Kanish Jadhwani
D15B - 22
MPL

Assignment - 1

Q1)

a) Explain the key features and advantages of using flutter for mobile app development.

→) flutter is a cross platform UI toolkit developed by google for building natively compiled applications for mobile, web and desktop from a single codebase key features and advantages include :-

(1) Hot reload :- Enables developers to instantly view changes without restarting the app.

(2) Widget-based architecture : UI components in flutter are widgets making the developement modular customizable

(3) Expressive : Flutter provides a rich set of customizable widget for creating visually appealing interface

(4) Single Codebase :- Develop once, deploy every-where, reducing development time and effort

(b) Discuss how the flutter framework differ from ~~t~~traditional approaches and why it has gained popularity in developer community.

→ 1) Flutter uses a reactive framework, where as traditional approaches are typically imperative.

2) Flutter offers a consistent UI across platform ensuring a native look and feel.

3) The use of dart language and widget based approach enhances developer, prodrctivity.

Q2) a) Describe the concept of the widget tree in flutter explain how widget composition is used to build complex user interface.

→ i) In flutter, the widget is a fundamental concept that represents the hierarchy of user interface elements in an application. Everything in flutter is a widget whether its a button, text, image, or even more children forming the hierarchy.

ii) The widget tree is composed by various types of widget, each serving a specific purpose. widgets in flutter can be bradly categorized into stateless and stateful.

iii) Stateless widgets are immutable and don't have any internet state, while stateful widget can change their internal state during the lifetime

(b) Provide examples of commonly used widgets and their roles in creating a widget tree.

→ Examples of commonly used widgets

i) Material App: Defines the basic structure of a flutter app.

ii) Scaffold: Represents the basic visual structure of the app, including the app bar and body.

iii) Container: A box model than can contain other widget, providing layout and styling

iv) Row and Column: Arrange child widgets horizontally or vertically.

v) Listview: Displays a scrolling list of widget.

Q3) a) Discuss the importance of state management in flutter application.

→ State management: It is a crucial aspect of building robust and efficient flutter applications. In flutter 'State' refers to the data that influences the appearance and behaviour of widgets managing state efficiently is essential for creating responsive, dynamic and scalable applications. here are some key reasons.

Why state management is important in flutter.

i) user interface updates
ii) Performance optimization
iii) Code maintainablity.
iv) Reusability and modularity
v) Persistence and navigation
vi) stateful widget limitations
vii) Concurrency and asynchrous operations.

b) Compare and contrast the different state management approach available in flutter. Such as Setstate, Provider and River Pod. Provide scenarios where each approach is suitable

(i) Set state !-

→ Simplicity !- 'Setstate' is the most straight forward way to manage state in flutter. It is built into the framework and is easy to understand.

PAGE NO.:
DATE : / / 20

- Setstate is appropriate for simple UI's, for Small to moderately complex UI's where the state changes are localized and the widget tree is not deeply nested 'setstate' can be sufficient.
Suitable Scenarios
- Small moderately sized apps.
- Simple UI's with limited interactivity
- learning and prototyping purposes.

(2) Provider
- Provider allows for speed and localized state management, reducing the need for prop drilling
- It is easy to integrate into flutter apps and offers a good balance between Simplicity and flexibility.
- Provider is widely used has good community Support.

Suitable Scenarios:
- Apps of varying sizes with moderate to Complex UI's.
- Situations where a centralized State management Solution is needed but without the complexity of other Solutions.

Amar KRISH Teacher's Signature

(3) Riverport :-
- It is Scoped and flexible
- It is immutable and reactive also provides inheritance.

Suitable Scenarios:
- large and Complex applications
- Situations where a more sophisticated, scalable and reactive state management solution is required
- Projects where dependency injection is crucial.

Q4) Explain the process of integrating firebase with a flutter application. Discuss the benefits of using firebase as a backend solution.

i) Create a firebase project
- Go to the firebase console and create a new project.
- follow the setup instructions,

2) Add firebase to flutter project
- In your flutter project, add the firebase SDK dependencies to the yaml file

3) Initialize firebase :-
- Import the firebase packages and initialize firebase in the main. dart file.

iv) Configure firebase Services :-
- Depending on the types of services you want to use, Configure them by following the specific setup instructions provided by the firebase

v) use firebase service in the App.

PAGE NO.:
DATE.: / / 20

Benefits

i) Real. time database
2) Authentication
3) Cloud function
4) Cloud firestore
5) firebase storage
6) hosting & analysis

Amar KRISH

Teacher's Signature