Assignment: Building a Tabbed iOS Application with Navigation and Detail Views

Objective

Develop an iOS application that features:

- A **Tab Bar** with at least **three tabs**.
- Each tab contains a **Table View** listing favorite items (e.g., **Films**, **Books**, **Music**).
- Selecting an item navigates to a **Detail View** using a **Navigation Stack**, allowing users to go back via the navigation bar.

Requirements

1. Tab Bar Implementation

- Create a **Tab Bar Controller** with at least **three tabs**.
- Each tab represents a category (e.g., **Films**, **Books**, **Music**).
- Assign appropriate **titles** and **icons** to each tab.

2. Table Views in Tabs

- Each tab contains a **Table View Controller**.
- Populate each table with a list of at least **10 favorite items** relevant to the category.
- Display relevant information in each cell (e.g., title, subtitle, thumbnail image).

3. Detail View Navigation

- When a user selects an item, push a **Detail View Controller** onto the navigation stack.
- The detail view should display comprehensive information about the selected item (e.g., description, release date, cover image).

4. Navigation Controllers

- Embed each **Table View Controller** in a **Navigation Controller**.
- Ensure the **Navigation Bar** is visible and functional in all views within the navigation stack.

5. User Interface Design

- Design a clean and intuitive **User Interface**.
- Use **Auto Layout Constraints** for responsiveness across different device sizes and orientations.

Deliverables

1. **Xcode Project**: Submit the complete Xcode project folder.
2. **Source Code**: Ensure code is well-organized, properly indented, and commented.
3. **Video Demo**: Include screen recording video demo of your app

Getting Started Tips

- **Tab Bar Controller**: In Interface Builder, drag a Tab Bar Controller onto the storyboard or set it up programmatically.
- **Navigation Controller**: Embed your Table View Controllers in Navigation Controllers (Editor > Embed In > Navigation Controller).
- **Table Views**: Use UITableViewDataSource and UITableViewDelegate protocols to manage data and interactions.
- **Detail View**: Create a new view controller for the detail view and pass data from the selected table cell.
- **Data Passing**: Use **Segue** identifiers and the prepare(for:sender:) method to pass data to the Detail View Controller.