Flutter Development Path Way

Topics Page Numbers

- **Data Introduction**
  - Overview
  - Introduction to dart programming language
  - Types, syntaxes, classes in dart
  - Object-oriented programming with dart
  - Constructors in dart.

- **Staying On Target With Dart**
  - Making a small application with dart itself. A card game, where players can shuffle, draw, remove cards with the use of object-oriented programming.
  - A more look at object-oriented programming design flow
  - Methods
  - Loops
  - Lists
  - Named parameters etc.

- **App Building With Flutter**
  - Introduction to flutter
  - App overview
  - Design process
  - Displaying content on the screen
  - Creating and using custom widgets

- **State Design With Flutter**
  - Stateful vs stateless widgets
  - Build method
  - Refactoring stateless and stateful widgets.

- **HTTP Request With Flutter**
  - Working with json
  - Casting json to model instances
  - Function references
  - Issuing http requests
  - Working with dart futures
  - Building list of widgets

- **Forms And Validation With Flutter**
  - Creating the login screen
  - Handling input types
  - Form validation
  - Global key referencing
  - Retrieving form values
  - Submitting the form
  - Code reuse with mixin
  - Mixin validator implementation

- **Reactive Programming With Dart**
  - Streams by analogy
  - Implementing stream listener
  - Adding stream transformer
  - Validation with streams

- **Advanced Flutter Architecture - The BLOC Pattern**
  - BLOC vs stateful widgets
  - The purpose of BLOC
  - BLOC design for text fields
  - Issues with BLOC access
  - Improving the BLOC API
  - Validator transformer
  - Cleaning up the controllers
  - BLOC application

- **Consuming BLOC Data**
  - The stream builders
  - Scoped BLOC approach
  - Provider implementation
  - Providers in action
  - Accessing the BLOC

- **Rx Dart For Reactive Programming**
  - Introduction to Rx dart
  - Combine latest in Rx dart
  - Interpreting stream values
  - Broadcast streams
  - Replacing controllers by subjects
  - Review of BLOC

- **Building Delightful Animation**
  - Animation library classes
  - Stateful widgets for animation
  - Widget structure
  - The ticker provider
  - Tweens with curves
  - Performing saving with animation builder
  - Nature of animation
  - Observing the use interaction
  - Order of drawing widgets
  - Tween ranges
  - Working with widget rotation
  - Box animation
  - Animated builder

- **Performant Data Fetching**
  - API challenges
  - API performance strategy
  - API provider implementation

- **Testing with Flutter**
  - Testing with dart
  - Writing expectations
  - Mocking http requests

- **Offline Data Storage**
  - SQLite data provider
  - Creating database connections
  - Creating tables with SQLite
  - Issuing queries
  - Massaging database returning maps
  - Implementing the repository
  - Casting lists and type annotation

- **Type Design**
  - Abstract classes
  - Repository interface design
  - Using source and cache
  - Ultimate reusability

- **On Demand Widget Rendering**
  - Data fetching concerns
  - Future builders
  - Problems with stream builder and its solution
  - Implementing scan stream transformer
  - Adding items to the controller
  - Stream subscription
  - Adding streams
  - Refactoring streams
  - Debugging with flutter
  - Conflicts with database and it's solution
  - Long lived cache values
  - Swipe to refresh implementation

- **Navigation With Flutter**
  - Navigation in flutter
  - Different types of routing
  - Parsing route settings
  - Recursive data fetching
  - Testing recursive data fetching
  - Widget stylings
  - Recursive rendering

- **Basic Git Classes**
  - Creating a repository on GitHub
  - Pushing local changes into the repository
  - Pulling and cloning projects from the repository
  - Resolving merge conflicts

- **More On Architecture**
  - Introduction to scoped model architecture
  - Working with scoped model

- **App Publishing**
  - Publishing the app to App store
  - Publishing the app to Google play store

- **Miscellaneous Works**
    - Working with connectivity
    - Working with location and google maps
    - Working with camera
    - Geocoding and reverse geocoding
    - Multipart form requests
    - Creating a custom library and publishing it under pub
    - Code refactoring shortcuts

- **Project Work**
    - At the end of the course, project work is handed to the students.