Project Title: Scripture Memorizer System

Project Description: In this project, you will create a desktop application that helps users memorize scriptures. The application will allow users to select a scripture from a list of available scriptures, and then display the scripture in a customizable format. The user can then practice memorizing the scripture by hiding certain words or phrases and trying to recall them from memory. The application will track the user's progress and provide feedback to help them improve their memorization skills.

Project Features:

- A database of scriptures that can be selected by the user.

- Customizable display options for the selected scripture, such as font size, color, and layout.

- A "hide" feature that allows the user to hide certain words or phrases in the scripture.

- A timer feature that tracks the user's progress and provides feedback on their memorization speed.

- A scoring system that rewards the user for correct answers and encourages them to improve their performance.

- A "review" feature that allows the user to review previously memorized scriptures and track their progress over time.

- A user account system that allows users to save their progress and track their performance over time.

Steps to be taken:

1. Database design: Create a database of scriptures that can be selected by the user. Each scripture should include the text and a unique identifier. Create a database schema and a set of tables to store the data.

2. User interface design: Design a user-friendly interface that allows users to select a scripture,customize the display options, and practice memorizing the scripture. Use Windows Forms to create the user interface.

3. Back-end development: Use C# to create the business logic of the application. Implement features such as the "hide" feature, timer, scoring system, and review feature.

4. Database integration: Integrate the scripture database with the application using ADO.NET or Entity Framework.

5. User authentication: Create a user authentication system that allows users to create accounts, log in, and save their progress. Use ASP.NET Identity or a similar framework to handle user authentication.

6. Testing and debugging: Test the application thoroughly to ensure it works correctly and handles errors properly.

Diagram:

```

User Interface <--> Business Logic <--> Database

```

The user interface interacts with the business logic, which communicates with the database. User authentication and data storage are handled by the database. The business logic provides the functionality of the application, while the user interface and database handle the presentation and data storage aspects of the application.

