Assignment – I (10%)

COMP 3123 – Full Stack Development – I

Submission: Sunday, 23rd Oct 2022, 11:59 PM

As a newly hired Jr. Software Engineer my manager assigned me a task to develop Backend application using NodeJS, Express and MongoDB. He also wants me to apply my VCS (**GitHub**) skills to develop these projects where I will commit and push all my code to **Student#_COMP3123_Assignment1** repository.

Following is the list of APIs to develop which accept all data as JSON Object whenever needed:

Sr. #	Method	Endpoint	Response Code	Description
1	POST	/api/user/signup	201	Allow user to create new account
2	POST	/api/user/login	200	Allow user to access the system
3	GET	/api/emp/employees	200	User can get all employee list
4	POST	/api/emp/employees	201	User can create new employee
5	GET	/api/emp/employees/{eid}	200	User can get employee details by employee id
6	PUT	/api/emp/employees/{eid}	200	User can update employee details
7	DELETE	/api/emp/employees?eid=xxx	204	User can delete employee by employee id

Sample Error Response:

```
{
        “status”: false,
        “message”: “Invalid Username and password”
}
```

Sample Request Payload for login:

```
{
        “username”: “p@p.com”,
        “password”: “password$123”
}
```

Sample Request Payload for login:

```
{
        “status”: true
        “username”: p@p.com,
        “message”: “User logged in successfully”
        “jwt_token”: “Optional implementation”
}
```

Sample Request Payload for employee:

```
{
        “first_name”: “Tam”,
        “last_name”: ”Harrow”,
        “email: : tam@hollywood.com,
        “gender”: “Male”,
        “salary”: 125500.00
}
```

MongoDB Database name: **comp3123_assigment1**

Users Collection

Field Name	Type	Constraint
_id	Object ID	Auto Generate
username	String (100)	Primary Key
email	String (50)	Unique
password	String (50)	May be encrypted with other fields

User can login using username/email and password

Employee Collection

Field Name	Type	Constraint
_id	Object ID	Auto Generate
first_name	String (100)	Required
last_name	String (50)	Required
email	String(50)	Unique
gender	String (25)	Male/Female/Other
salary	Float	Required

Notes:

- Make use of express.Routes() and modules
- Validate the data whenever required
- Return error details or success response details whenever required
- All data must be sent back and forth in JSON Object format
- Optionally apply JWT security concept to secure all your API calls
- No late submission

Submission:

1) MongoDB console screenshots
2) Postman API collection
3) Screenshots of each API tested with response and payload
4) Project ZIP file after **deleting node_modules folder**
5) GitHub Project Link
6) Sample user detail to login
7) Any comment
8) If hosted on Heroku or any platform share the details with urls

Evaluations:

Points	Evaluation Component
35	5 points for each 7 working API without database connection
70	10 points for each 7 working API with database connection
10	10 Point for sending validation and error messages to client
10	10 point maintaining GitHub repository with read me file
10	Uploading valid screenshots with appropriate name and documentation

Communication:

- Please contact on pritesh.patel2@georgebrown.ca or SLACK channel for any question or query.
- NO communication before 3 days for submission deadline

***~~~ Wish you all the Best ~~~***