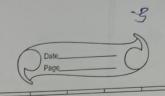
Teacher's Sign.:

	Page
	MAD Assignment I
9.1>	a) Explain the key features and disadvantages of using flutter for mobile app development.
	using flutter for mobile app developement.
7	3,4110,1314111
(flutter is a cross platform UI toolkit developed by
1	godie prompiles applications
1-2-2-1	Top mobile, wer and desktop from single codebon
	tey teature and advantages include:
9	- 1101 reload. thatber developers to instant
*	view charges without restarting.
	Widget based architechture:
	VI-components in Hutter are widgets
	modular and customizably
	3. Expressive UI: flutter provides a rich
	appealling interfaces.
	appealling interfaces.
001	every seducina de value ope once, develore
	everywhere, reducing development time and effort:
0)	dic. 1 11 months & minus bring of prof.
nh in	5. Ctrong community of months of
	5. Strong community support: A large and active community contributes to a wealth of
-01-	Community commounts to a wealth of
10	resources and packages.
5-1	the safe was a comment of provided the
3)	Discuss hour the Flutter Control
	Discuss how the flutter framework differs from
apt o	popularity in the developes it is gained
	popularity in the developer community.
1341	of Nidgette, each serving a greatly referred
	A THE REPORT OF THE PERSON OF

traditional approaches are typically imperative.

- 2. Flutters offen a consistent est across platforms ensuring a native look and feel.
- 3. The use of dart language and widget based approach entrances developer of productivity.
- 4. popularity arises from efficient developement process, performance, and me vibrant community!
- in flutteer. Explain how asidget composition.
 - concept that represents the hierarchy of user interfaces elements in an application. Every fin flutter is a widget, whether ity a button, text, image or even the entire application itself.
 - 2. The widget tree is composed of various type of widgets, each serving a specific purpose widgets in futter can be broadly categorized into two: stateless and statelfur.

Teacher's Sign.:



3. steeteless widgets are immutable and don't have any internal state, while stateful widgets can change their internal steete during their lifetime.

by provide examples of commonly used widgets.

1. Material app: defines the structure of a futter app.

2. Jeaffold: Represents the basic visitual interface of the app, including the app bar and body.

8. Container: A box model that can contain.
other widget, providing layout and styling

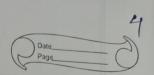
4. Raev and column. Arrange child and widge horizontally or verticelly,

5. hist view: Displays a scrolling list of widgets.
6. Hoating action button: represents a flooling action button.

9.3) a) Discuss the importance of state management in flutter application

> state management is a crucial cospect of building robust and efficient flutter application. In I flutter "state" refers to doubt finat infrance

Teacher's Sign.: ___



appearances and behaviour of widgets. Managing state effectively is essential for creating of responsive, dynamic and scalable applicat--ions. Here are some key reasons con state management is important in flutter. 1. Vser Interferce apolates 2. performance optimization

2. code managability.

4. Recepbility and and modularity.
5. persistence and navigation.

6. étateful widget navigation

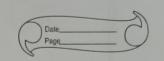
opproaches in flutter, such as setstate provider, and riverpod, provide sienarios Juhere each copproach is suiterble.

This is the most streight forward way to manage state in flutter. It is built into the fromework and is early to understand for beginners

- Appropriate for simple 019s:

& for small and moderately complex UI" where state changes are tocalized and widget free and the widget tree is not deeply nested is it state am be sufficient. cons!

set state is limited to the descendant



over rebuilding widgets: It triggers rebuild of an entire widget and its subtree, potent-ially, causing performance issues for large applications.

Suitable scenarios:

- small moderately sized applications simple. UIII with limited interactivity
- 2. Provider: Pros:

- scoped state-management:

provider allows localized and scoped
statemanagement reducing the need for
prop drilling

- Easy integration: It is easy to integrate into flutter applications and offers a good balance between simplicity end flexibility

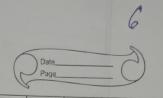
- Large community support:

provider is widely used and has a
good community support.

- Learning curve:

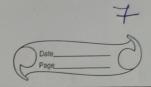
- Global scopes: In some cases global state might be unintentionally created.

Teacher's Sign.:



Teacher's Sign.: _

51 , 230	3 witch he along the
and the	- Application :
	-ate to of verying sizes with moder-
	- Applications of verying sizes with moder-
	3. Riverpod!
	Pros!
1. 11.	- scoped and flexible,
474124	- Provider inheritemoe.
	- Provider inheritemore. - Immuatable and reactive.
	Cons:
	- hearning course: takes time to learn things initially. - some advanced features may not require for simpler applications.
	things initially
	- some advanced fratumes
	require for simpler conditions
	pricetion.
9.4>	a) Explain the process of the
0 + 5 1	a) Explain the process of integrating firebon with flutter applications. 1. create a firebone project: - Go to firebone console and weater men project
>	1. Create a finalization
4414-4	- GO to timber a consistent
	men project consoje and weaten
	To the state of th
	- follow the setup instruction.
	All the state of t
	2. Add firebone to flutter application In your flutter project, add firebone SPK dependencies to the 'yaml filesh.
	In your flutter projet, add firebase
	SPK dependencies to the 'ayaml tites.
-	3. Intialize firebare
	- Import the firebone packages and initialize in the main dourt' five.
	In the main dourt' five.



Teacher's Sign.: _

configure finebone service;

- Depending on the service you want to

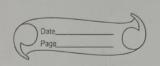
use (authentication finestore) configure them

by following specific setup instructions provided by firebond vie firetoire services in app.

rimplement firetoire service in your app

code. Bonefit of using firebone: I. Real-time database. 2. Authentication. y. cloud function 5. Pinebane Horago. 6. hostly and analytics. F. Authentication state management. 8. secure and stateble. 9. Easy setup and integration. by Highlight the services commonly used in flutter development (firebone) and provide brief overview, of how data symchronization is achieved. 1. Authentication: Firebone authentication for user osign-In.

2. Fire store: A NO-SGL databane for red time dates synchronization.



3. firebone cloud meaaging! Push notifi--cations for engaging users!

* Dated synchronization!

1. Listeners and Arcams!

Apl's for to listen for changes in date whether it in firetone, the realtime date bane or other firestone rervices

2. Reactively updating UIS: flutters

stream builder' evidget is commonly used to reactively update UI componency based on the changes in data streams.

When data changes on server, the stream emits the data, thiggering cerebuild of the associated UI.

3. Offline support:

Firebone services provide built in offline support. Flutter app can work Geownlessly offline and when connectivot is restored, changes made offline are automatically synchronized with senser.