03/05

Mohit. Aileni

Assignment -2

MPL

(2)

Define Progressive webApp and explain its significance in Modern web development Discuss the key characteristics that differentiate PWP, From traditional mobile apps

A progressive web app is a web application that delivers a native applike experience using Modern web technologies. It offers offline access, push notification and fast performance while being installable From a browser without needing an appstore

1) Installation :- PWAs install via browser while Mobile apps Store downloads
2) cross platforms :- PWA's work on all devices whereas Mobile apps need Seperate devllopment For android and IOS
3) storage :- PWA's take up less space compared to large Mobile apps

2. Define responsive web design and explain its importance in the context of progress web apps

→ Responsive web design

→ ensures web pages adapt dynamically to different screen sizes using flexible grids media queries and CSS techniques

Importance in PWAs

1) ensures seamless experience on all devices

2) Improves accessability, performance and SEO

3) Enhances user engagement with layout

Comparison

1) Responsive:- uses flexible grids and media queries for all screens

2) Fluid:- uses percentage based widths for smooth resizing

Describe the lifecycle of service, including registration, installation and activation phases.

Registration :- web page registers the service worker using navigator.serviceWorker.register()

Installation :- caches static assets triggered by install event

Activation
clears old caches and prepare for control, triggered by activate event

Termination and update
Auto terminates when idle when a new version is registered.

Qu. explain the use of IndexedDB in the servi worker for data storage.

=) • IndexedDB in service workers.

IndexedDB is used in service workers for offline data storage beyond static assets, enabling persist structured storage for API responses, user data and form inputs.

• why use it?

1) Storage dynamic data offline for PWAs.
2) asynchronous & efficient, preventing UI blocking
3) persistent storage, even after closing the brow

• example -

1) open /create a database:-

```
const request = indexedDB.open("mydatabase", 1)
request.upgradeneeded = (event) => {
event.target.result.createObjectStore("datastore",
                    { keypath: "id" });
};
```

IndexedDB in service workers ensures reliable offl access by caching structured data, improving perf performance and user experience.

=X=

MAD & PWA Lab

Journal

Experiment No.	Assignment-1
Assignment 1 Questions	1. a) Explain the key features and advantages of using Flutter for mobile app development. b)Discuss how the Flutter framework differs from traditional approaches and why it has gained popularity in the developer community. 2. a)Describe the concept of the widget tree in Flutter. Explain how widget composition is used to build complex user interfaces. b) Provide examples of commonly used widgets and their roles in creating a widget tree. 3. a) Discuss the importance of state management in Flutter applications. b) Compare and contrast the different state management approaches available in Flutter, such as setState, Provider, and Riverpod. Provide scenarios where each approach is suitable. 4.a) Firebase Integration in Flutter: Explain the process of integrating Firebase with a Flutter application. b)Discuss the benefits of using Firebase as a backend solution. Highlight the Firebase services commonly used in Flutter development and provide a brief overview of how data synchronization is achieved.
Roll No.	2
Name	Mohit Ailani
Class	D15B
Subject	MAD & PWA Lab
Lab Outcome	LO1: Understand cross platform mobile application development using Flutter framework LO2: Design and Develop interactive Flutter App by using widgets, layouts, gestures and animation LO3: Analyze and Build production ready Flutter App by incorporating backend services and deploying on Android / iOS
Grade:	

Roll No: (2)

Mohit Alani

DISB DATE: 03

Assignment No: 1

Explain the key features and advantages of using Flutter for mobile app development.

Flutter is a popular open-source UI toolkit developed by Google for building natively compiled applications for mobile (iOS & Android) web and desktop from a single codebase.

Key features of Flutter:

1. Single codebase: write once, run on multiple platform (iOS, Android web, desktop).
2. Dart programming Laungugc: Uses Dart, which is optimized for fast performance and ahead-of-time (AOT) compilation
3. HOT Reload: Instantly reflects changes in the app without restarting, making development faster and more efficient
4. Rich widget Library: Provides a vast collection of customizable widgets that support Material Design and cupertino styles for a native look and feel.

Advantages of using Flutter:

1. Faster Development Time: Hot reload and a single codebase reduce development
2. Cost Effective: since developers write one codebase for multiple platforms, it reduces costs associated with maintaining separate teams for iOS & Android.
3. consistent UI: Flutter renders everything using its own engine, ensuring a uniform look across.

DATE:

Q1 b) Dicuss how flutter framwork differs from traditional approaches & why it has gained popularity in the dev community?

Ans Flutter uses a single codbase for multiple platforms unlike traditional native development that requires platform specific UI components but instead renders using its own skia graphic engine, ensuring consistency. Unlike React native, which uses a JavaScript bridge, Flutter compiles directly to native ARM code, offering better performance. Its hot reload feature allows devs to see changes instantly, making development faster more efficient.

Flutter has gained popularity due to its faster development cost efficiency & cross platform support. Devs prefer it as reduces development time & costs while delivering high performance apps. Its customizable widget system ensures a smooth, native like experience.

Q2 a) Describe the concept of the widget tree in flutter & the widget composition is used to build complex UI.

Ans. In flutter, everything is a widget (button, text, lay out). These widgets are arranged in a hierarchical structure as the widget tree. The widget tree determines the UI. widget composition to build complex UI:

Flutter encourages a composition-based aproach rather than inheritance

Instead of creating large monolithic widget, developers build small, reusable widget that are combined to complex UIs.

DATE:

ex: A column widget can hold multiple Text & Button widget creating a structured layout

Provide ex. of commonly used widgets & their roles in creating a widget tree

structural widgetts

scaffold :- provide basic structure of a screen

container :- Used for layout styling

column & Row :- Used for vertical & Horizontal layout

Interactive widget

Textfield :- For user input

Elevated Button :- clickable buttons

styling widget

Padding :- Adds spacing around widget

Align, centre :- Adjust alignment

List & scrollable widget

Listview :- scrollable list

Gridview :- Provide, Display items

ex :- simple widget Tree

```
Scaffold(
appBar : AppBar(title : Text("flutter App")),
body : Column(
children : [
Text("welcome to flutter!")
Elevated Button Compressed : (), {}, Child : Text("Click
Me")),
],
),
),
```

DATE:

Q3 a. Discuss the importance of state management in application

Ans. Importance of state management in Flutter Application, State ~~ment~~ management refers to handling dynamic data, that changes overtime

In Flutter, The UI rebuilds when the state changes.. ensuring the app remains interactive and responsive Proper State management helps in performance

Q3 b Compare and contrast different state-Management in Flutter approches, available in Flutter, such as SetState, provider, RiverPod, provide Scenarion where each approach is suitable

Ans:- comparision of State Management Approaches in Flutter Basic State Management by Calling setState() To update UI, Small-apps, Simple UI updates toggling a switch, provider User inherited widget to efficiently manage state accross the widget tree.

2. apps requiring modular and Scalable State Management.

xplain the process of integrating Firebase with a Flutter application

ntegrating Firebase with Flutter

d its benifits

ntegration Process

up Firebase Console

reate Firebase project

egister the app for android and ios

ownload and add google services.JSON or google service - info.plist (ios)

nstall Firebase dependencies

ort

void main() async {

Firebase_core : latest_version

Firebase_auth : latest_version

cloud_Firestore : latest_version

Initialize Firebase in Flutter

Benefits

No need to manage servers.

Provide Authentication, database

etc

FOR EDUCATIONAL USE

DATE:

Q4.b highlight the Firebase service commonly used in Flutter development and brief overview of how data sync is achieved

→ Commonly used Firebase services in Flutter and data sync service Functionality :-

- Archbase Authentication uses Sign-in cloud Firestore Nasist database for real time data Syncing.

→ Data Synchronisation in Firebase :-
Firestore allows real-time data syncing using snapshot listener

```
Firebase.Firestore.instance.collect
('message', snapshots().
  listen (snapshot) {
  For (my doc in snapshotdocs) {
     print (doc.d());
  }
});
```