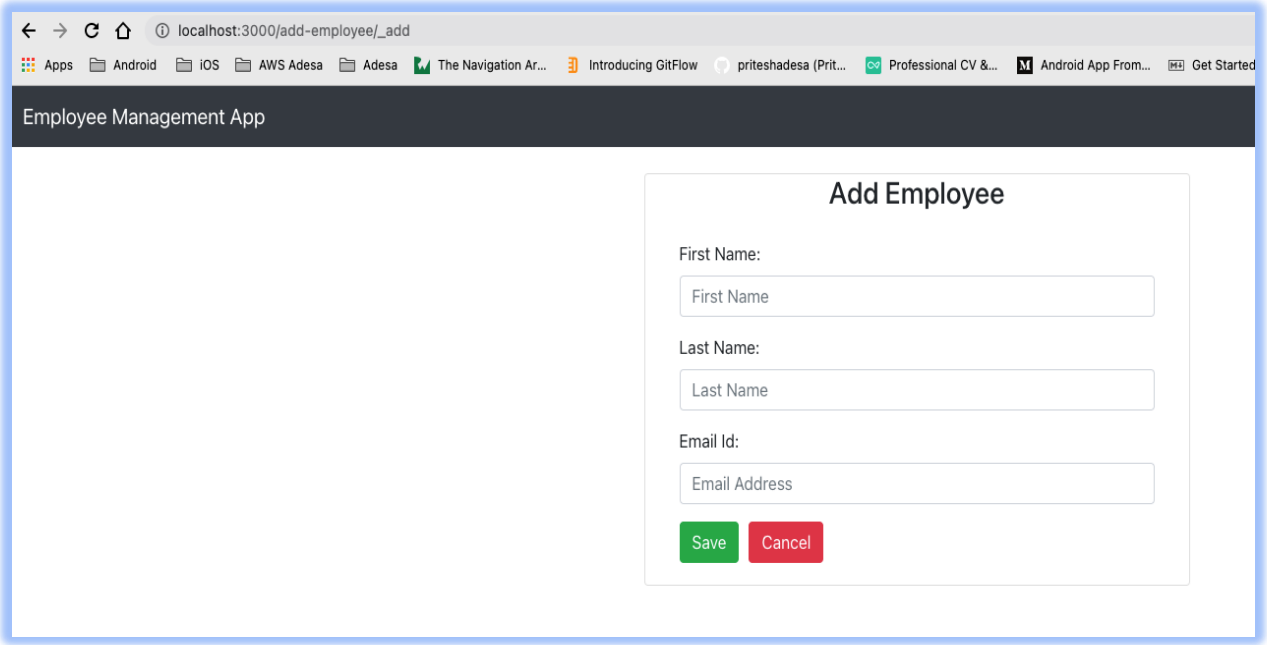


PART – II: - Creating React JS Application

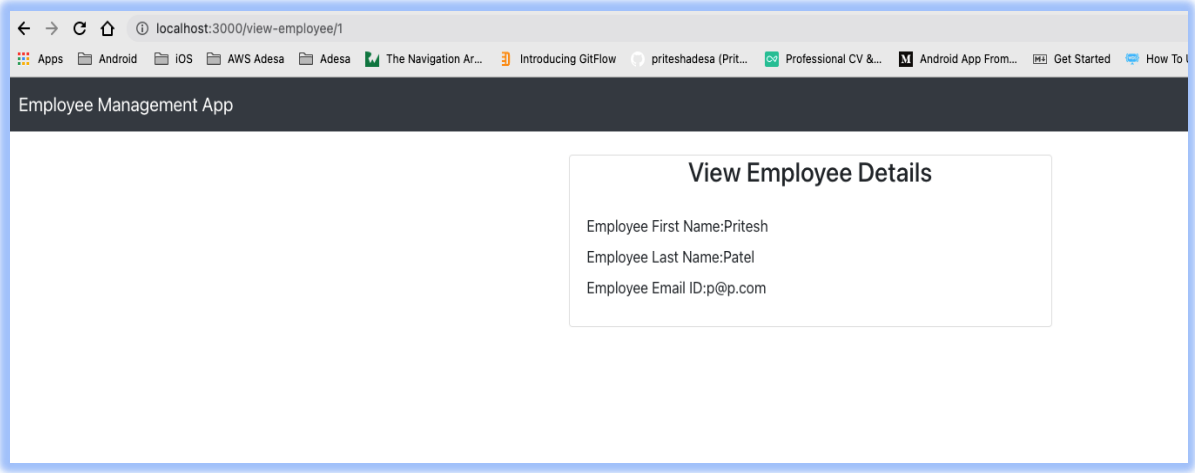
1. Create ReactJS application name **studentID_assignment2_frontend** to perform CRUD operation. Consume REST API exposed by your backend application. Also create **GitHub** repo with same name
2. Implement appropriate navigation/Menu techniques to load pages
3. List employees screen



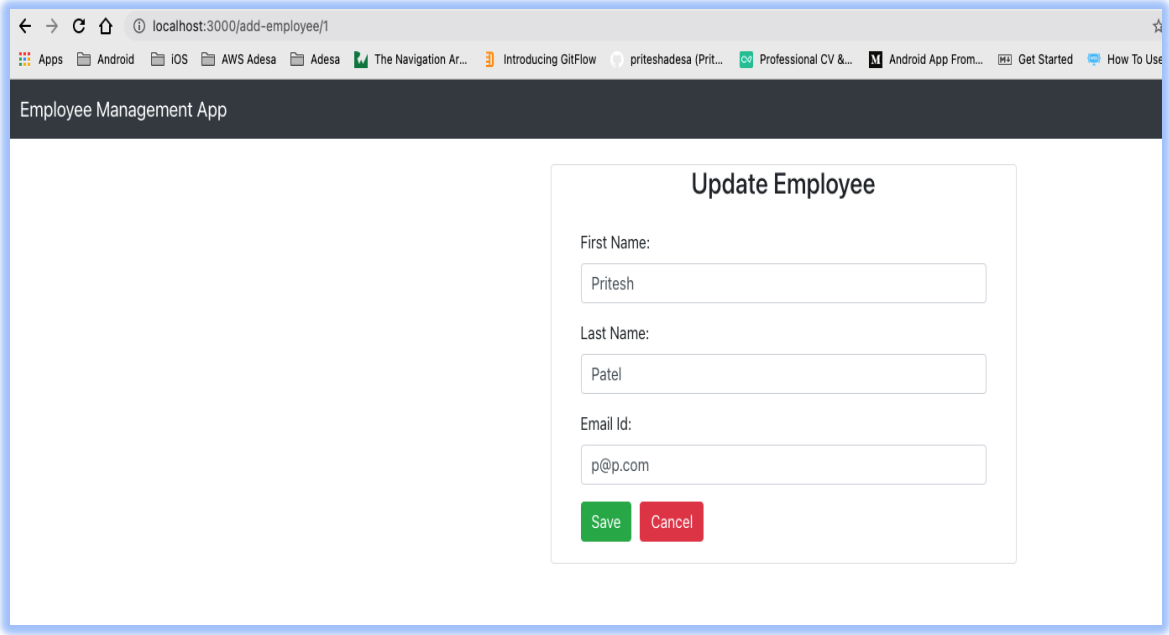
4. Add new employee



5. View Employee details when you click on View button



6. Update employee information when you click Update button



7. Delete employee record when you press delete button

NOTE:

- **Not 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

- **Part-I:** 45% - Each Correct API 9%, for any error or half/wrong implementation **BIG ZERO**
- **Part-II:** 45% - Each Correct screen with professional design 9%, 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**.

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 of backend and frontend application

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