**🌟 Epics and User Stories Framework**

**Epic 1: Bible Viewing & Navigation**

* **User Story 1.1:** As a user, I want to view a specific Bible chapter in my selected version so that I can read and study it.
  + Acceptance Criteria:
    - The user can select a Bible version.
    - The user can navigate to any chapter.
    - The chapter is displayed cleanly in the chosen UI.
* **User Story 1.2:** As a user, I want to quickly switch between different Bible versions so that I can compare translations.
  + Acceptance Criteria:
    - Bible version selection dropdown/menu.
    - The app updates the displayed chapter instantly when the version is switched.
* **User Story 1.3:** As a user, I want to easily navigate to different books and chapters so that I can read the entire Bible.
  + Acceptance Criteria:
    - Navigation tree or dropdown for books.
    - Chapter numbers for each book.

**Epic 2: Search & AI Interaction**

* **User Story 2.1:** As a user, I want to search for a Bible verse or keyword so that I can find passages relevant to my study.
  + Acceptance Criteria:
    - Search box.
    - Search results with links to chapters.
* **User Story 2.2:** As a user, I want to ask questions about the Bible in natural language (chatGPT-like interface) so that I can understand difficult passages or get study help.
  + Acceptance Criteria:
    - ChatGPT interface.
    - It pulls data from Bible APIs and commentary databases.

**Epic 3: User Profiles & Personalization**

* **User Story 3.1:** As a user, I want to create an account so that my preferences and notes are saved.
  + Acceptance Criteria:
    - Sign-up/login forms.
    - Account creation and secure authentication.
* **User Story 3.2:** As a user, I want to save notes on specific Bible verses or chapters so that I can reference them later.
  + Acceptance Criteria:
    - Notes linked to specific verses/chapters.
    - Ability to edit or delete notes.
* **User Story 3.3:** As a user, I want to personalize my reading experience (font, color themes, etc.) so that it’s comfortable for me.
  + Acceptance Criteria:
    - User profile settings for customization.
    - Settings are persistent across sessions.

**Epic 4: Community & Sharing**

* **User Story 4.1:** As a user, I want to share my notes with others so that we can discuss and learn together.
  + Acceptance Criteria:
    - Option to share notes with community or specific users.
    - Access control for notes.
* **User Story 4.2:** As a user, I want to participate in discussion boards so that I can discuss passages or topics with the community.
  + Acceptance Criteria:
    - Forum/discussion board interface.
    - Threads organized by topic.

**Epic 5: Flexible & Extendable Frontend**

* **User Story 5.1:** As a developer, I want the front end to be modular and extendable so that new features can be added easily.
  + Acceptance Criteria:
    - Use a component-based front-end framework (e.g., React, Vue).
    - Clear documentation on how to add new UI components.
* **User Story 5.2:** As a developer, I want the app to support multiple data sources (APIs, databases, indexes) so that it can integrate with different providers.
  + Acceptance Criteria:
    - Abstract data layer with adapters for different sources.
    - Ability to plug in new data sources with minimal changes to front-end logic.

**🔧 Supporting Capabilities & Future Extensions**

Here’s how this framework supports your key requirements and flexibility for growth:

✅ **User Management & Profiles**

* Secure login/authentication (OAuth, JWT).
* User-specific settings, bookmarks, and notes.

✅ **Bible Data Handling**

* API-driven architecture to pull Bible text and commentary from multiple APIs.
* Data layer that can switch between API, local DB, or search index.

✅ **UI Flexibility**

* Front-end built on a modular, component-based framework (React or Vue recommended).
* Plugin architecture for adding new features like notes, discussions, video/audio, etc.

✅ **Community & Sharing**

* Sharing of notes with permissions.
* Discussion board as a separate module (could be powered by tools like Discourse, Firebase, or custom-built).

✅ **ChatGPT Interface**

* Integrated AI assistant to answer questions and provide contextual insights.

**🏗️ Next Steps: Building the Product Backlog**

Would you like me to:

✅ Break these epics and user stories down into a detailed **product backlog** with priorities and tasks?  
✅ Sketch out a **system architecture diagram** to visualize how APIs, databases, and front end will work together?  
✅ Provide **technology recommendations** for each part of the stack (e.g., front-end frameworks, authentication systems, API integrations)?