

# Webpage Explanation: Student Information

## Introduction

This document explains the functionality of the "Student Information" webpage that displays student details, courses taken, and related information.

## Webpage Overview

The webpage is designed to allow users to enter a student code and view information about the student, including their name, age, gender, city, courses taken, and total cost of the courses.

## Features

### Form Submission

- The user is presented with a form where they can enter a student code.
- Upon clicking the "Submit" button, the entered student code is sent to the server for processing.

### Student Details

- If the entered student code is valid, the webpage displays various details about the student, including:
  - Name
  - Student Code
  - Age
  - Gender
  - City

### Courses Taken

- The webpage lists the courses that the student has taken along with their corresponding prices.
- The total number of courses taken and the total cost of the courses are displayed.

## Usage

- To view a student's course details, enter one of the following student codes in the provided form:
  - S101
  - S102
  - S103
  - S104
  - S105

- S106
  - S107
  - S108
  - S109
  - S110
- After entering the student code, click the "Submit" button to retrieve and display the relevant information.

## Design and Layout

- The webpage has a clean and user-friendly design.
- The background color is light gray, providing a subtle and pleasant visual experience.
- The content is displayed within a centered container with a slightly darker background.

## Code Explanation

The following sections provide explanations for the key components of the code:

### Flask App Initialization

- The Flask app is created using the `Flask` class.
- The student and course dictionaries are defined to store student and course information.

### HTML Template

- The HTML template (`index.html`) is used to structure and render the webpage.
- It includes placeholders for displaying student details and course information.

### Form Submission and Data Retrieval

- The user enters a student code in the form and submits it.
- The server receives the student code, retrieves the corresponding student information, and processes it.

### Rendering Data in HTML

- If the student information is found, the server renders the data in the HTML template.
- Student details and courses taken are displayed with appropriate formatting.

## Conclusion

The "Student Information" webpage provides a user-friendly interface to retrieve and display student details and their course information. It demonstrates the integration of Flask, HTML templates, and data retrieval to create a dynamic and informative webpage.