Custom Course Details Generator

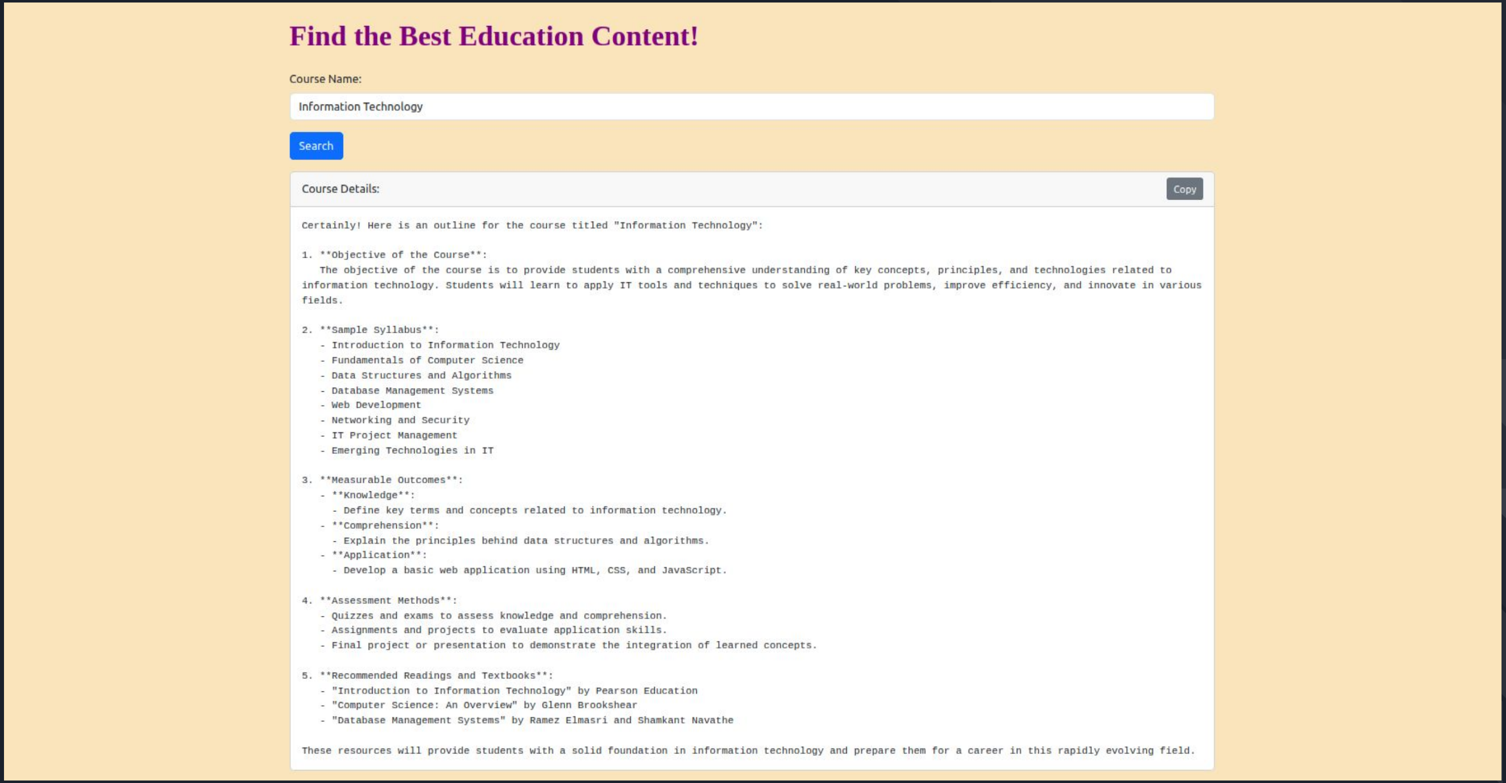
Bolt IoT - Artificial Intelligence Training

Yogesh Prashant Rane
yogeshrane.contact@gmail.com
+91 9527790023

Develop a web-based AI tool that dynamically generates educational content for any given course title. This tool will use HTML, JavaScript, and OpenAI's API to create a user-friendly interface where educators and students can input a course or subject title and receive a detailed, AI-generated outline that includes:

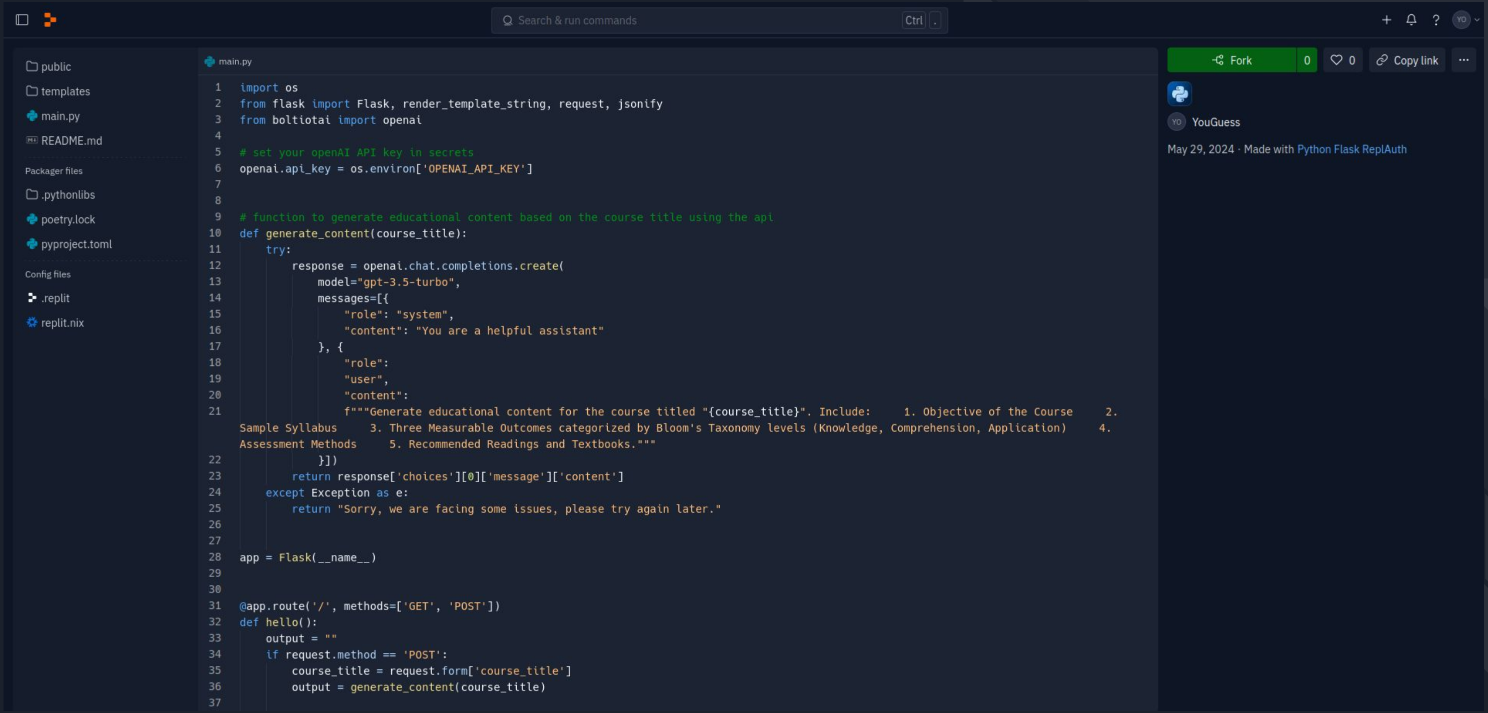
1. **Objective of the Course**: A concise statement that describes the purpose and goals of the course.
2. **Sample Syllabus**: An AI-generated syllabus outline that covers the main topics and modules to be taught.
3. **Three Measurable Outcomes**: Specific, measurable learning outcomes categorized according to Bloom's Taxonomy levels: Knowledge, Comprehension, and Application.
4. **Assessment Methods**: Suggestions on how to evaluate the learning outcomes through various forms of assessment.
5. **Recommended Readings and Textbooks**: A list of AI-recommended resources, including books, articles, and other materials relevant to the course content

Project View



```
Certainly! Here is an outline for the course titled "Information Technology":

1. **Objective of the Course**:
   The objective of the course is to provide students with a comprehensive understanding of key concepts, principles, and technologies related to
information technology. Students will learn to apply IT tools and techniques to solve real-world problems, improve efficiency, and innovate in various
fields.

2. **Sample Syllabus**:
   - Introduction to Information Technology
   - Fundamentals of Computer Science
   - Data Structures and Algorithms
   - Database Management Systems
   - Web Development
   - Networking and Security
   - IT Project Management
   - Emerging Technologies in IT

3. **Measurable Outcomes**:
   - **Knowledge**:
     - Define key terms and concepts related to information technology.
   - **Comprehension**:
     - Explain the principles behind data structures and algorithms.
   - **Application**:
     - Develop a basic web application using HTML, CSS, and JavaScript.

4. **Assessment Methods**:
   - Quizzes and exams to assess knowledge and comprehension.
   - Assignments and projects to evaluate application skills.
   - Final project or presentation to demonstrate the integration of learned concepts.

5. **Recommended Readings and Textbooks**:
   - "Introduction to Information Technology" by Pearson Education
   - "Computer Science: An Overview" by Glenn Brookshear
   - "Database Management Systems" by Ramez Elmasri and Shamkant Navathe

These resources will provide students with a solid foundation in information technology and prepare them for a career in this rapidly evolving field.
```

Link to the Replit of project:
https://replit.com/@YouGuess/customcoursedetailsgenerator?v=1



May 29, 2024 · Made with Python Flask ReplAuth

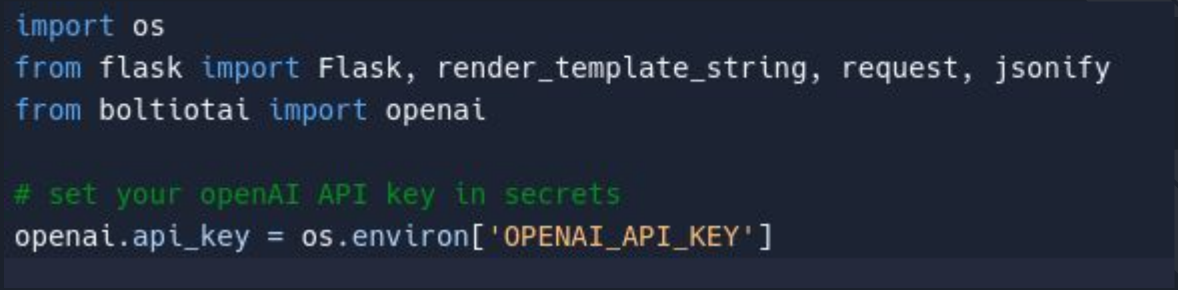
```python
import os
from flask import Flask, render_template_string, request, jsonify
from boltiotai import openai

# set your openAI API key in secrets
openai.api_key = os.environ['OPENAI_API_KEY']


# function to generate educational content based on the course title using the api
def generate_content(course_title):
    try:
        response = openai.chat.completions.create(
            model="gpt-3.5-turbo",
            messages=[{
                "role": "system",
                "content": "You are a helpful assistant"
            }, {
                "role":
                "user",
                "content":
                f"""Generate educational content for the course titled "{course_title}". Include:     1. Objective of the Course     2.
Sample Syllabus     3. Three Measurable Outcomes categorized by Bloom's Taxonomy levels (Knowledge, Comprehension, Application)     4.
Assessment Methods     5. Recommended Readings and Textbooks."""
            }])
        return response['choices'][0]['message']['content']
    except Exception as e:
        return "Sorry, we are facing some issues, please try again later."


app = Flask(__name__)


@app.route('/', methods=['GET', 'POST'])
def hello():
    output = ""
    if request.method == 'POST':
        course_title = request.form['course_title']
        output = generate_content(course_title)

```

How to use?

- Open the webpage.
- Type in the course name of your requirement.
- Press the “Search” button and wait.
- You will get detailed information about your course within a few seconds!

Writing the Code

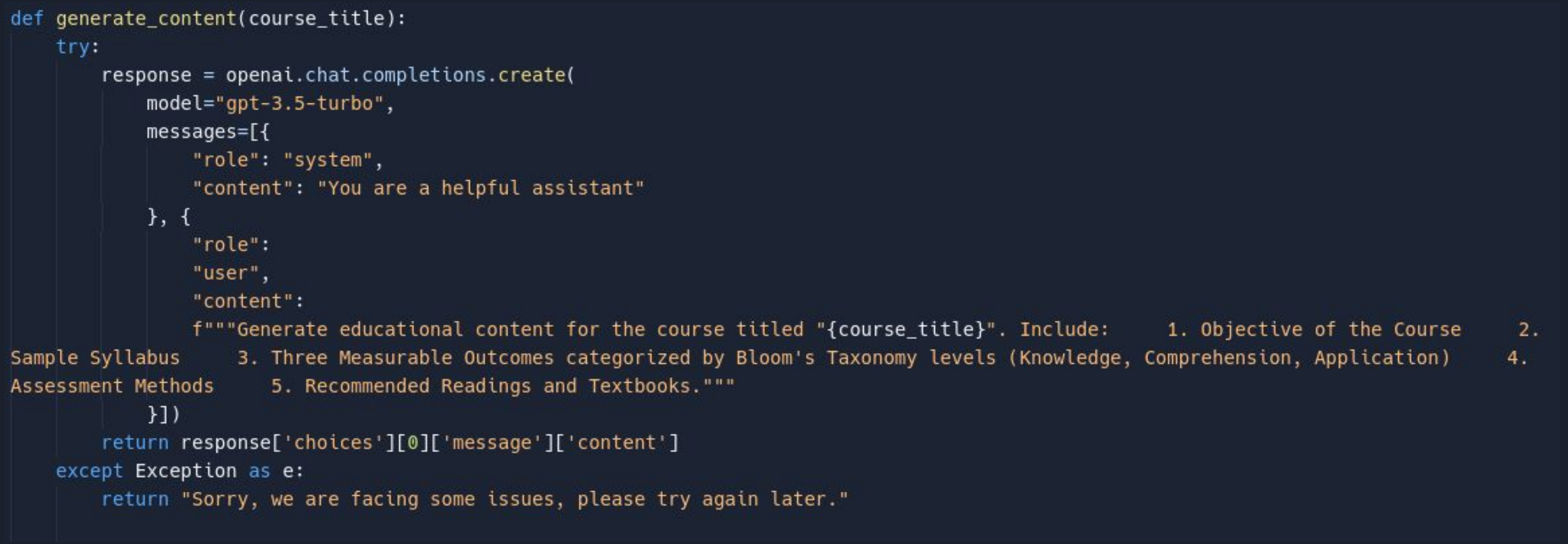
1. Importing the necessary Python modules and packages for building a web application that uses OpenAI's API



```
import os
from flask import Flask, render_template_string, request, jsonify
from boltiotai import openai

# set your openAI API key in secrets
openai.api_key = os.environ['OPENAI_API_KEY']
```

- **'os'**: Interacts with the operating system to fetch environment variables.
- **'Flask'** components (Flask, render_template_string, request, jsonify): Builds and manages the web application.
- **'openai'** from **'boltiotai'** : Accesses OpenAI's API for generating content.

2. **‘generate_content’** to generate educational content

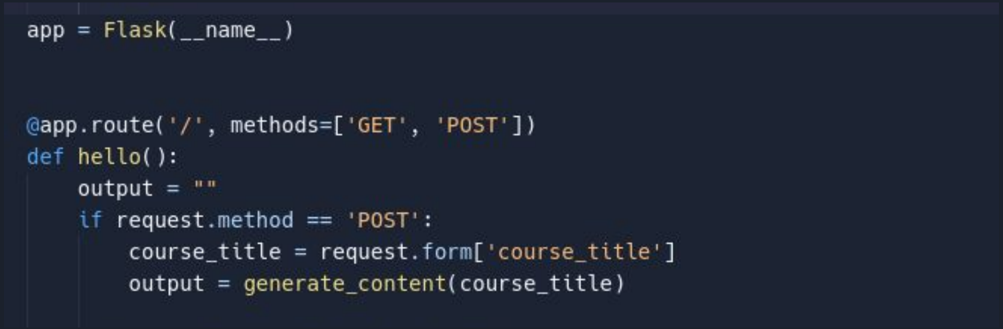


```python
def generate_content(course_title):
    try:
        response = openai.chat.completions.create(
            model="gpt-3.5-turbo",
            messages=[{
                "role": "system",
                "content": "You are a helpful assistant"
            }, {
                "role":
                "user",
                "content":
                f"""Generate educational content for the course titled "{course_title}". Include:     1. Objective of the Course     2.
Sample Syllabus     3. Three Measurable Outcomes categorized by Bloom's Taxonomy levels (Knowledge, Comprehension, Application)     4.
Assessment Methods     5. Recommended Readings and Textbooks."""
            }])
        return response['choices'][0]['message']['content']
    except Exception as e:
        return "Sorry, we are facing some issues, please try again later."
```

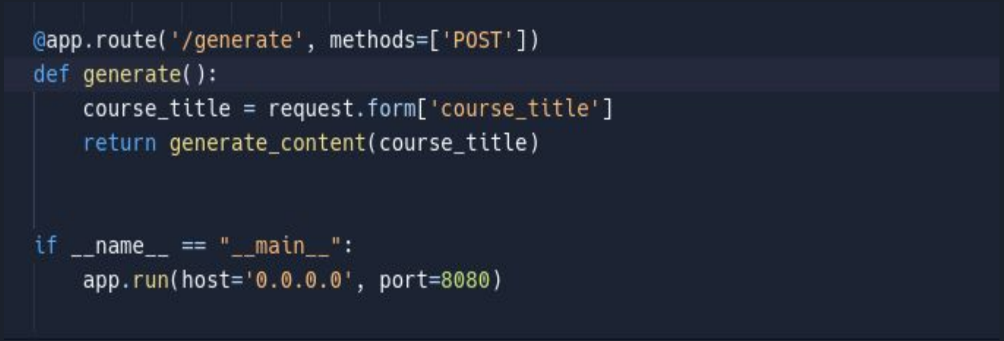
2. **‘generate_content’** to generate educational content

- The function calls the **‘OpenAI API’** to create a chat completion using the **‘GPT-3.5-turbo model’**.
- It sends a **prompt** to the model, asking it to generate educational content for the specified **course_title**.
- The prompt instructs the model to include specific sections: **objectives** of the course, a **sample syllabus**, three **measurable outcomes** categorized by **Bloom's Taxonomy** levels (knowledge, comprehension, application), **assessment methods**, and recommended **readings and textbooks**.
- If the API call is **successful**, the function **extracts** the generated **content** from the API response and returns it.
- If an **error** occurs during the API call, the function catches the exception and returns a message indicating that there are issues, asking the user to try again later.

3. Using Python **‘Flask’**



```python
app = Flask(__name__)


@app.route('/', methods=['GET', 'POST'])
def hello():
    output = ""
    if request.method == 'POST':
        course_title = request.form['course_title']
        output = generate_content(course_title)
```



```python
@app.route('/generate', methods=['POST'])
def generate():
    course_title = request.form['course_title']
    return generate_content(course_title)


if __name__ == "__main__":
    app.run(host='0.0.0.0', port=8080)
```

3. Using Python **'Flask'**

- **Create Flask App Instance**

```
app = Flask(__name__)
```

This line initializes a Flask web application by creating an instance of the Flask class.

- **Define Root Route ('/'):**

```
@app.route('/', methods=['GET', 'POST'])
def hello():
    output = ""
    if request.method == 'POST':
    course_title = request.form['course_title']
    output = generate_content(course_title)
```

❏ This function handles the root URL (/) of the web application. It supports both GET and POST methods.
❏ If the request method is GET, it will display the initial form.
❏ If the request method is POST, it extracts the course_title from the submitted form data and uses it to call the generate_content function. The generated content is then stored in the output variable.

3. Using Python **‘Flask’**

- **Define Generate Route ('/generate')**:

```
        @app.route('/generate', methods=['POST'])
    def generate():
        course_title = request.form['course_title']
        return generate_content(course_title)
```

  - ❑ This function handles the /generate route, it only accepts POST requests.
  - ❑ It retrieves the course_title from the submitted form data and calls the generate_content function with this title.
  - ❑ The function directly returns the generated content as the response.

- **Run the Flask App**:

```
        if __name__ == "__main__":
        app.run(host='0.0.0.0', port=8080)
```

This block ensures that the Flask app runs when the script is executed directly. It starts the Flask web server, making the application accessible at port 8080.

3. Code for **'Webpage'**

- **HTML and Content Structure**

```
		<!DOCTYPE html>
<html>
<head>
		<title>Course Content Suggestor</title>
		<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css" rel="stylesheet">
</head>
<body style="background-color:#F9E4BC;">

<div class="container">
		<h1 class="my-4" style="color:purple; font-family:Times New Roman; font-weight: bold;">Find the Best Education Content!</h1>
		<form id="tutorial-form" onsubmit="event.preventDefault(); generateTutorial();" class="mb-3">
		<div class="mb-3">
		<label for="course_title" class="form-label">Course Name:</label>
		<input type="text" class="form-control" id="course_title" name="course_title" placeholder="Enter the course title you want educational content for" required>
		</div>
		<button type="submit" class="btn btn-primary">Search</button>
		</form>
		<div class="card">
		<div class="card-header d-flex justify-content-between align-items-center">
		Course Details:
		<button class="btn btn-secondary btn-sm" onclick="copyToClipboard()">Copy</button>
		</div>
		<div class="card-body">
		<pre id="output" class="mb-0" style="white-space: pre-wrap;">{{ output }}</pre>
		</div>
		</div>
</div>
</body>
</html>
```

3. Code for **'Webpage'**

- **HTML and Content Structure**
  - ❏ The <head> section includes the title of the page and a link to the Bootstrap CSS for styling.
  - ❏ The <body> section starts with a background color set to #F9E4BC.
  - ❏ The container class from Bootstrap is used to center the content with some padding.
  - ❏ The <h1> tag displays the page title with specific styling.
  - ❏ The <form> element contains an input for the course title and a submit button. The onsubmit event is set to call generateTutorial and prevent the default form submission.
  - ❏ The card component is used to display the generated course details.
  - ❏ The card header contains a button to copy the generated content to the clipboard.The card body contains a <pre> element with the output ID to display the generated content, using white-space: pre-wrap to preserve formatting.

3. Code for **'Webpage'**

- **JavaScript Functions**

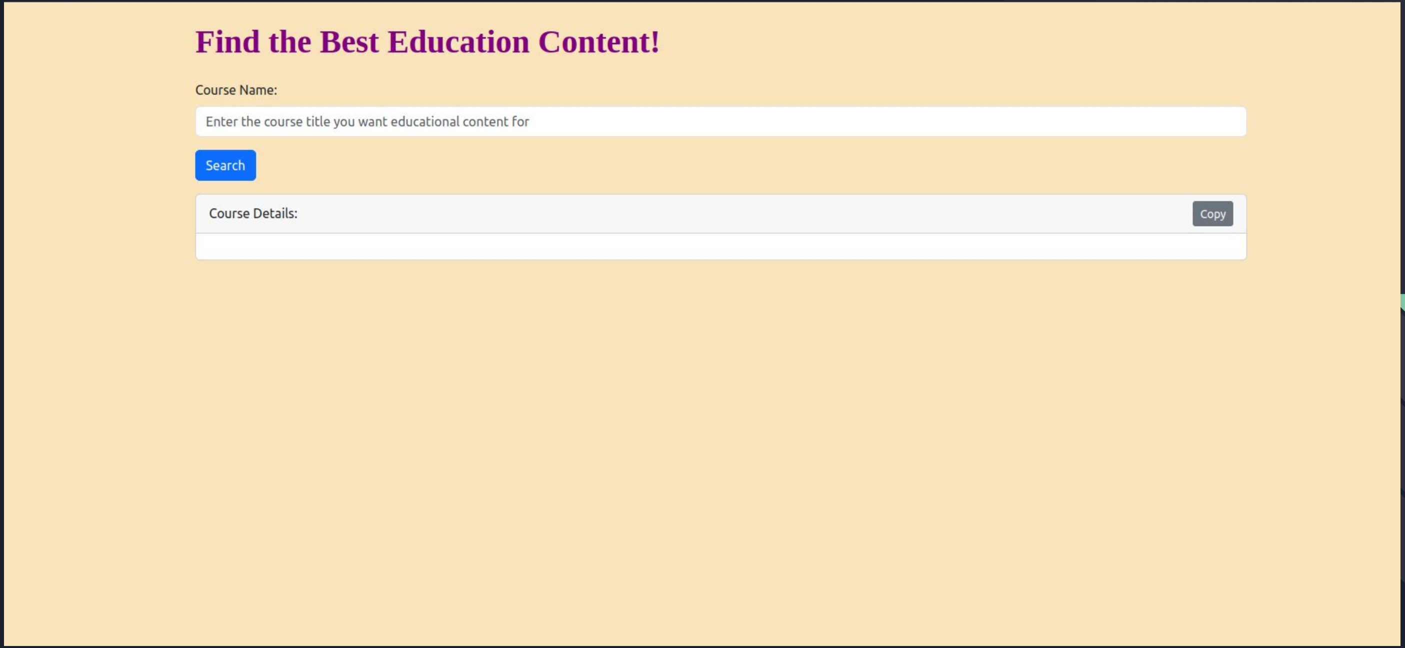
```
<script>
async function generateTutorial() {
const course_title = document.querySelector('#course_title').value;
const output = document.querySelector('#output');
output.textContent = 'Finding the best content for you...';
const response = await fetch('/generate', {
method: 'POST',
body: new FormData(document.querySelector('#tutorial-form'))
});
const newOutput = await response.text();
output.textContent = newOutput;
}

function copyToClipboard() {
const output = document.querySelector('#output');
const textarea = document.createElement('textarea');
textarea.value = output.textContent;
document.body.appendChild(textarea);
textarea.select();
document.execCommand('copy');
document.body.removeChild(textarea);
alert('Copied to clipboard');
}
</script>
```

3. Code for **‘Webpage’**

- **JavaScript Functions**

  ★ **generateTutorial Function**:
    - This function is triggered when the form is submitted.
    - It fetches the value of the course title entered by the user.
    - It sets the output element's text to "Finding the best content for you...".
    - It makes an asynchronous POST request to the /generate route with the form data.
    - It waits for the response, retrieves the text, and updates the output element with the generated content.

  ★ **copyToClipboard Function**:
    - This function copies the content of the output element to the clipboard.
    - It creates a temporary textarea element, sets its value to the content of the output element, and appends it to the document body.
    - It selects the text in the textarea, copies it to the clipboard, and then removes the textarea element.
    - It shows an alert indicating the content has been copied.

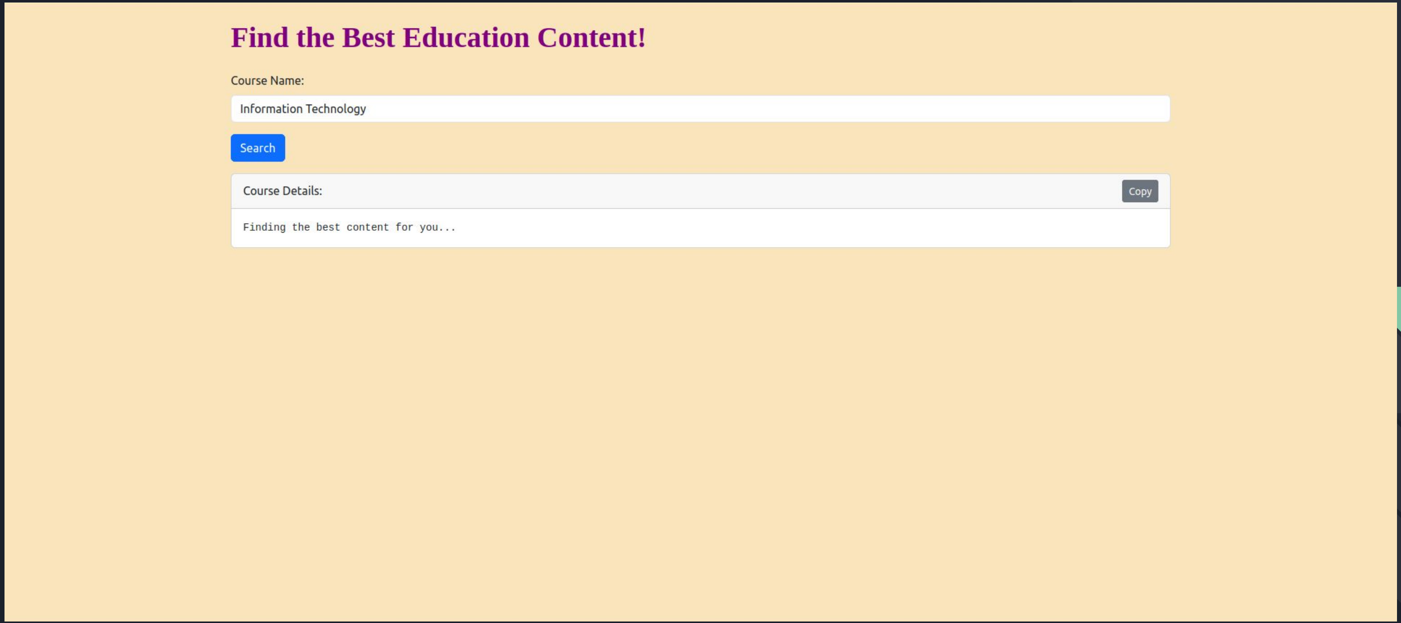


Find the Best Education Content!

Course Name:

Enter the course title you want educational content for

Default webpage screen.



Find the Best Education Content!

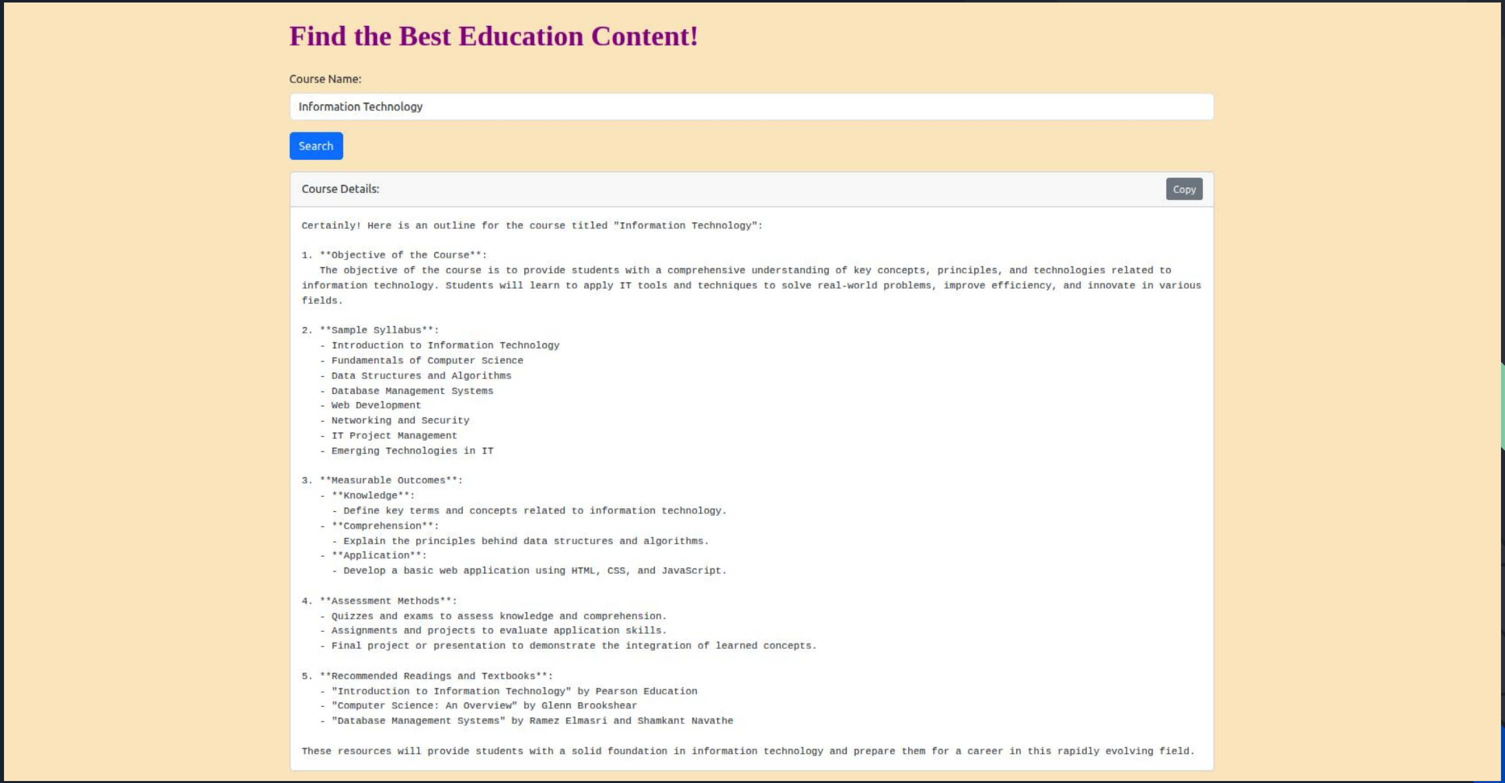
Course Name:

Information Technology

Course Details:

```
Finding the best content for you...
```

Waiting for course contents and other details.



```
Certainly! Here is an outline for the course titled "Information Technology":

1. **Objective of the Course**:
   The objective of the course is to provide students with a comprehensive understanding of key concepts, principles, and technologies related to
information technology. Students will learn to apply IT tools and techniques to solve real-world problems, improve efficiency, and innovate in various
fields.

2. **Sample Syllabus**:
   - Introduction to Information Technology
   - Fundamentals of Computer Science
   - Data Structures and Algorithms
   - Database Management Systems
   - Web Development
   - Networking and Security
   - IT Project Management
   - Emerging Technologies in IT

3. **Measurable Outcomes**:
   - **Knowledge**:
     - Define key terms and concepts related to information technology.
   - **Comprehension**:
     - Explain the principles behind data structures and algorithms.
   - **Application**:
     - Develop a basic web application using HTML, CSS, and JavaScript.

4. **Assessment Methods**:
   - Quizzes and exams to assess knowledge and comprehension.
   - Assignments and projects to evaluate application skills.
   - Final project or presentation to demonstrate the integration of learned concepts.

5. **Recommended Readings and Textbooks**:
   - "Introduction to Information Technology" by Pearson Education
   - "Computer Science: An Overview" by Glenn Brookshear
   - "Database Management Systems" by Ramez Elmasri and Shamkant Navathe

These resources will provide students with a solid foundation in information technology and prepare them for a career in this rapidly evolving field.
```

Webpage with required contents of the course.

Thank you.