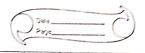
Anushka Karhadkar DISB 28



Teacher's Sign.: _

	Page
	Assignment No-1
	U
0.1.9	Exploin the key features and advantage of using
	Flutter for mobile app development.
	The contract development
-	Flutter is on open source UI software development
3	toolkit created by Google, which enables developers to build natively compiled applications
	for mobile, web, and desktop from a single
	codebase. Here are key features and advantages
	of using Flutter:
	keu Features -
	9) single Codebase: Flutter allows developers to
	write code once and deploy it on both ios and
	Android platforms, reducing development time
	and effort. The single codebase opproach ensures
	consistency across different platforms. ii) Hot Reload: one of Flutter's Standaut features
	is its Hot Reload functionaility. Developers can
	make changes to the code and instantly see the
	results in the running application without
	restarting 1t.
	lii) High Performance: Flutter apps are complied
	to native apps. It doesn't rely on an intermediate
	interpretation layer, resulting in smooth
	animations and excellent performance.
-	



Advantages:

i) Faster Development: With a single radebase and the Hot Reload feature, Flutter significantly reduces development time, allowing developers to iterate quickly and deliver updates faster.

ii) Cost - Effective: Building and maintaining a single radebase for multiple platforms reduces development and maintence costs compared to managing separate codebase.

for POS and Android:

iii) Consistent UI Across Platform: Flutter ensures

- (iii) Consistent UI Across Platform: Flutter ensures

 a consistent look and feel ocross different

 platform, making it easier for developers

 to maintain a unified brand identity

 and user experience.
- b) Discuss how the Flutter Framework works different
 . from traditional approaches and why
 it has gained popularity in the developer
 community.

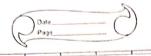
Flutter differs from traditional approaches
in following ways:
i) Widget - Based Architecture: Flutter uses
a widget - based architecture where The
entire UI is constructed using composable
and customizable widgets. This is different

From & traditional approaches that might



Teacher's Sign.: ___

	involve XMI layous.
	if) Dart Programming Language: Flutter uses
	dart as its programming language. Dart is
	language designed for simplicity, speed and
	broauce in the
	iii) cost - Effectiveness Building and maintaing
,	a single codebase for multiple platforms with
6	Flutter can be more cost-effective.
	The second secon
2.2.a)	Describe concept of widget tree in Flutter.
	Explain how widget composition is used to
	build complex mer interfaces.
	ide value to be at the set of
	In flutter, the concept of widget tree is
	pundamental to the Frameworks architecture.
1	The widget tree is a hierarchical struture
	composed of various widge to, where each
6	Widget represents a component or a part
•	of the user interface. Understanding
	the widget tree is crucial for building
	complex and dynamic user interfaces
	efficiently!
	widget composition:
	i) Immutable and Stateley, Stateful widgets:
	widgets are immutable once created
100	a widget connot be changed. However
-	Flutter distinguishes between staless and
- 4	Stateful widgets.
	₹



Teacher's Sign.:

ii) Container Widgets: Basic structural widgets like container, Row, column are used to define layout of The User Interface iii) Text and image widgets: widgets like text and Image are used for displaying. text and images. These can be embedded within other widgets to create a more complex UT. iv) Input widgets: Flutter provides widgets for user input such as TextField, checkbox, Radio, etc. These can be combined with other widge to create forms and interactive elements. v) custom widgets: Developers can create Their own custom widgets by composing existing widgets. b) Provide examples of commonly used widgets and their roles in creating widget tree. Examples of widgets and their roles in widget tree are, i) Contain er : The container widget is a versatile box model that can contain other widget and is used for layout purposes.

```
Container (
   color: colors. blue
   width : 100,
  height: 100,
  Child: Text ('Hello!)
 ii) column and Row
 Role: These widgets allows to arrange child
 widgets vertically or horizontally
   Column (
    children: [
    Text ('Ttem!'),
Text ('Ttem?'),
  Displays a paragraph of text.
  Text ( Hello!)
iv) List View
 Creates a scrollable, linear 11st of widgets
Eg, List View (
        ListTile (Htle: Text ('Item!')).
        ListTile (title: Text('Item 21)),
```

	r) card:
	A material design card that can to contain
	various widgets.
	Eq. Card (
	child: Column (
	children: C
	Image.networc' https://example.com/img.jpg)
)	1 satTile
	tille: Text(' (and Tile'),
	Subtitle: Text ('Card Subtitle')
)
),
(2.0)	Diameter T
g.3·a)	Digcuss The importance of StateManagement
	in Flutter applications.
_	State Management is a critical aspect of
	Flutter application development, and its
	effective handling is crucial for building
	responsive dynamic and maintainable
	user interfaces. State refers to the data or information that can change during the
	or information that can change during the
	runtime of an application. Importance
	of State Management is as follows -
	i) UI Responsiveness:
	Efficient state management ensures that
	the user interface responds quickly to mer

	interactions. By updating only the necessary parts of the UI in response to state changes
1	flutter application can maintain smooth and
	responsive user experience.
	i) Dynamic User Interfaces:
	many applications require dynamic and
	interactive user interfaces. State management
	enables developers to update the UI dynamically
6	based on user Input, changes in data, or
	other events.
	III) code organization:
\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \	Proper state management contributes to
	a weil-organised eodebase. separating
	the UI presentation logic from the burness
	logicand state management helps in
	maintaining clean and modular code.
	iv) Form Handling and validation:
<u> </u>	State management nelps in handling user
	Input, validating data and providing feedback
	to usen in realitime.
,)	Compare and contrast the different state
b)	management in approaches avoilable in
	Flutter, Such as setstate, Provider and
	Riverpod. Provide Scenarios Where each approach
	is suitable.
	15 Sullable.
	i) For sotstate() Approach.
1 -	i) For setState () Approach: Usage - O Ideal for small to medium-sized
	application.
	applicario
aT	H Aleman e e e

	@ quick prototyping-
	Pros -
	(1) simplicity: Fary to learn and understand.
	Desimplicity: Easy to learn and understand. Desimplicity: Easy to learn and understand. Desimplicity: Easy to learn and understand.
	Com -
	1) Limited to the widget's subtree.
	(2) can read to 's pagnerri cade' when used
	extensively.
_	
	i) 'Provider' Approach
_	Usage -
_	O suitable for medium-sized to large-sized
_	applications.
_	2 Well-suited for dependency injection
_	and global state management:
_	Pros -
_	D light weight: Fary to setup ② Grood for managing app-level state.
1	(2) Good for managing app-level state.
_	3 Strong integration with Flutter -
_	widgen.
-	Cons-
_	1) May become complex 2) limited features.
_	C) TIMITED TEATURES
	iii) 'Riverpod' Approach
	Usage -
_	O Suitable For medium to large sized apps.
	@ Designed to overcome some limitations
	of Provider.

CONTRACTOR OF THE PERSON

	A SAME AND
	Pros -
	@ Built on top of 'Provider', providing additional
	() saire en rop : I revider, provising adarmond
	features and improvements.
	1 Often improved testability and scalability.
	Com:
	Relatively new compared to other solution.
@	Scenarios for each approach:
	i) Set state ()
	@ Building a simple counterapp
	1 Prototyping a basic UI with minimal
	State.
	ii) Provider'
MORROVA DAMES FOR STANSON	1) Managing user authentication state
	globally
	Q Depart dans de la langua for services and mandidade
	Dependency injection for services and repositories
•	MI) Riverpod
	De large e-commerce applications.
To the second se	@ Application, with complex dat models.
Q.4.a)	Explain The process of integrating firebase
	with a Flutter application.
= 1	Discuss me benefits of using Firebase as
	backend solution.
	BUCKERY SOLUTION
	Tirel and the state of the stat
-	Integrating Firebase with a Flutter applicable
	involves a series of steps to set up
The state of the s	Firebase in a project configure it
U	II and the second secon

	and leverage its features. Following are The steps— i) Create a Firebase Project () Go to the console.	H .
	i) Create a Firebase Project	H.
11	i) Create a Firebase Project	
	V	
	(3) Click am " add Protect".	W
	3 once project is created you will be redirected	1
	to dash board.	
	ii) Add app to Firebase.	
	O click on "Add app" and select the	
	appropriate platform.	
	2) Follow setup instructions.	
	3) Download and add configfiles.	
	III) Configure dependencies	
	iv) Initialize Firebase in the app.	
	Benefits of Using Firebase as Backend Solution	
	i) Real-time Database:	
	It offers real-time NosqL'database,	
	Cloud Firestore, which Synchronizes data	
	in real-time across clients.	
	ii) Authentication:	
	It provides secure authentication methods,	
	supporting email/password, social media	
	logins, phone authentication and more.	
	ili) Easy integration with Flutter	
	It has official plugins and packages	
_	For Flutter, making Integration straightforward	- 4
		N.
F		
		1

Highlight The Firebase Services commonly used in Flutter development and provide a brief overview of how data synchronisati is ochieved. Following are The Firebale services commonly used in Flytter developmenti) Firebase Authentication Provides secure authentication methods, including email/password, social media logins, and phone authentication. ii) (loud Firestore: A Nosqu, cloud-hosted database for Storing and syncing data in real-time. collection and documents, making it easy to query and synchronize date across de vices. iii) Firebase Real-time database: A Nosqu database that stores and syncs data in real time. It uses a TSON-like structure, and changes made to the database are immediately reflected across all connected devices. Data Syncronization: 7 Firebasels real-time database synchronizes data in following waysi) websockers: Websockers used to

maintain a persistent connection between the client and server. 11) Listener pattern: Clients subscribe to specific data, paths, and when changes occur, me gerrer punes updates to subscribed clients in real-time.