CSE1500 – Web Technology

Assignment Three: **Express + MySQL**

Dear Students,
In our recent lectures, we have covered Express and MongoDB. This assignment will focus on practicing MySQL because you are already familiar with it. We will provide you with all the necessary code and materials to start. This assignment aims to guide you through a sample project, which can be found in the GitHub repository at **https://github.com/RaddyTheBrand/Nodejs-UserManagement-Express-Hbs-MySQL**. Your task is running the project and making a few minor modifications. A complete video tutorial is available at **https://youtu.be/1aXZQcG2Y6I** (2 hours video!)

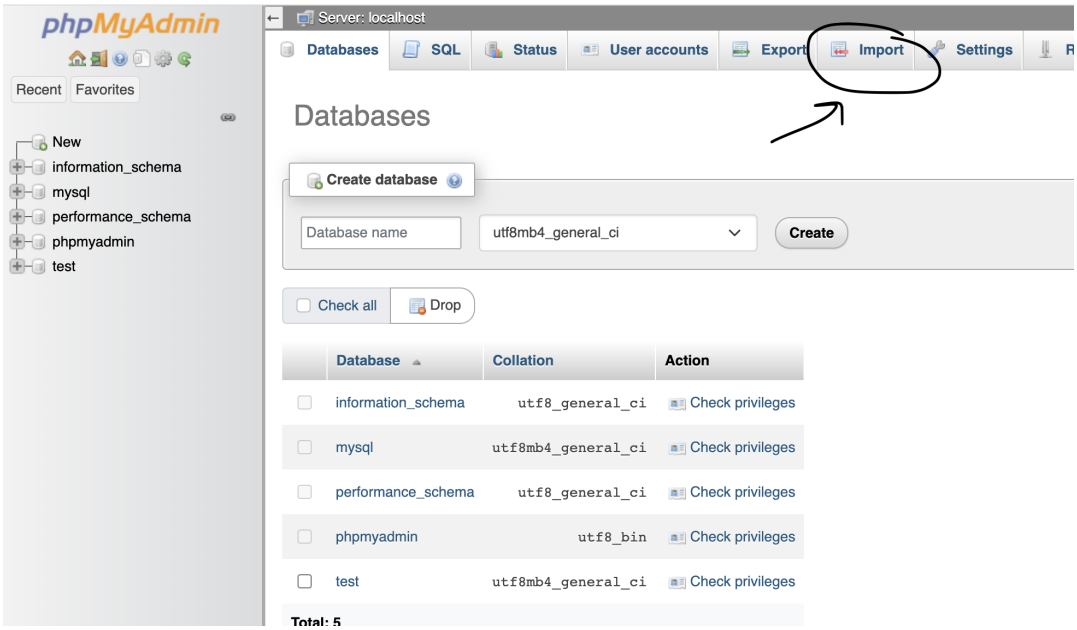
Note: We have minimized the workload for the third assignment as much as possible. This will allow you to focus on other assignments/subjects without spending extensive time. By working on this project, you will also gain experience using the MySQL database.

STEP ONE: INSTALLTION OF XAMPP

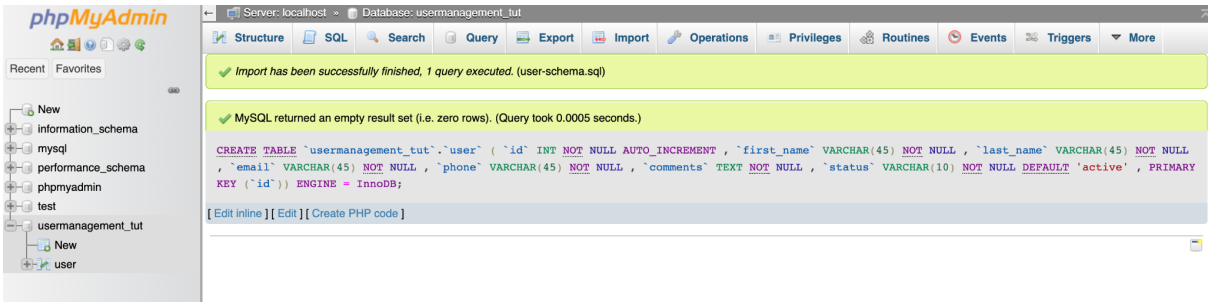
1. Download the project folder and rename it however you want.
2. To ensure that we are on the same page, please install the XAMPP server. XAMPP is a software packet that contains four key components, i.e., X stands for cross-platform compatibility, A (Apache web server), M (MySQL database).

STEP TWO: IMPORTING A DATABASE

1. To set up the sample database, we will follow a video. Please watch this video from 31:10 Database Setup (https://youtu.be/1aXZQcG2Y6I?t=1871).
2. Create a database name 'usermanagement_tut'. Please name it precisely because we can import the schema in the next step. If you wish to create the database from scratch, please do so.
3. Import a user-schema file that I provided in the project files (you can skip this step and instead follow the instructor video if you want to create a database from scratch)



3. After importing the DB file, you will see the confirmation message:



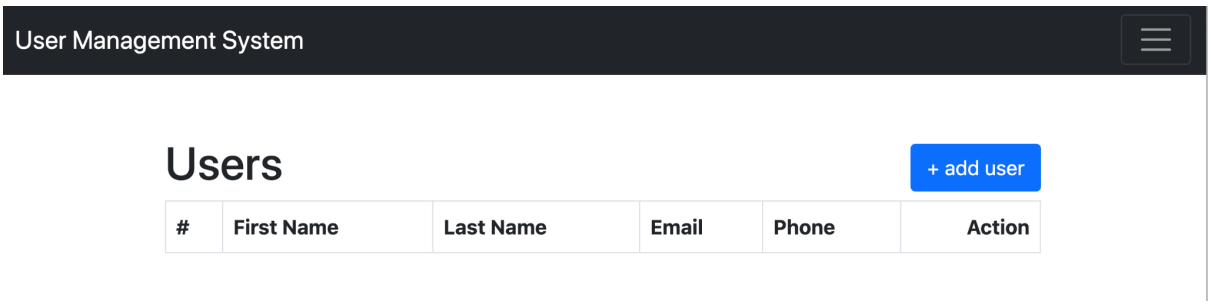
4. Create a .env file where we store our variables, as the video explains.

You can add the following variables in this file:

```
DB_HOST = localhost
DB_NAME = usermanagement_tut
DB_USER = root
DB_PASSWORD = password
```

The above file was imported in the app.js file as **require('dotenv').config();**

5. Next, you need to install the SQL driver for Nodejs using the following command: npm install mysql
6. After configuring all this, head over to the terminal and enter **npm start**, and then visit this page: http://127.0.0.1:5000/, and you will see the following page (do not forget to install **nodemon** if you have not installed it already):



STEP THREE: CODE INSPECTION

1. Carefully navigate to the desired folders, observe the folder's contents, and try to understand how it relates to the previous lectures.
2. Try adding a few new users.
3. Click on the edit button and observe how the app generates dynamic routes.
4. Use the search function to look for specific users.

STEP FOUR: CODE REFACTORING

After careful inspection and playing around with the application, it's time to make some minor edits.

TASK # 1: Add a model class

Currently, the app is not structured per the **model view controller** design pattern we learned in the last lecture. Your job is to add a **models** folder at the project's root and create a model class (**User**) that can perform all the database-related operations. Move all the code from the **controllers** that involve database operations to the model class. You can look at the code examples from **lecture seven** if you need help to achieve this.

Also, move the controller's folder to the root level and remove the server folder. **Please carefully update all the references to other files.**

TASK # 2: Replace handlebars with EJS

The author of this code has used handlebars, also a templating engine. Your second task is to replace this templating engine with the EJS engine we studied last time.

STEP FIVE: ADD A NEW FEATURE

1. In the main dashboard, add a new button labeled "Activate".
2. When the user clicks on this button, a request should be sent to the server to update the selected user's status from "None" to "Active". (check the database column name 'status')

3. At the same time, the row of the selected user in the dashboard should change color to light green, and the label should change from activate to deactivate. If the user clicks on the "deactivate" button again, the status of the user should be updated to "None," and the row color should change to its original color.

BONUS: Integrating MongoDB into this project can enhance exam preparation, though it is not mandatory for the assignment.

Assignment submission

1. Please submit your assignment in a **single ZIP** with your name/registration number as the project name.
2. Also, convert your code and deliverables into a **single PDF file** so the teaching team can assess them using a software tool.

Good Luck!