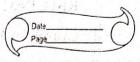
	Name: - Amit - B. Nayak
	Roll No:-46
	Claso: - DISB  MAD Assignment -1
	1 1351ghmevii
(1.0	Explain the Bo. Intra
	DExplain the Rey Jeatures and advantages of using flutter
$\rightarrow$	1 worth abo acretobush
	i) (roos-Plathorm Perelopment:- Flutter allows developers to
	write a single platform (codebase) to create apps. For
	both Android and Ios platforms. This significantly
es de	reduces development time and effort.
	ii) that Reloast: This beature enables developers to see
	the changes made in the cycle almost is instant in
	the app. It increases productively and allows for
	facter iterations
	iii) Rich set of Widgeto: - Flutter provides a comprehensive
	set of pre-designed widgets that bollow specific
	design language like Material Design (600glp) and
	Capartino (Apple)
Nest	iv) part language! - Flutter voes Part, a language by
	boggle, which is easy to learn and obers advanced
	features like just-in-time compilation and ahead-of-
	time compilation.
	in mind again to the many states of gras
(12	is a boundary menther a south him electric returns
(P)	Discuss how the Flutter framework differ from traditional
	approaches and why it has sained popularity in the
	developer community.
->	i) Single Codebose for Multiple Platforms: - Traditional
	approaches often require seperate codebase for
	different platforms. Flutter eliminates this need.
	ii) widget - Contrix Devign: - Unlike traditional approached where

UI components might depend on the platform, Flutter's UI is built using a rich set of customizable widgets.

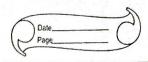
enouring a consistent dook a cross platforms.

Teacher's Sign.:



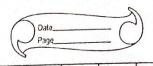
iv) Versability:- Flutter's ability to run an multiple platforms beyond just mobile (like web and deaktop) makes it a versatile choice for full-stack development Q'2 a) Describe the concept of the widget tree in Flutter Explain how widget composition is used to build complex user interfaces. i) widget Tree: - In Flutter, the UZ is built using widgets, which are the book building blocks of a flutter app. The widget tree represents the hierarchical arrangement of these widgets. a It's a structure in which widgets are nested within ii) widgeto: - widgeto in Flutter can be thought of as elemento of UI, ranging from a simple text field to a complexe animation. There are two types of widgets. Stateless and Stateful. Stateless inidgets don't change over time , while stateful widgets can update their state. iii) widget\_ (omposition:- Flutter uses a composition over inheritance principle. This means you build complete UIs by composity simple widgets complete widgets are broken down into smalley simpler widgeto and these are then combined to create the desired uz. iv) (ustom widgeto: levelopers con create watom widgets by combining several simpler widgets. This moduler approach enhances reveability and simplifies the maintenance of the codebase:

Teacher's Sign.:



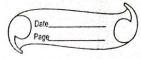
Teacher's Sign.: \_

Q.5.(p	Provide examples of commonly used widgets and there their
	rules in creating a widgel tree.
>	i) Scathold: - Acts as the book structure for moterial design
	apps. It provides a structure to odd opphars, drawers,
	smack bars, and bottom navigation bors.
	ii) Container: - A multi-purpose widgets wood for styling, padding,
-	margino, border, and positioning. It can hold a single child
	undget.
	ii) List View: - A scrollable list widget. It's used to display a
0	list of items when the total number of items is not
	known beforehand or is too large.
4	iv) stack: - Allowo for overlying widgets on top of each other.
	It's weeful for creating overlapping elements.
	V) Icon: - Displays icon from a predefined set or custom
	icons often words used in buttons or app bars
	death and promite glass with a solution of a particular
6.3.(9)	Discuss the importance of state management in Flutter
	applications.
7	State management is a critial aspects of building robust and
8	official Flutter applications. In Flutter, appearance and behaviour
	of widgets. Managing state effectively is essential for
	creating responsive, dynamic and scalable applications.
	Managing state effectively is essential for creating responsive,
	dynamic and scalable environment applications Here are some
	key reasons arraby state management is important in
	Flutter.
	1. Voer Interface Updates.
	2. Performance Optimization.
	3. Code Maintainability
	4. Revsability and Modulariti

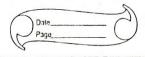


Teacher's Sign.: \_

	And the second s
3.11	5: reroistance and Novigation.
	6. Stateful Widget Limitation
	7. Concurrency and Agenchronous Operations
	see a company of the second of
(b)	Compare and controot the different state management
	approached available in Flytter, such as set state,
	Provider, and Riverpod. Provide scenanior where each
	approach ie suitable.
	the bridge of the hoolest fill about here by section of the latter of
7	1. Set State:
	Profit out of the formation of the state of
alle d	a) Simplicity: - 'set state' is the most straightforward way to
	manage state in Flutter. It is built into the framework
	and is easy to understand. for beginners
	b) Appropriate for simple vz:- For small to moderately
	complex us, where the state changes are localizard
	and the widget tree is not deeply nexted, state?
	(an be sufficient.
1 - 2 - 2	Consistibilità de atragas humo a a transporsa dell'ist
	(a) Limited to the widget Tree: - 'set state' is dimited to
	the widget where it is called and its descendants.
	(b) Over-rebuilding widgets:- It triggers a rebuild of the
	entire widget and its subtree potentially, causing
	performance issues for larger applications.
	Suitable Sconoviosi-
	* suitable scenarios:-
	- Small to moderately sized applications
	- Simple UIs with limited interactivity.
	- Learning and prototyping purposes.
	네 [2017년 14일 전 14일 전 14일 대한 14일 전 14

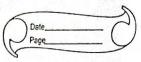


	Page
2	Provider:
	luo:-
	Scoped state Management:
	comoraer allows for scoped and localized state
	The state of the s
	and flexibility.
	(c) Large Community Support: Provider is widely used and has
	good community support.
	Constant of Dealers of Dealers of the
	(one:-
	(b) (detal constraints (unverse
	to order scope - In some coner along state mint
	unintentionally created:
	* Scitched C
	* Suitable Scenarios:-
	(a) Applications of versons vorying sizes with moderate to
9	as in pierce of
	(b) Situations where a contralized state management solution
4	is needed but without the complexity of other solutions.
3.	Riverpod:
	froe-
1	(a) Scoped and Flexible
The state of the s	(b) Provider Inheritance
	CI Immutable and reactive.
	Confis-
	Col Learning Curre: - Similar to 'Provider', "Piverpod"
	b) Advanced Features: Some of the advanced beatures many not be
	necessary for simpler applications, adding unnecessors complexity
	Teacher's Sign.:



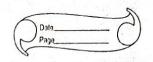
Teacher's Sign.: \_

	* Suitable Scenonios:
	61 Lorge and complex applications.
	b) situations where a more a suphisticated, scalable, and
	reactive state monogement solution is required.
	corrected where dependency injection in a crucial.
	Consideration.
	and the state of t
Q · to al	Explain the process of integrating birebose with a Flutter
	apprication places the benefits of using Firebone as a
	SOLVITON.
$\rightarrow$	1. Create a Firebuse project:-
	(0) to to the Firebase console and create a new provent
	(b) Follow the setup instructions.
	They had been now and read to the the
	2. Add Fireboxe to Flutter projects -
	(a) In your flutter project, add the firebase SPK
	dependencies to the 'yourd' bile.
	I Standbon Alice ous is paracon notice for multipastor
	3. Initialize Firebose:
	(a) Import the fireboxe packages and imitialize Freboxe
	in the 'main dort! file it modified the hology
	4' Configure Firebase Services:-
	a) Depending on the services you want to use (authentication,
	birestore, etc.), configure them by following the specific
	setup in structions provided by Firebook.
	contract two start and the
	5. Use Firebase Services in the App:
	cal Implement Firebase services in your apper code.
144	Conservation of the constant of the same o
	And the second of the second o



Teacher's Sign.: \_\_

	Benchita at wine 1)
	Benefito of voing firebuse:-
	r Real-time Patabase.
	2. Authentication
	3. Cloud Functions.
	Le Cloud Firestore
	5. Firebose Storage.
	6. Hosting and Analytics
	Mulhertication State Management.
	secure and scalable,
	9. Eagy setup and Integration.
0.4.6	Highlight the Firebose services commonly used in Flutter
	development and provide a brief overview of how data
	synchronization is achieved.
->	Common Firebase Senices in Flutter development are:
	1. Authentication: - Firebooe Authentication for user sign in: -
	2. Firestore: A Nosol detalesse 1 - 001 11
	2. Firestore: - A Nosol database for real-time data synchronization
	3. Firebase (loud Messing (FIM):- Push Notifications
	for monaging users.
	A) Data Synchronization:
	1. Listeners and streams: - Firebose services use listeners
	and streams extensively. Flutter development can use
	etrosm - based Abits to 1500 1 5
	stream-bosed APIs to listen for changes in data,
	whether it's in Firestore, the feattime Patabase or
	other Firebook services.
	H - BROND HE HONDER A. H.



- 2. Reactively updating US: Flutter's 'stream Builder widget's commonly word to reactively update the the UZ components bookd on the changes in data streams. When data changes on the server, the stream emits to new data, trigening a rebuild of the associated UZ.
- 3. Offline Support: Firebose services like provider built-in offline support. Flutter apple can work seamlessly offline and when connectivity is reshared, changes made offline are automatically synchronized with the server

Teacher's Sign.: