

COMP229 – Web Application Development

Assignment 2

Portfolio Application – Node.js, Express REST APIs & MongoDB

Due Week #6 – Saturday 11:59 PM

Portfolio – Node.js, Express REST APIs & MongoDB

Maximum Mark: 100

Overview: Create the Node.js Express exports REST APIs that interacts with MongoDB Database using Mongoose ODM for your Portfolio application. (Note: The Front-end of the application is already created in Assignment1).

PART I – Reference (Week4 slides)

1. Using MongoDB database, create **(20 Marks)**

- A database by name Portfolio.
- Create the following models with their respective property. (5 Marks: Functionality).

I. contacts

```
firstname: string  
lastname: string  
email: string
```

II. projects

```
title: string  
completion: Date  
description: string
```

III. services

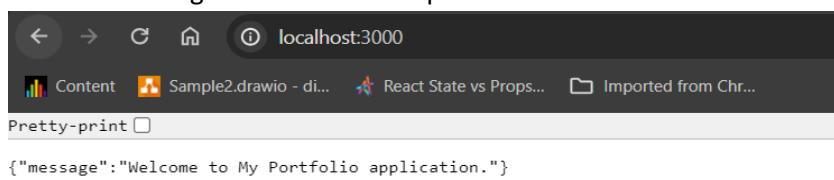
```
title: string  
description: string
```

IV. users

```
firstname: string  
lastname: string  
email: string  
password: string  
created: Date  
updated: Date
```

2. Configure the Backend of your portfolio Application by creating the server.js file in the root folder for your server **(10 Marks):**

- Create a router with a route to the root endpoint of the server. It should return a JSON object with the message shown in the snapshot below:



The screenshot shows a browser window with the address bar set to 'localhost:3000'. Below the address bar, there are several tabs: 'Content', 'Sample2.drawio - di...', 'React State vs Props...', and 'Imported from Chr...'. At the bottom of the browser window, there is a text area containing the following JSON code:

```
{"message": "Welcome to My Portfolio application."}
```

3. Configure the server (60 Marks):

- a. Add the configuration for the MongoDB database, ensure your application is connected to the database with a log in the server console.
- b. Add to the configuration of Express with the middlewares http-errors, morgan, and cors.
- c. Create an error handler and register as the last middleware of Express.

PART II – Reference (Week5 and 5 slides)

- d. Write the controllers (separate files) for contacts, project, services, and users.
- e. Create routers (separate files) and define the routes for handling all CRUD operations for the contact, services, project and users api.

Below is an overview of the REST APIs that will be exported:

contacts

Methods	Urls	Actions
GET	api/contacts	get all contacts
GET	api/contacts/:id	get contacts by id
POST	api/contacts	add new contact
PUT	api/contacts/:id	update contact by id
DELETE	api/contacts/:id	remove contact by id
DELETE	api/contacts	remove all contacts

projects

Methods	Urls	Actions
GET	api/projects	get all projects
GET	api/projects/:id	get projects by id
POST	api/projects	add new project
PUT	api/projects/:id	update project by id
DELETE	api/projects/:id	remove project by id
DELETE	api/projects	remove all projects

services

Methods	Urls	Actions
GET	api/services	get all services
GET	api/services/:id	get services by id
POST	api/services	add new service
PUT	api/services/:id	update service by id
DELETE	api/services/:id	remove service by id
DELETE	api/services	remove all services

users

Methods	Urls	Actions
GET	api/users	get all users
GET	api/users/:id	get users by id
POST	api/users	add new user
PUT	api/users/:id	update user by id
DELETE	api/users/:id	remove user by id
DELETE	api/users	remove all users

- 4. a) Test the REST APIs above using Postman, Thunder client or any tool you are familiar with. e.t.c.
b) Provide the screen snapshot of the test. (10 Marks)**

SUBMITTING YOUR WORK

Your submission should include:

1. A zip archive of your Portfolio Project files.
2. A link to GitHub repository of your backend
3. A link to live backend server running on a cloud provider (Ex.: Render.com)
4. A PDF file showing the snapshots

This assignment is weighted **10%** of your total mark for this course.

Late submissions: **25%** deducted.