Database-driven Web Applications with Flask

COMP1048: Databases and Interfaces (2024-2025)

Matthew Pike & Yuan Yao

Table of contents

1 Lab Overview

In this lab, you will gain practical experience connecting web applications to databases using Flask and Python. It serves as a useful introduction to Coursework 2, covering key concepts and skills needed for the assignment. This hands-on approach will help you feel prepared and confident in addressing more advanced challenges.

2 Getting Started

To begin this lab on creating database-driven web applications with Flask, follow these steps:

Step 1: Download Lab Materials

First, obtain the lab materials by downloading them from Moodle. Unzip the starter code into a folder of your choice, which will be your working directory for this lab. Note, the starter code is the same as the example code from the lecture. You should rename the folder according to your student number, e.g. `12345678`.

Step 2: Open Your Code Editor

Visual Studio Code is recommended as it supports Python and web technologies (HTML, CSS). Open your working directory in Visual Studio Code.

Step 3: Open a CLI (Terminal/CMD)

Open a command line interface (CLI) window within your working directory. If you need a refresher on how to do this, look back at Lab 001 for guidance.

Step 4: Run the Flask Application

In the CLI, run `python app.py` (or `python3 app.py` if you're on a system where Python 3 is not the default). Navigate to `http://127.0.0.1:5000/` in your web browser; you should see the following:

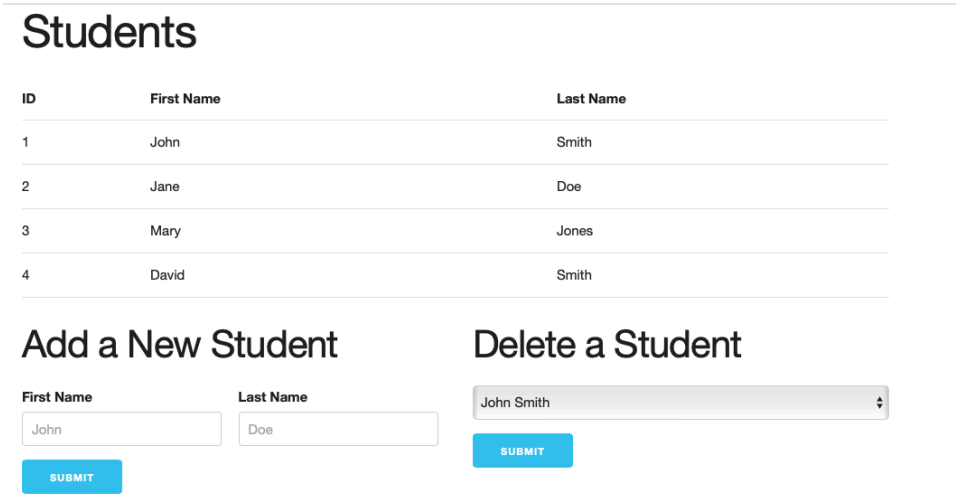

Figure 1: Screenshot of the index page

Step 5: Familiarize Yourself with the Code

Take a moment to familiarize yourself with the code in the `app.py`, `index.html`, and `base.html` files. You may ignore the CSS files for now, as they are not relevant to this lab.

Step 6: Begin Lab Tasks

With these steps completed, you're now fully set up to start working on the lab tasks.

3 Tasks

3.1 Adding Student Average Grades

Extend the `app.py` and `index.html` files to display the average grade of each student, in the table on the index page (`templates/index.html`). The average grade should be displayed in the table on the index page, and should be rounded to two decimal places. All students should be displayed, even if they have no grades.

3.2 Create a Modules Page

Extend the `app.py` adding a new route (`/modules/`) which renders a new `modules.html` template. The `modules.html` should extend `base.html` and include a table displaying the following information for each modules:

- Module code
- Module title
- Module credits
- Number of students enrolled on the module
- Average grade of students enrolled on the module

The table should be sorted by module code, in ascending order. The average grade should be rounded to two decimal places. All modules should be displayed, even if they have no students enrolled.

3.3 Bonus Task: Create and Delete Modules

This task is optional and is not required for the lab. It is provided as an additional challenge for those who wish to extend their learning.

You'll see in the `templates/index.html` interface, there are controls for adding and deleting students. Extend your solution to provide similar functionality for modules.

4 Submitting Your Lab Work

There is no submission required for this lab. However, you are encouraged to keep a copy of your work for future reference. The model solution for this lab will be released on Moodle after the lab session has finished. The model solution is intended to provide a reference for your coursework 2.