

Overview

In this task, I focused on setting up and testing the Flask API endpoint that returns MCQs and flashcards based on the lecture summary.

Task Summary

- I created the `/generate_quiz` Flask route to accept summary input
 - I used the functions `build_prompt()` and `call_ai_model()` provided by Person A
 - I returned the output in JSON format
 - I tested everything using Postman, including edge cases
-

Flask Integration (app.py)

```
from utils.quiz_generator import build_prompt, call_ai_model
```

```
@app.route('/generate_quiz', methods=['POST'])
```

```
def generate_quiz():
```

```
    data = request.get_json()
```

```
    summary = data.get('summary', '')
```

```
    if not summary:
```

```
        return jsonify({'error': 'Summary is required'}), 400
```

```
    prompt = build_prompt(summary)
```

```
    result = call_ai_model(prompt)
```

```
    return jsonify(result)
```

Postman Testing Steps

1. I opened Postman and created a new **POST** request
2. Set the **URL** to: `http://localhost:5000/generate_quiz`

3. Added the following header:
Content-Type: application/json
4. In the **Body** → **raw** → **JSON**, I entered:

```
{  
  "summary": "Machine learning is a subset of AI that helps computers learn from data."  
}
```

5. Clicked **Send** and reviewed the response.

Expected Output Format

json

CopyEdit

```
{  
  "mcqs": [  
    {  
      "question": "...",  
      "options": {  
        "A": "...",  
        "B": "...",  
        "C": "...",  
        "D": "..."  
      },  
      "correct_answer": "A"  
    },  
    {  
      "question": "...",  
      "options": {  
        "A": "...",  
        "B": "...",  
        "C": "...",  
        "D": "..."  
      },  
      "correct_answer": "B"  
    },  
    {  
      "question": "...",  
      "options": {  
        "A": "...",  
        "B": "...",  
        "C": "...",  
        "D": "..."  
      },  
      "correct_answer": "C"  
    },  
    {  
      "question": "...",  
      "options": {  
        "A": "...",  
        "B": "...",  
        "C": "...",  
        "D": "..."  
      },  
      "correct_answer": "D"  
    }  
  ],  
  "flashcards": [  
    {  
      "question": "...",  
      "answer": "..."  
    },  
    {  
      "question": "...",  
      "answer": "..."  
    },  
    {  
      "question": "...",  
      "answer": "..."  
    },  
    {  
      "question": "...",  
      "answer": "..."  
    }  
  ]  
}
```

```
"question": "...",  
"answer": "..."  
}  
]  
}
```

Edge Case Testing

- Sent an **empty summary** → Got an error as expected
 - Tried a **very short summary** → Got fallback content or still valid structure
 - Sent **no 'summary' key** → Got 400 error
-

File Structure Summary

project-root/

```
├─ app.py          ← I added the route here  
├─ utils/  
├─ ┬─ quiz_generator.py  ← Used functions from Person A  
├─ ┬─ test_quiz_generation.py ← Optional testing file  
├─ ┬─ requirements.txt
```

Final Checklist

- /generate_quiz route implemented
- API tested in Postman with various inputs
- Output JSON validated
- Error handling checked