



IOS APP Development

Section 1 – iOS App Development Introduction

- **Single View App**
- **Label**
- **TextField**
- **Button**
- **Stepper TextView**
- **Slider**
- **DataPicker**
- **MultiScreen App**
- **Alert Controller**
- **Controller Lifecycle**
- **Image View**
- **AutoLayout**
- **Size Classes**
- **Scrollview**
- **Field Validation TableView**
- **Custom Cell Tableview**
- **CollectionView**
- **Multithreading and GCD(Swift+App)Gestures**
- **Popover Controller**
- **Gallery And Camera Image Assets and App icon**
- **Animations**
- **Custom View with Xib Navigation Controller With Code(push/pop)**
- **Navigation Controller With Storyboard**
- **Audio and Video Player**
- **SQLite Database And SQL**
- **SQLite Integration**
- **Navigation Drawer**
- **Tab Bar Controller**
- **JSON Parsing GET, POST, PUT, DELETE**
- **Cocoa Pod Integration**
- **UserDefault And Modal Segue**
- **Core Data CRUD Operation**

Section 2 – iOS App Development Advanced

- **AlamoFire CocoaPod**
- **SwiftyJSON CocoaPod**

- **Local Notifications**
- **Push Notifications**
- **Google Integration**
- **OTP Verification**
- **MVC vs. MVVM Pattern**
- **Location Tracking**
- **GeoCoding**
- **Publishing App**

1. Create a single-view app that displays a label, text field, and button. Implement functionality where the button changes the label text based on the input from the text field.
2. Build an app with a stepper and a text view. The stepper increments or decrements a value, and the text view displays the updated value.
3. Develop a multi-screen app with a navigation controller. Implement a login screen that navigates to a home screen upon successful authentication.
4. Create an alert controller that displays a message and allows the user to perform an action, such as confirming or canceling an operation.
5. Implement an image view that displays an image fetched from a URL using asynchronous loading techniques.
6. Design a form with various input fields (text fields, sliders, data pickers) and perform field validation to ensure the entered data is valid before submitting.
7. Create a table view with custom cells that display data fetched from a JSON API. Implement pagination and pull-to-refresh functionality.
8. Build a collection view that displays a grid of images fetched from a server using JSON parsing and asynchronous loading.
9. Implement multi-threading and Grand Central Dispatch(GCD) to perform time-consuming tasks in the background while keeping the UI responsive.
10. Add gesture recognize to views, such as tap, swipe, and pinch gestures, and implement corresponding actions.

-
1. Design a popover controller that presents additional information or options when a button or view is tapped.
 2. Develop an app that allows users to select images from the gallery or take photos using the camera. Implement functionality to save and display the selected images.
 3. Add animations to UI elements, such as fade-in, slide-in, or rotation animations, to enhance the user experience.
 4. Create a tab bar controller with multiple tabs, each displaying different content or functionalities within the app.
 5. Implement a navigation controller programmatically for pushing and popping view controllers onto the navigation stack.
 6. Design a navigation controller using storyboards with multiple view controllers and implement navigation between them.
 7. Build an audio and video player app that plays media files from local or remote sources.
 8. Integrate SQLite database into an app and perform CRUD operations, such as creating, reading, updating, and deleting data records.
 9. Implement a navigation drawer (side menu) that slides in from the side and provides navigation options to different app sections.