Basic Quiz Application

Step 1: Define the Quiz Data

First, we need to set up the quiz questions, choices, and correct answers.

```python
# quiz_data.py
quiz_data = [
    {
        "question": "What is the capital of France?",
```

```
        "choices": ["Paris", "London", "Berlin", "Madrid"],
        "answer": "Paris"
    },
    {
        "question": "Which planet is known as the Red Planet?",
        "choices": ["Earth", "Mars", "Jupiter", "Venus"],
        "answer": "Mars"
    },
    {
        "question": "What is the largest ocean on Earth?",
```

```
        "choices": ["Atlantic", "Indian",
"Southern", "Pacific"],
        "answer": "Pacific"
    }
    # Add more questions as needed
]
```

Step 2: Create the Quiz Logic

Next, we will write the logic to display the quiz, collect user answers, and calculate the score.

```python
# quiz_app.py
```

```
from quiz_data import quiz_data


def run_quiz():
    score = 0
    for idx, item in enumerate(quiz_data):
        print(f"Q{idx + 1}: {item['question']}")
        for i, choice in enumerate(item['choices']):
            print(f"  {i + 1}. {choice}")

        answer = input("Your answer (1-4): ")

        try:
            answer_idx = int(answer) - 1
```

```
            if item['choices'][answer_idx] ==
item['answer']:
                print("Correct!\n")
                score += 1
            else:
                print(f"Wrong! The correct answer is:
{item['answer']}\n")
        except (ValueError, IndexError):
            print(f"Invalid input. The correct
answer is: {item['answer']}\n")


    print(f"Your final score is {score} out of
{len(quiz_data)}.")
```

```
if __name__ == "__main__":
    run_quiz()
```

Explanation

1. **Quiz Data**: The `quiz_data` is a list of dictionaries where each dictionary represents a quiz question, the possible choices, and the correct answer.

2. **Quiz Logic**: The `run_quiz` function iterates over the `quiz_data`, displays each question and its choices, collects user input, checks if the input is correct, and calculates the score.

3. **Error Handling**: The function includes basic error handling for invalid inputs (e.g., non-integer values or out-of-range numbers).

Running the Quiz

To run the quiz, save the `quiz_data.py` and `quiz_app.py` files in the same directory and run `quiz_app.py` using Python:

```sh
python quiz_app.py
```

Expanding the Application

Here are a few suggestions to expand this basic quiz application:

- **Graphical User Interface (GUI)**: Use libraries like `tkinter` or `PyQt` to create a GUI for the quiz.

- **Persistent Storage**: Store quiz data in a database (e.g., SQLite) or a file (e.g., JSON) for easy updates and scalability.

- **Multiple Quiz Categories**: Implement different categories of quizzes and allow the user to select a category.

- **Timed Quiz**: Add a timer for each question to increase the difficulty level.

- **Score Tracking**: Save scores to a file or database to track user progress over time.