Lab 3: Web Development using Node& MongoDB

Objective:

- To develop web pages using Node & Mongo DB

Submission:

- Checkpoints that need to be shown to the course teacher during the lab.

Introduction:

This lab has two goals:

1. To introduce to handle form submission using Node and PUG (Jade)
2. To introduce how to work with Node and MongoDB

You will extend the last task from the last lab.

In this lab, you will learn how to develop web pages using Node, NPM and Jade.

There are two tasks that you will need to complete in this lab. Each task has a checkpoint. ***Complete all checkpoints and show it to your teacher.***

Task-1: Handling Form (5 marks)

In this task, you will need to handle HTML forms using Express and PUG (Jade). For this, follow the tutorial from this URL: https://www.tutorialspoint.com/expressjs/expressjs_form_data.htm

To setup the environments, you should re-use the contents from the task 4. Your main objective is to handle the form data and then display them in another Jade page.

While following the tutorial, if you receive an error similar to “Cannot find module pug”, then install the pug module using npm.

Also note that while following the tutorial you might find get an undefined error for your “req.body” variable in the console. This is because the behaviour of body-parser package has been changed since the tutorial was published. To rectify this error follow the steps:

- Remove this: **app.use(bodyParser.urlencoded({ extended: true }));**
- Add **var urlencodedParser = bodyParser.urlencoded({ extended: true })**
- Replace this line**app.post('/', function(req, res)** with **app.post('/', urlencodedParser, function(req, res)**

As the checkpoint for the task, **change the labels to “Movie Name” and “Movie Year”**. Once a user submits the form, show the submitted data to a webpage.

Task-2: Connection with MongoDB (10 marks)

In this task, you will configure your web application for interacting with database. For this we will use MongoDB. For those who are not familiar with MongoDB, it is NoSQL database. That is MongoDB does not use traditional relational tables as used in any SQL database. MongoDB uses JSON-like documents with optional schemas, meaning you do not create pre-defined schemas beforehand to use this DB. In addition, you can store and retrieve JSON data within MongoDB, a convenient feature for web applications.

At first you should install MongoDB in your workstation by following the four steps provided from this link: https://docs.mongodb.com/manual/tutorial/install-mongodb-on-ubuntu/#install-mongodb-community-edition

Once installed, you can use the following commands to manage your MongoDB server:

- To start: sudosystemctl start mongod
- To check status: sudosystemctl status mongod
- To ensure MongoDB start after a system restart: sudosystemctl enable mongod
- To stop: : sudosystemctlstopmongod
- To restart: sudosystemctl restart mongod
- To start a mongo client session with the MongoDB: mongo/mongosh

Similar to the previous task, you try to understandthe tutorial from this link, up to the retrieving the document section:

- https://www.tutorialspoint.com/expressjs/expressjs_database.htm

Now, create a web application for a movie database. For this, edityour contents from Task 1 and add the following functionalities:

- /addMovie route where a pug template will show four fields: Movie Name, Movie Director, Movie Release Year and Genre.
- /viewMovie route to retrieve any single movie utilising the movie name and show its information on the web page using a PUG template.