0	Name + Vishal . R. Gori
	Class: - D15B - 18
	Page
4 34 2	MAD ASSIGNMENT -1.
achbed .	c) Discuss boo the Flatter formercase different former
01	TE 12 COUNTY COUNTY ON A H MA LOW AS A MARCHAN
27	D. Explain the key features and advantages of using
1	Flutter for mobile app development.
> ?	Flutter is a loss on a series of its
1	development by a love and open-source mobile application
22	development framework created by google. It is used for developing high performance, visually
	attractive and responsive apps for IDS, Android and
	web Platform.
	Flutter is based on the Part programming and uses the
07.5	skia graphics library to render its components
	Key Features & Advantages:
1)	Hot reload: It makes it possible to see the changes in
	the code instantly set seflected on the UI, which
	speeds up the process to work on the outlook of
00	the application.
19 4)	Cross-platform developments: Flutter enables developers to write code that works on different platform. Which
	means, two different applications on two discines can
	use the same codebase.
	Ea - Lamoura & Lahone house different OS but the same and
with 3	Eg: Samsung & Iphone have different OS but the same app will work on & both devices.
3)	Widget based Architecture: UI components in fuller are
	widgets making the development modular & customizable.
4)	Open Source = Flutter is an Open-Source Plattoom, is tree
	of cost & has detailed documentation & communities available
	online.

Teacher's Sign.:

Q.2) a) Describe the concept of the widget tree in flutter Explain how widget composition is used to build complex user interfaces.

> 1) In flutter a widget tree is a hierchial structure that represents the user interface of an application It consists of

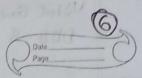
(B)	Vistal Good
	DISB-18
Stun	invidiual elements called widgets, which are the building blocks of UT.
2)	The widget tope describes the composition and organization of
	The widget tope describes the composition and organization of these widgets, defining how they nest and interact with each other.
2	each other?
3)	Widgets are objects in Flutter that define the structural & visual elements of the UI. Widgets are either stateless widgets
'\	* visual elements of the UI. Widgets are either
()	
	Stoteful widgets.
4)	The widget tree has a hierarchial structure similar to
	DOM in web development. At the root of the tree is the
	top-level widget, typically a material App. As we go
	down the tree, widgets encapsulate & compose other
5	widgets, forming a parent-child relationship.
9/	This results in composability to crarte more complex
	So - Para wid at our and win contiliated abilities like stand
	Egt fow widget con contain multiple children like text, image I Icon widgets.
06	Stuteless widgets are immutable and don't change
	over time, while stateful nidgets can hold
	mutable state which is coursely for managing
	denumic elements in the 127.
7)	mutable state which is crucial for managing dynamic elements in the DI. Stateful widgets have lifecycle that includes methods like initstate build and dispose
	like initstate build and dispose
0.2 (b)	Provide examples of commonly used widgels & their moles
//	Provide examples of commonly used widgels I their roles in correcting a widget tree.
⇒ D	Container: It is a versatile box model that can contain other widgets, providing padding margin & decoration.
	Other widgets, providing padding margin & decoration.
	Teacher's Sign.:

- Appropriate for Simple UIs: for small to moderately

manage state in Flutter. It is built into the framework and is easy to understand for beginners.

- Simplicity: 'Set State' is the most straight forward way to

- Learning Currie



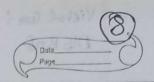
complex UIs where the state changes are localized and the widget tree is not deeply nested Setstate' - Limited to the Widget Tree: 'SetState' is limited to the widget where called and its descendants. - Over-rebuilding Widgets: It triggers a rebuild of the entire widget and its subtree potentially se causing performance issues for larger applications. Suitable Scenarios! - Small to moderately sized applications - Simple UIs with limited interactivity. Learning and prototyping purposes. 2. Provider: - Scoped State Management:

Provider allows for scoped and localized State management
reducing the need for prop drilling: - Eusy Integration: It is easy to integrate into Flutter applications and offers a good balance between simplicity and flexibility. - Large Community Support:
Provider is widely used and has good community support

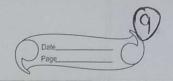
1	Vishal Growi
	015 8 78
	Date
-	- Colat of Sugar & To come over als I all the
1	- Grobal Scope: In some cases global state might be unintentionally crealed.
	Chearen de Chearen
	Suitable Scenarios:
1	- Applications of varying sizes with moderate to complex U - Situations where a controlized state management
	Solution is needed but without the complexity of
	other solutions.
	The special policy of the Freehold way of
	Kiverpod:
1	Pros
1	- Scoped and Flexible:
	- Provider Inhesitance:
	- Immutable and Reactive
-	Const
3	Learning Curve: Similar to Provider; Riverpod'.
	- Advanced teatures: Some of the endvanced features
	May not be necessary for simples applications
	- Advanced Features: Some of the advanced features may not be necessary for Simplex applications, adding unnecessary complexity.
	Suitable Scenarios:
	- Lorge and complex applications.
	- Situation where a more sophisticated, scalable and reactive state management solution is required

- Projects where dependency injection is a chitial consideration.

Teacher's Sign.:



94) a	Explain the process of integrating Firebose with a flutter
	Explain the process of integrating Firebose with a flutter application. Discuss the benefits of using livebose as
	a backend solution.
	Suitable Scenarios:
41.	Coeale a Fischose Project
	- 610 to the Firebuse Console and create a new project
	- 600 to the Firebuse Console and create a new project- - Follow the setup instructions.
2.	Add Finobes o to flutter project
	- In your flutter project, add the Firebuse SDK dependen
	- gies to the '-yami' file.
3.	- In your flutter project, add the Firebuse SDK dependen - cies to the '-yamı' file. Initialize firebuse
	- Import the Firebose packages and initialize Firebuse in
	Import the Firebose packages and initialize Firebuse in the main dart file
	Configure Firebase Services:
	Depending on the services you want to use (authentication
	firestord, etc), configure them by following the specific
	setup instructions provided by firebose.
2.	Use Fixeboxe sorvices in the App:
	- Implement firebose services in your app code.
	Bonolise of Malu Finchason
	Benefits of Using Firebose. 1. Real-time Database
	2 Authentication
	3. Cloud Functions
	4. Cloud Firestore
	5. Firebuse Storage
	6. Hosting and Analytics
	7. Authenticution State Management
	8. Secure and Scalable
	9. Fasy Setup and Integration
	Teacher's Sign.:



- b) Highlight the firebose services Commonly used in Flutter development and provide a priet overview of how duta Synchronization is achieved.
- > Common Fixebase Services in Flutter Development are: 1 Authentiaution: Firebose Authentication for user sign-in
 - Pirestore: A Nosal dutabase for seal-time dute
- Synchronization

 Synchronization

 Firebase cloud Messaging (FCM): Push notifications

 Los engaging Users.

 - * Data Synchronization

 1. Listeness and streams: Firebose services use listeners and Stocams extensively. Flutter developers can use Stream-based APIs to listen for changes in data whether its in Firestore. The Realtime Database or other Firebase Services.
- 2. Reactively Updating UI: Flutters 'Stream Builder' widget is commonly used to realtively update UI componence bused on the changes in data streams, when data changes on the server, the stream emits new data, trigeoring a sebuild of the associated UZ.
 - 3. Offline Support: Firebose sorvices provide Built in offline Support. Flutter apps can work seamlessly offline an when connectivity is restoral, changes made offline are custometially synchronized with the server.