

Prerequisite Document

1. problem statement : Many individuals struggle with selecting outfits that suit their unique body types & personal styles. The lack of personalized fashion guidance leads to frequent wardrobe dissatisfaction.

Description of Styline App - It is a personalized fashion recommendation mobile application developed using flutter. The app aims to guide users in choosing clothing that best suits their body type by

Taking an interactive body analysis test

Identifying user's body type

recommending outfit choices based on different occasion

Existing Applications & Their drawbacks

pinterest - offers fashion inspiration through photos & boards

drawbacks - lack personalization

Smart closet - virtual wardrobe & outfit planner

drawbacks - complex UI, no body type detection

4) Our solution - Styline App

Styline bridges the gap by providing a data driven, user-centric fashion experience

- Interactive test - gathers key metrics & answers to determine body shape
- Tailored recommendations
- Simple UI/UX

Advantages of StyleMe

personalization - outfit suggestions tailored to individual body type

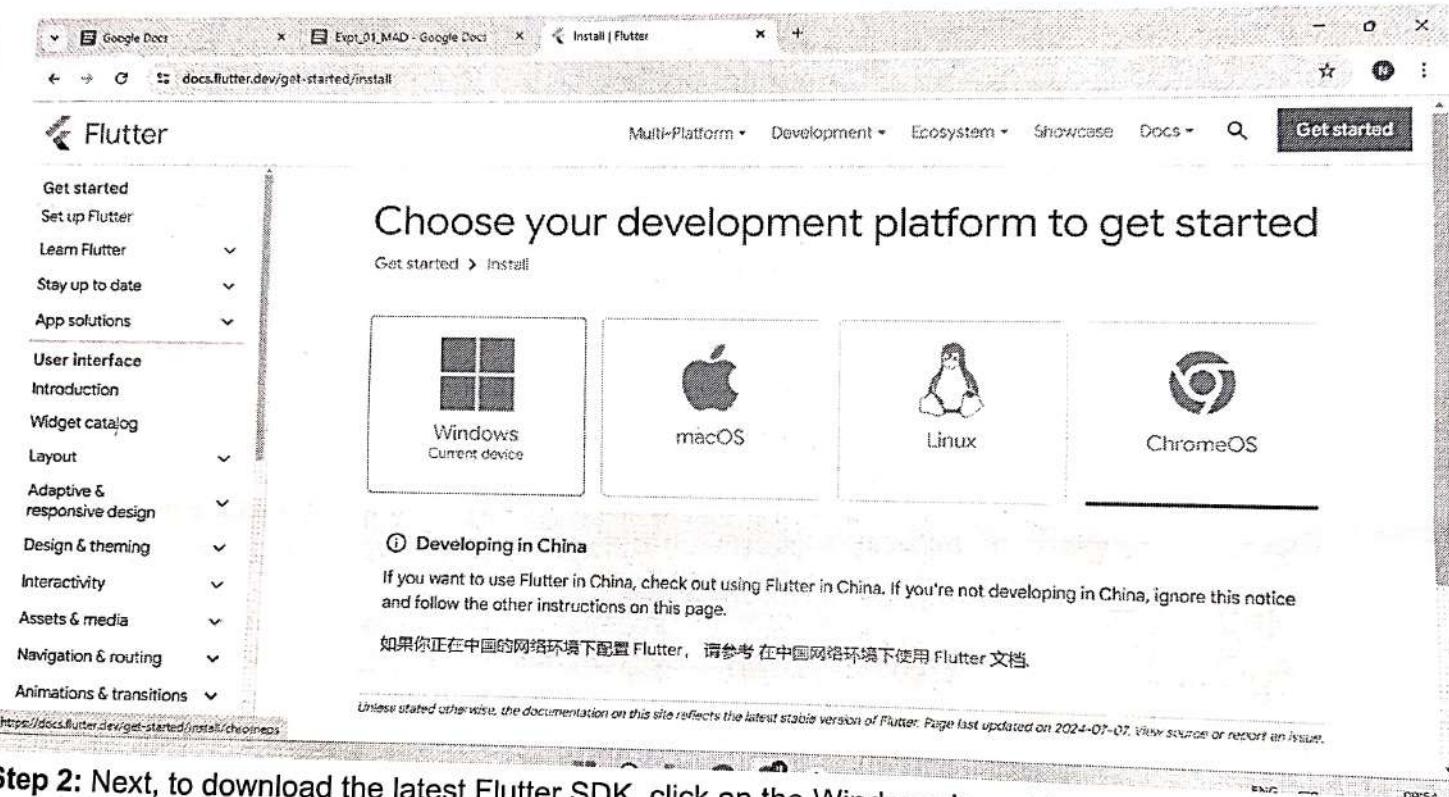
confidence boost - helps user feel confident & cool in what they wear

reduces time spent deciding what to wear or buy

EXP 1: Installation and Configuration of Flutter

Install the Flutter SDK

Step 1: Download the installation bundle of the Flutter Software Development Kit for windows. To download Flutter SDK, Go to its official website <https://docs.flutter.dev/get-started/install> , you will get the following screen.



Step 2: Next, to download the latest Flutter SDK, click on the Windows icon. Here, you will find the download link for SDK.

Step 3: When your download is complete, extract the zip file and place it in the desired installation folder or location, for example, C: /Flutter.

Step 4: To run the Flutter command in regular windows console, you need to update the system path to include the flutter bin directory. The following steps are required to do this:

Step 4.1: Go to MyComputer properties -> advanced tab -> environment variables. You will get the following screen.

Environment Variables

User variables for Student

Variable	Value
FLASK_APP	application.py
GHCUP_INSTALL_BASE_PREF...	C:\
OneDrive...	C:\Users\Student.VESIT513-15\OneDrive
Path	C:\Users\Student.VESIT513-15\AppData\Local\Programs\Python\La...
TEMP	C:\Users\Student.VESIT513-15\AppData\Local\Temp
TMP	C:\Users\Student.VESIT513-15\AppData\Local\Temp

New... Edit... Delete

System variables

Variable	Value
ComSpec	C:\WINDOWS\system32\cmd.exe
DriverData	C:\Windows\System32\Drivers\DriverData
NUMBER_OF_PROCESSORS	12
OS	Windows_NT
Path	C:\Program Files (x86)\VMware\VMware Player\bin;C:\Program Fil...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
PROCESSOR_ARCHITECTURE	AMD64

New... Edit... Delete

OK Cancel

Step 4.2: Now, select path -> click on edit. The following screen appears

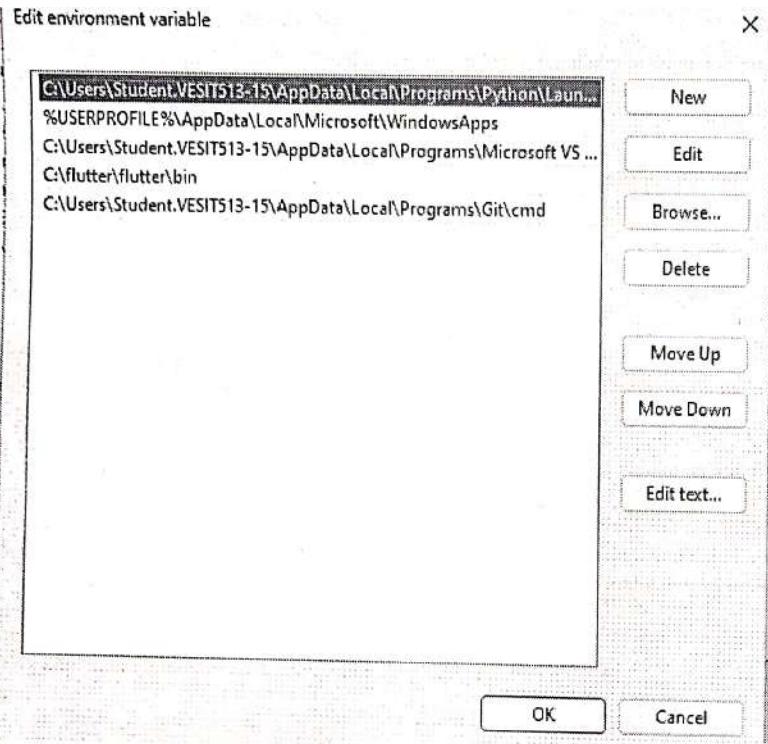
X

it environment variable

New
Edit
Browse...
Delete
Move Up
Move Down
Edit text...

OK Cancel

Step 4.3: In the above window, click on New->write path of Flutter bin folder in variable value



Step 5: Now, run the \$ flutter command in command prompt.

Now, run the \$ flutter doctor command. This command checks for all the requirements of Flutter app development and displays a report of the status of your Flutter installation.

Step 6: When you run the above command, it will analyze the system and show its report, as shown in the below image. Here, you will find the details of all missing tools, which required to run Flutter as well as the development tools that are available but not connected with the device.

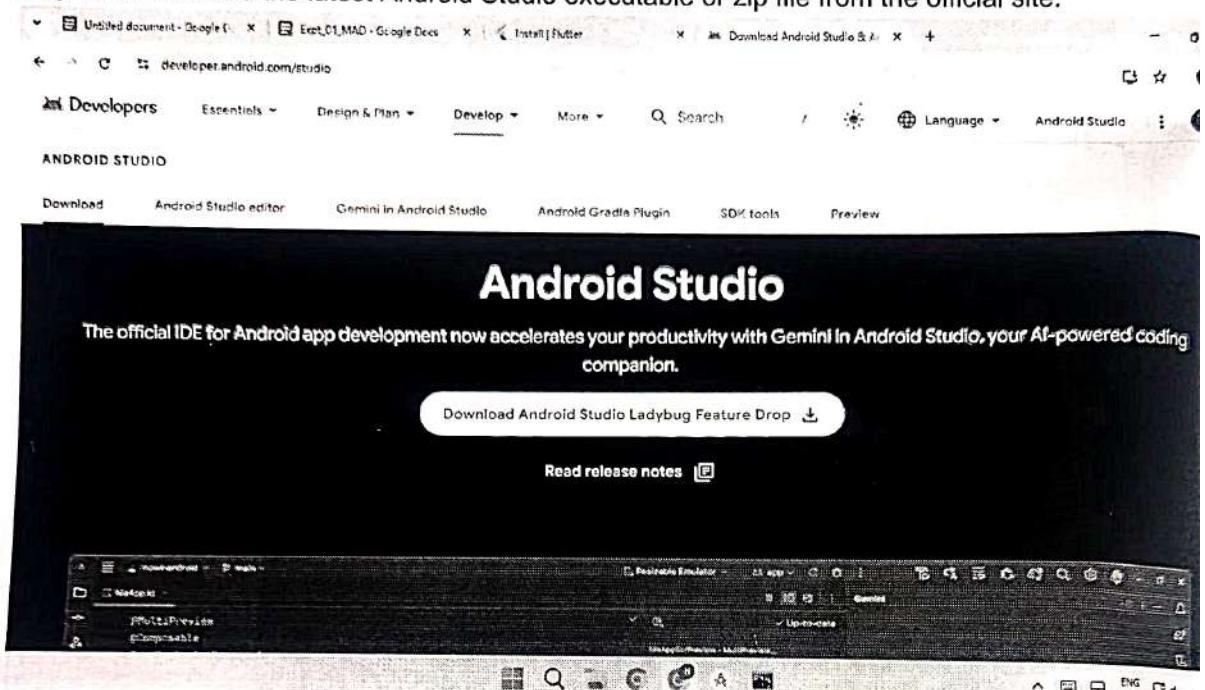
```
Command Prompt - flutter - flutter doctor
Run 'path/to/sdkmanager --install "cmdline-tools;latest"'
See https://developer.android.com/studio/command-line for more details.
X Android license status unknown.
  Run 'flutter doctor --android-licenses' to accept the SDK licenses.
  See https://flutter.dev/docs/get-started/install/windows#android-setup for more details.
Chrome - develop for the web
Visual Studio - develop Windows apps (Visual Studio Community 2022 17.3.1)
X Visual Studio is missing necessary components. Please re-run the Visual Studio installer for the "Desktop
  development with C++" workload, and include these components:
    MSVC v142 - VS 2019 C++ x64/x86 build tools
      - If there are multiple build tool versions available, install the latest
        C++ CMake tools for Windows
        Windows 10 SDK
Android Studio (version 2023.1)
VS Code, 64-bit edition (version 1.85.1)
Connected device (3 available)
Network resources

Doctor found issues in 2 categories.

C:\Users\Student.VESIT513-15>
```

tool in your system, then you need first to install the Android Studio IDE. To install Android Studio IDE, do the following steps.

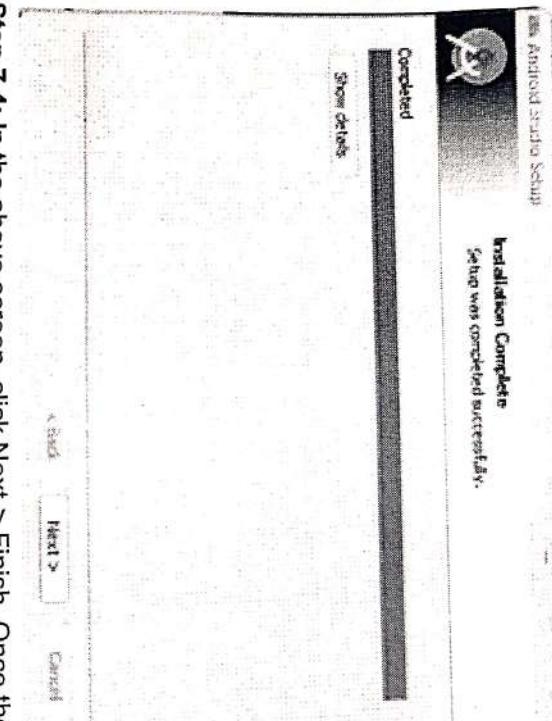
Step 7.1: Download the latest Android Studio executable or zip file from the official site.



Step 7.2: When the download is complete, open the .exe file and run it. You will get the following dialog box.



Step 7.3: Follow the steps of the installation wizard. Once the installation wizard completes, you will get the following screen.



Step 7.4: In the above screen, click Next-> Finish. Once the Finish button is clicked, you need to choose the 'Don't import Settings option' and click OK. It will start the Android Studio.

Step 7.5: run the \$ flutter doctor command and Run flutter doctor --android-licenses command.

```
C:\Users\Admin>flutter doctor --android-licenses
[Android] SDK not found. Please download it from: https://developer.android.com/studio
[Standard output] [49%] Fetch remote repository...
[Standard error] [50%] No repository configured for branch 'main' [50%]
All SDK package licenses accepted.
```

```
Warning: Errors during XML parse:
Warning: Additionally, the fallback loader failed to parse the XML.
Warning: Errors during XML parse ] 49% Fetch remote repository...
Warning: Additionally, the fallback loader failed to parse the XML...[=====] 100% Computing updates...
All SDK package licenses accepted.
```

```
C:\Users\Admin>flutter doctor
```

```
Doctor summary (to see all details, run flutter doctor -v):
[!] Flutter (Channel stable, 3.27.2, on Microsoft Windows [Version 10.0.22631.4751], locale en-IN)
[!] Windows Version (Installed version of Windows is version 10 or higher)
[!] Android toolchain - develop for Android devices (Android SDK version 35.0.1)
[!] Chrome - develop for the web
[!] Visual Studio - develop Windows apps (Visual Studio Build Tools 2019 16.11.43)
[!] Android Studio (version 2024.2)
[!] VS Code (version 1.96.4)
[!] Connected device (3 available)

[!] Network resources

• No issues found!
```

```
C:\Users\Admin>
```

Step 8.1: To set an Android emulator, go to Android Studio > Tools > Android > AVD Manager and select Create Virtual Device. Or, go to Help->Find Action->Type Emulator in the search bar.

Step 8: Next, you need to set up an Android emulator. It is responsible for running and testing the Flutter application.



System Image

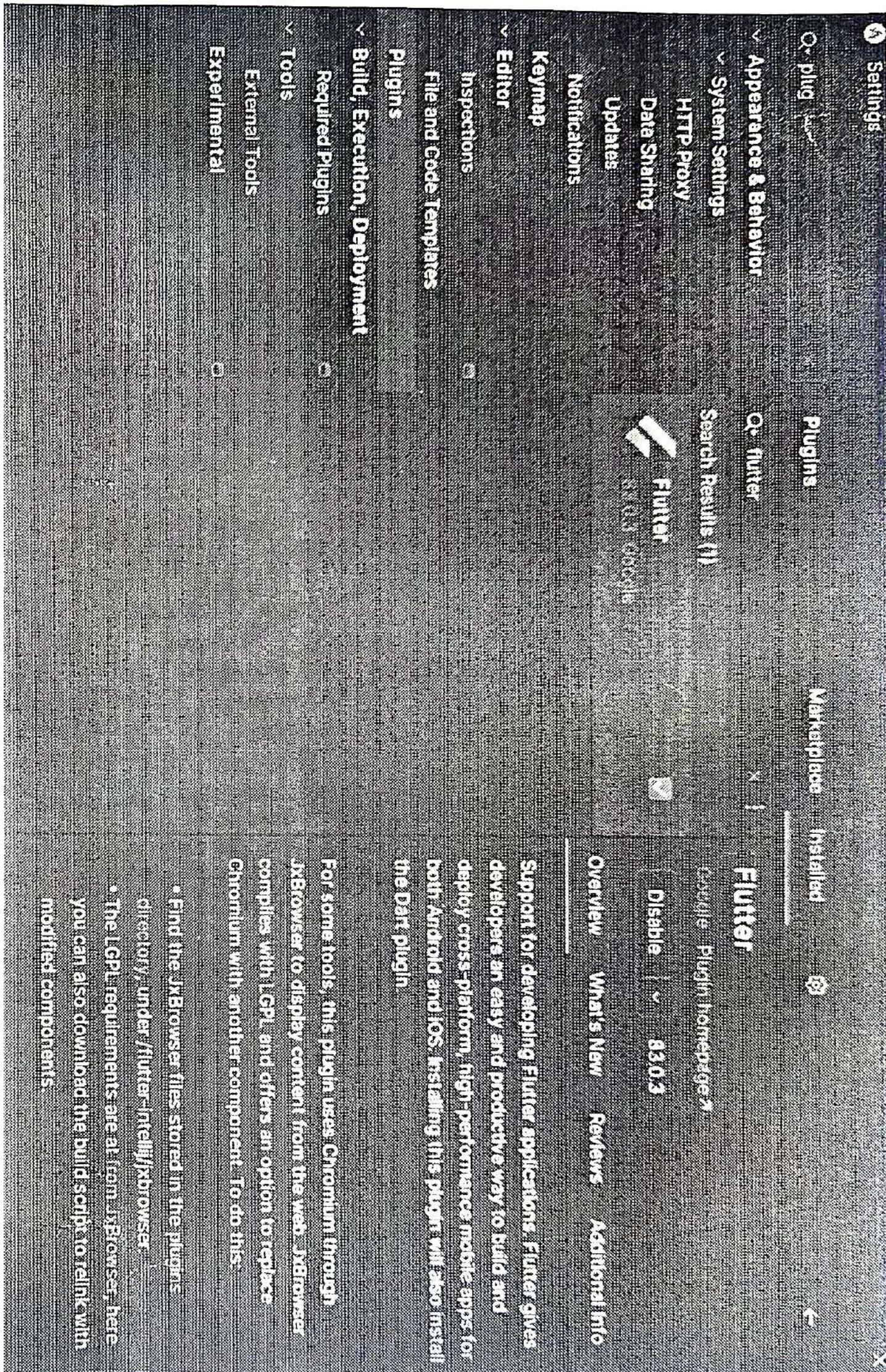


Select a system image

100% 64
We recommend these Google Play images because this
Android is manufactured with Parsons. Our

Step 8.5: Last, click on the icon pointed into the red color rectangle. The Android emulator displayed as below screen.

Search Everywhere Double Shift
Go to File Ctrl+Shift+N
Recent Files Ctrl+E
Navigation Bar Alt+F11
Drop files here to open them



D15B

17

Experiment No 2

Aim: Basic flutter UI design with common widgets. To design & implement flutter interface incorporating common widgets such as text field, button, switches, sliders & other interactive elements to understand basic flutter.

Theory:

procedure :- 1) project setup
create new project on ~~android studio~~

setup main. dart as an entry point

configure basic android app structure

~~Implementation steps:~~

- create a staticful widget for the form screen
- add the following widget
- image widget
- text field
- switch widget
- slider widget
- button widget
- check box for terms & conditions

Testing :-

Tested widget rendering & layout
verified component interactions

checked form responsiveness

Results

- properly structured widget tree
- working from elements to layout size & media
- responsive layout with proper spacing

Conclusion: In this task, experiment we acquainted with the fundamental concept of par-
-amming language & learnt to import position custom UI components using

flexible layout & responsive layout sizes
flexible width for the input, button, and
flexible gap between visual elements

Next, task with responsive

checkbox, dropdown & switch

checkbox, dropdown, list box &
checkbox, switch

list box, form

checkbox, form

checkbox, dropdown & switch

checkbox, dropdown

checkbox, dropdown

Experiment No: 02

Aim: To Design Flutter UI by Including Common Widgets

Theory:

Flutter is an open-source UI software development kit (SDK) created by Google, used to develop applications for mobile, web, and desktop from a single codebase. It provides a wide range of widgets that help in building flexible and visually appealing UIs.

Common Flutter widgets include:

1. **Scaffold** – Provides the basic structure of an app, including an **AppBar**, **FloatingActionButton**, and a **body**.
2. **Container** – A versatile widget for designing UI components with padding, margin, color, and decoration properties.
3. **Row** and **Column** – Used to arrange widgets horizontally and vertically, respectively.
4. **ListView** – A scrolling widget used to display a list of items dynamically.
5. **Text** and **TextStyle** – Used for displaying and styling text in the application.
6. **Image** – Displays images from assets, network, or memory.
7. **ElevatedButton** – A material design button that responds to user interaction.
8. **Navigator** – Helps in navigating between different screens in the application.

Code:

```
import 'package:flutter/material.dart';

class WelcomePage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.black,
      body: Padding(
        padding: const EdgeInsets.symmetric(horizontal: 20.0),
      ),
    );
  }
}
```

```
    child: Column(
      mainAxisAlignment: MainAxisAlignment.start,
      crossAxisAlignment: CrossAxisAlignment.start,
      children: [
        SizedBox(height: 40),
        Text(
          "Welcome to StyleMe!",
          style: TextStyle(
            color: Colors.white,
            fontSize: 28,
            fontWeight: FontWeight.bold,
          ),
        ),
      ],
    ),
  ),

```

```
hintStyle: TextStyle(color: Colors.blueAccent),  
enabledBorder: UnderlineInputBorder(  
borderSide: BorderSide(color: Colors.blueAccent),  
,  
focusedBorder: UnderlineInputBorder(  
borderSide: BorderSide(color: Colors.blueAccent, width: 2),  
,  
,  
),  
SizedBox(height: 20),  
Align(  
alignment: Alignment.center,  
child: ElevatedButton(  
onPressed: () {},  
style: ElevatedButton.styleFrom(  

```

Welcome to StyleMe!

Let's get introduced. Please enter your profile name - how we should call you.

Enter profile name here

Shashikumar Chintawar

D15R

G

Experiment No 3

Enhanced flutter UI with Icons, Images, & custom fonts

AIM:- To include icons, images, fonts in flutter procedure:

Setup & config

add google fonts dependency to pubspec.yaml
 configure access dependency for images
 setup necessary material with imports

Implementation steps

enhance UI components with icons

added person icon to name field
 added notification icon to switch
 added check icon to submit button

Testing

verified proper loading of custom fonts
 checked icon rendering & re-alignment
 Tested overall UI consistency

Icons- custom icons can be added using packages like flutter_launder - icons

Images- flutter supports asset images, network images. A num-

Image: flutter supports asset images, network images. A num-

Exper

e: Shrushti Chinta

No: 09

id: D15B

image
The pubspec.yaml file must be updated to image

- fonts: custom fonts can be included by adding of file and specifying them in pubspec.yaml

results:

- material icons integrated
- custom font called hake
- improved visual hierarchy
- professional looking interface element

Conclusion

Basic flutter widgets provide a solid foundation for interactive UI. Material design icons & custom fonts significantly improve widget usage, efficiently handles complex animation code. Proper planning of widget structure is crucial.

Use Image.asset
Using NetworkImage
3. Adding a font file
Flutter allow
Steps to add font
Define the

To include icons,

allows developer to include icons. These icons: provides built

ing Material Icons (Icons/home, si on(FontAwesome) . Adding Image assets:

Adding Local Images can be a

Define them in flutter.yaml

assets:

- asset

Experiment No: 03

Aim: To include icons, images, fonts in Flutter app **Theory:**

Flutter allows developers to enhance their app's UI by incorporating icons, images, and custom fonts. These elements help in improving the app's aesthetics and user experience.

1. Including Icons:

Flutter provides built-in Material Icons and supports custom icon sets.

Using Material Icons:
Icon(Icons.home, size: 30, color: Colors.blue);

Using Custom Icons (from packages like FontAwesome):
import 'package:font_awesome_flutter/font_awesome_flutter.dart';
Icon(FontAwesomeIcons.heart, size: 30, color: Colors.red);

2. Adding Images:

Images can be added from local assets or loaded from a network.

Adding Local Images:

Place images in the assets folder.

Define them in pubspec.yaml:
yaml flutter:

assets:
 - assets/images/logo.png

Use Image.asset in the app:
Image.asset('assets/images/logo.png', width: 100, height: 100);

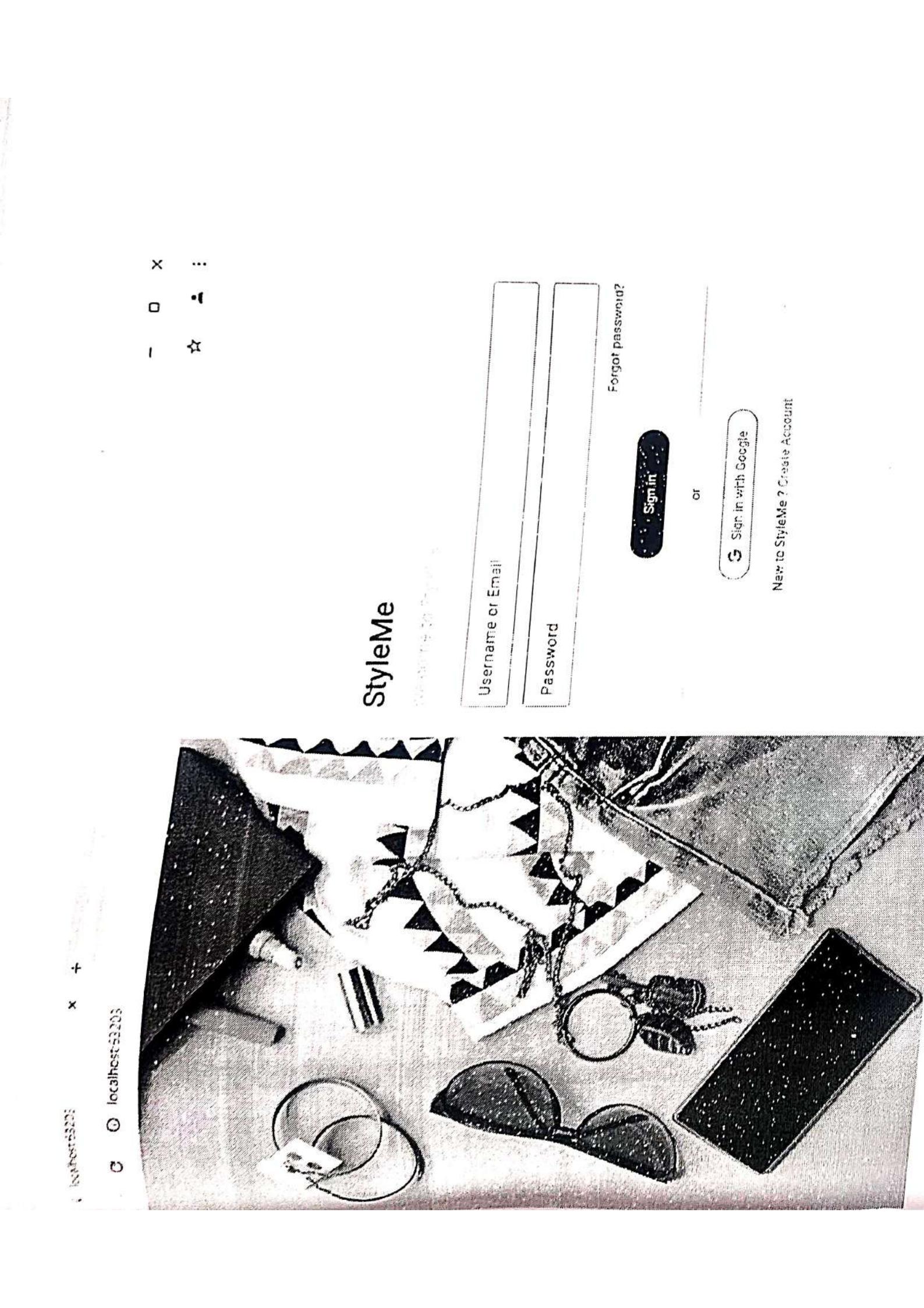
Using Network Images:
Image.network('https://example.com/image.png', width: 100, height: 100);

3. Adding Custom Fonts:

Flutter allows integrating custom fonts for a unique design.

Steps to add a custom font:
Place font files in the assets/fonts folder.

Define them in pubspec.yaml:
yaml flutter:



StyleMe

Username or Email

Password

[Forgot password?](#)

[Sign in](#)

or

[Sign in with Google](#)

[New to StyleMe? Create Account](#)

Experiment No 4

SCM

Aim: Create an interactive form using form widget

Theory: Introduction to forms in flutter

Forms in flutter allows user to enter & submit data. The form widget is used to group multiple form fields such as text form field, & etc manage their validation & submission

Components of flutter form

global key < formstate >: helps in managing the form's state

Textformfield : Always used to Input Text

Validation logic: It should include validation & submit the form

Submit button: used to validate & submit the form

Implementation steps:

Create a form with a global key < formstate >

Add a Textfield with validation logic

Create a button to validate & submit the form

Conclusion: This experiment demonstrates how to create an interactive

```
import 'package:flutter/material.dart';

void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text("Create Account"),
        backgroundColor: Colors.blueAccent,
      ),
      body: Padding(
        padding: const EdgeInsets.all(20.0),
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            Text(
              "Create Account Form",
              style: TextStyle(fontSize: 24, fontWeight: FontWeight.bold),
            ),
            SizedBox(height: 10),
            Text("Complete your profile by filling in this account creation form."),
            SizedBox(height: 20),
            TextField(
              decoration: InputDecoration(labelText: "First Name"),
            ),
            TextField(
              decoration: InputDecoration(labelText: "Last Name"),
            ),
            TextField(
              decoration: InputDecoration(labelText: "Email"),
            ),
            TextField(
              decoration: InputDecoration(labelText: "Birthdate"),
              keyboardType: TextInputType.datetime,
            ),
            TextField(
              decoration: InputDecoration(labelText: "Preferred Username"),
            ),
            SizedBox(height: 20),
            Text("Do you want to receive updates by email?"),
            Row(
              children: [
                Text("Yes"),
                Text("No"),
              ],
            ),
          ],
        ),
      ),
    );
  }
}
```

Create Account Form

Complete your profile by filling in this account creation form.

First Name

Last Name

Email

Birthdate

Preferred Username

Do you want to receive updates by email?

Yes

No

Create Account

Experiment No 5

Page No.	1
Date	10/10/2023

Aim: To apply navigation, routing & gestures in flutter app

Theory: Navigation in flutter

Navigation in flutter refers to switching b/w different screens (pages) in an application. It is managed using Navigator widget, which maintains a stack of routes.

Types of Navigation

1. Push Navigation :- (Navigator, push) adds new screen on top.
2. pop navigation :- (Navigator, push pop) removes the current screen
3. Named routes :- (Navigator, pushNamed) navigate using pre-defined route name

Routing in flutter:

Routing is the mechanism that determines how a user navigates b/w diff pages.

Types of routing

1. imperative
2. declarative

Gesture In flutter:

Flutter provides the GestureDetector widget to handle user touch interactions like tapping, swiping & dragging.

1. Tap gesture
2. double tap gesture

Conclusion:

Navigation allows smooth transition between the screens. Routing makes the app structured & easier to understand. Gesture enhances user interaction by enabling smooth and intuitive navigation.

Aim:

To apply navigation, routing, makes the app structured & easier to understand. Gesture enhances user interaction by enabling smooth and intuitive navigation.

Navigation, routing, and gesture which manages a stack of routes and pop methods. Name push and pop methods. Name more manageable.

Gestures enhance user interaction. GestureDetector widget. ComposeGestureDetector.

By effectively implementing navigation, resulting in seamless dynamic, resulting in seamless navigation.

Conclusion:

Implementing navigation, routing, and gesture which manages a stack of routes and pop methods. Name push and pop methods. Name more manageable.

MAD Ex 5

Aim: To apply navigation, routing, and gestures in a Flutter app.

Theory:

Navigation, routing, and gestures are essential for building interactive and user-friendly Flutter applications. Navigation allows users to move between screens using the Navigator widget, which manages a stack of routes. Routing can be implemented using named routes or direct push and pop methods. Named routes help in organizing navigation, making the app structure more manageable.

Gestures enhance user interaction by detecting taps, swipes, and drags using the GestureDetector widget. Common gestures include tapping for button clicks, swiping for navigation, and pressing for additional actions. Flutter's built-in gesture recognition makes UI interactions smooth and intuitive.

By effectively implementing navigation, routing, and gestures, Flutter apps become more dynamic, resulting in seamless transitions and an enhanced user experience.

Conclusion:

Implementing navigation, routing, and gestures in a Flutter app enhances user experience by enabling smooth screen transitions and interactive controls. The Navigator and named routes simplify navigation, while the GestureDetector widget allows intuitive touch interactions. Properly integrating these features ensures a seamless, responsive, and user-friendly app.

Wardrobe Dashboard

Welcome to StyleMe

Enhance your style with personalized recommendations

Find Wardrobe Faculty



2



Experiment No 6

90/101
Aim: setting up flutter with flutter for android & ios app

Theory:

What is flutter?
Flutter is a Back-end -on -a -service (aaS) provided by google that helps developers build mobile & web applications with features like authentication, real time database, cloud storage & analytics.

Steps to set up flutter in a flutter app

Step 1: create a flutter project

Step 2: add flutter to your flutter app

Step 3: install flutter packages in flutter

Step 4: Initialize -11-

Step 5: verify flutter connection

Conclusion: flutter is a powerful tools that simpleifies backend development for mobile apps. The integration process includes creating a flutter project, adding platform-specific configuration files & initializing flutter in

Auth

Authentication

Sign in with Email or Phone number
Sign in with Google or Facebook
Sign in with Twitter or LinkedIn
Sign in with GitHub or Bitbucket
Sign in with Microsoft or OpenID Connect

The following Authentication features will stop working when Firebase Dynamic Links is disabled:
Authentication for mobile apps as well as Cordova/React support for web apps

Search by email address, phone number, or user ID

Field	Value
Provider	password
Created	Mar 22, 2015
Last login	Mar 27, 2015

PWA Experiment - 7

DATE:

Aim: To develop a progressive web app (PWA) that can be installed on a device like a native application & provide offline functionality using a service worker & a manifest file

Theory: A progressive web App (PWA) is a modern web application that provides an app-like experience on the web. It allows users to install the app on their device, work offline, & local content faster PWA's leverage the following key components:

A background script that enables offline functionality by caching resources. It intercepts network requests & serves web app manifest

A JSON file that provides metadata about the PWA such as the name, theme, color, start URL & icons. It makes the web app installed on device

Cache first strategy

The cache first approach is used in this PWA meaning that if resources are available in cache they are served first, otherwise they are fetched from the network

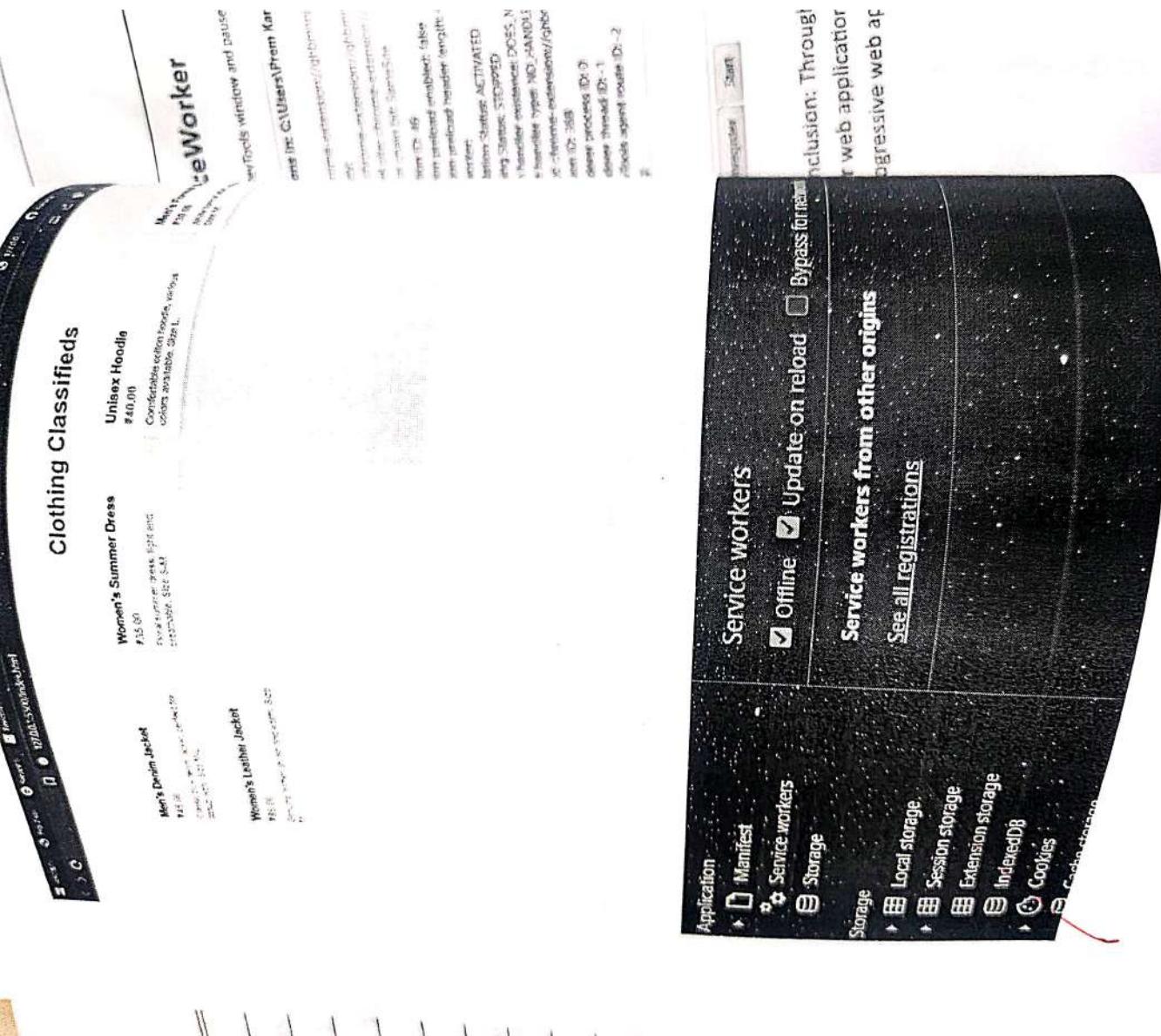
Conclusion: The implementation successfully demonstrates the PWA architecture by enabling installation offline access & caching users can install the app & even if there is not internet connected previously loaded pages will still be accessible. The project highlights the benefits of PWA such as improved performance

② manifest.json

```
  "name": "Clothing Classifieds CRASH",  
  "short_name": "Classifieds",  
  "start_url": "index.html",  
  "display": "standalone",  
  "background_color": "#4682B4",  
  "theme_color": "#333333",  
  "description": "A PWA for showcasing clothing classifieds",  
  "scope": "https://clothing-classifieds.com",  
  "orientation": "portrait",  
  "categories": ["shopping", "clothing", "fashion"],  
  "icons": [  
    {  
      "src": "images/app-icon.png",  
      "sizes": "512x512",  
      "type": "image/png"  
    },  
    {  
      "src": "images/app-icon1x.png",  
      "sizes": "512x512",  
      "type": "image/png"  
    }  
  ],  
  "screenshots": [  
    {  
      "src": "images/screenshot1.png",  
      "type": "image/png",  
      "sizes": "512x512",  
      "label": "Homepage - Clothing Classifieds",  
      "form_factor": "narrow"  
    },  
    {  
      "src": "images/screenshot2.png",  
      "type": "image/png",  
      "sizes": "512x512",  
      "label": "Grid view of listing",  
      "form_factor": "wide"  
    }  
  ]  
}
```

③ index.html

Clothing Classifieds



PWA ~~and~~ Experiment 8.

Aim: To develop progressive web app (PWA) that enable installation offline functionality using a service worker and a web manifest file

Theory: A progressive web app (PWA) is a modern web application that provides an app-like experience on the web. It allows users to install the app on their device, work offline & load content faster. PWA's leverage the following key components:

1. Service worker

A background script that enables offline functionality by caching resources. It intercepts network requests & serves cached response when offline.

2. Web App Manifest

A JSON file that provides metadata about the PWA such as the app's name, theme, short start URL & icons. It makes the web app installed on device.

3. Caching strategy

The cache first approach is used in this PWA meaning that if resources are available in cache they are served first otherwise they are fetched from the network.

Conclusion: The implementation successfully demonstrates the PWA architecture by enabling installation offline access & caching users can install the app even if there is no internet connection previously loaded pages will be still accessible. The project highlights the benefits of PWA such as improved performance.

```
1 Welcome          2 index.html  3 manifest.json
4 offline.html  5 ExtensionLiveServer 6 serviceWorker.js
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
559
560
561
562
563
564
565
566
567
568
569
569
570
571
572
573
574
575
576
577
578
579
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
618
619
619
620
621
622
623
624
625
626
627
628
629
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
688
689
689
690
691
692
693
694
695
696
697
698
699
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
788
789
789
790
791
792
793
794
795
796
797
798
799
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
879
879
880
881
882
883
884
885
886
887
888
889
889
890
891
892
893
894
895
896
897
898
899
899
900
901
902
903
904
905
906
907
908
909
909
910
911
912
913
914
915
916
917
918
919
919
920
921
922
923
924
925
926
927
928
929
929
930
931
932
933
934
935
936
937
938
939
939
940
941
942
943
944
945
946
947
948
949
949
950
951
952
953
954
955
956
957
958
959
959
960
961
962
963
964
965
966
967
968
969
969
970
971
972
973
974
975
976
977
978
979
979
980
981
982
983
984
985
986
987
988
989
989
990
991
992
993
994
995
996
997
998
999
999
1000
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1098
1099
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1189
1190
1191
1192
1193
1194
1195
1196
1197
1197
1198
1199
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2498
2499
2500
25
```

Clothing Classifieds

1114 Simpler Dress

Men's Denim Jacket
#145

Women's Summer Dress

Unisex Handie

Women's Leather Jacket

Women's Summer Dress
P. 1415

Men's Formal Shirt

Men's Denim Jacket
#4540

Women's Summer Dress \$35.00

Men's Formal Shirt
\$20.00

Woman's Leather Jacket
\$5.60

Clothing Classifieds

© Whirl's New in Day Tools 135

102

157

Aim:

To implement & understand service worker
PWA like fetch sync & push for an e-commerce

DATE:

Theory: Service worker is a background script that runs independently in a browser & enables

functionalities like offline caching & enables push notifications. It acts as proxy between the web application & the network enhancing performance, availability & user experience

Key features of service workers:

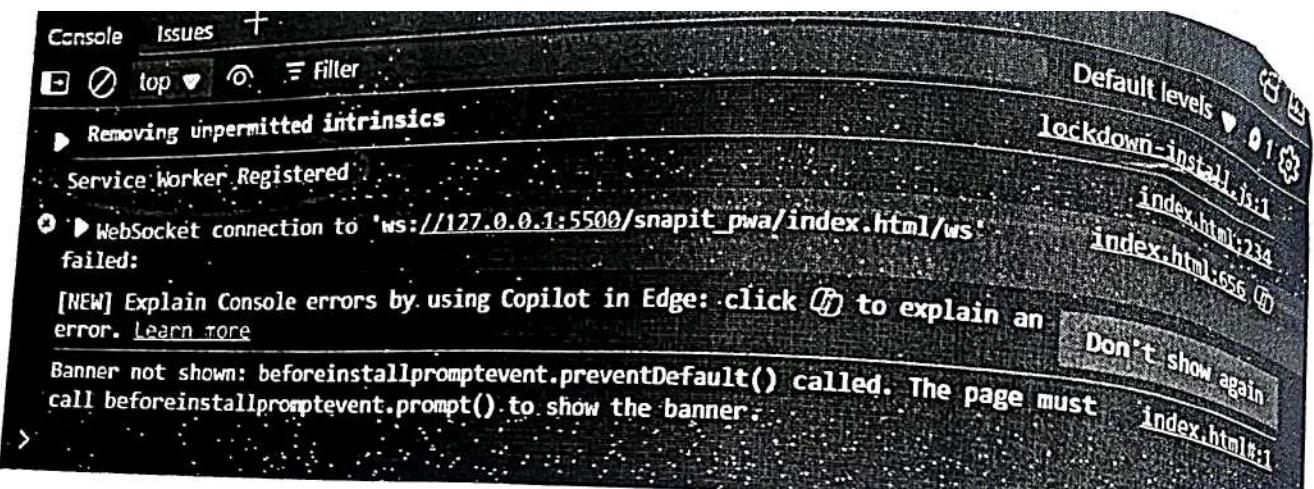
- Runs in the background without interaction
- Intercepts network requests & manages caching using the cache API
- Works only over HTTPS for security reasons
- Uses promises & standard DB to handle data persistence

~~fetch event: Manages network requests & caching~~
~~Cache first: uses cached data first, fetching from the network only if needed~~
~~Network first: tries fetching from network first, falls back to cache if offline~~
~~Sync event: enables background sync when internet is restored~~
~~push event: handles push notification service~~

Conclusion: Thus we successfully implemented service worker which fetch, sync & push in PWA enabling offline functionality

```
// activate event
self.addEventListener("activate", event => {
  event.waitUntil(
    caches.keys().then(keys => {
      promise.all(
        keys.map(key => {
          if (key != staticCacheName) {
            return caches.delete(key);
          }
        })
      );
    })
  );
});

// fetch event
self.addEventListener("fetch", event => {
  event.respondWith(
    caches.match(event.request).then(response => {
      return response || fetch(event.request).catch(() => caches.match("./offline.html"));
    })
  );
});
```



Conclusion: We implemented the functionality of offline web cache capture so that in the absence of a stable internet connection, the app would display a generic waiting page.

~~(JULY)~~ ✓
Aim: To study & Implement the development of an e-commerce PWA to github pages

Theory: Github pages is a free hosting service provided by Github for deploying static pages directly from repository. It is used for hosting personal projects, documentations & simple web applications.

- features of github pages
- supports Jekyll blogging
- allows custom URLs

provides an automatic page generator

- pros of github pages
- free & easy to use
- direct integration with github repositories
- supports custom domains

~~Cons of github pages~~

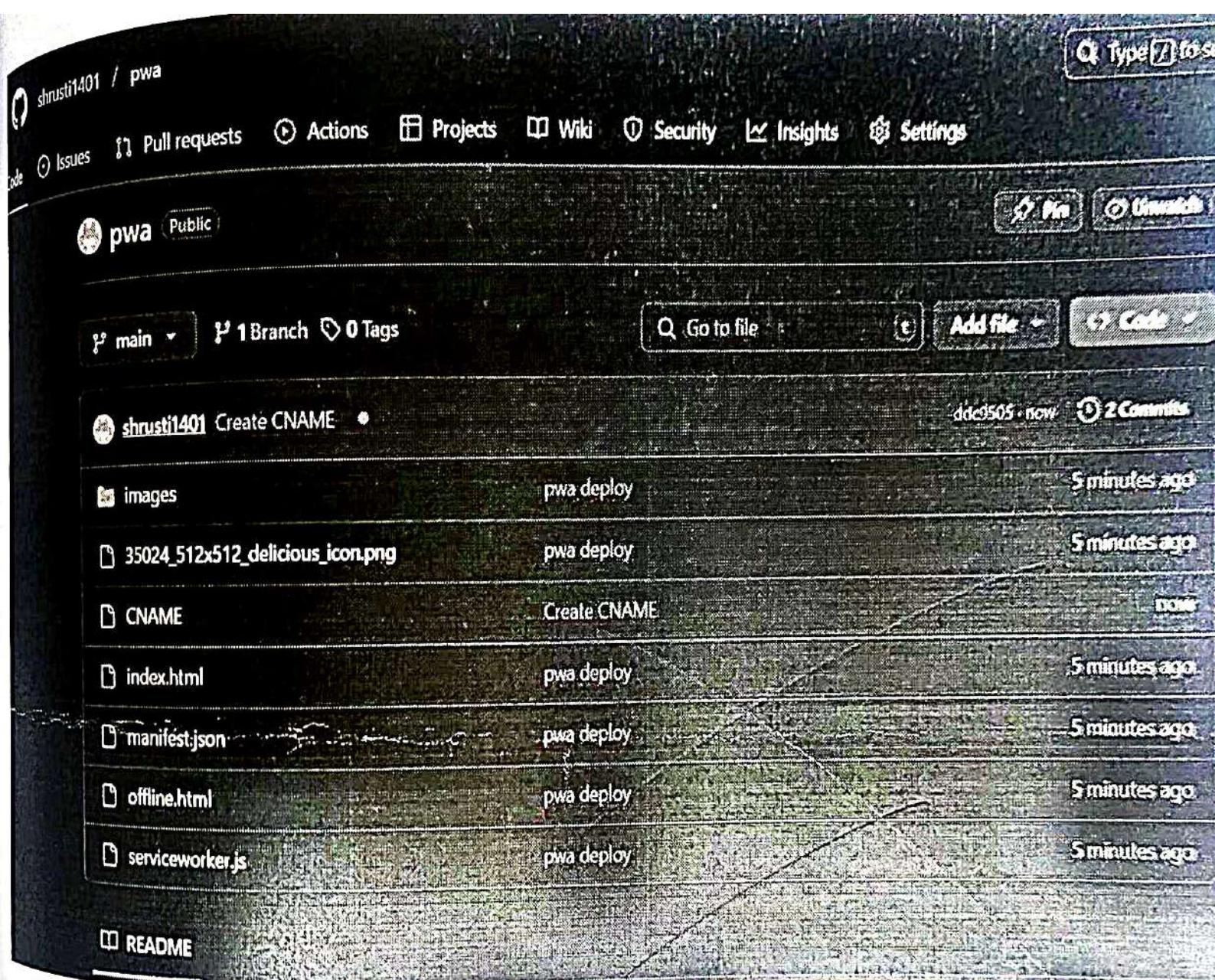
- code is public unless using a paid private repository
- limited plugin support for Jekyll

Firebase as an alternative

Firebase is a real-time backend platform by Google offering cloud-based storage & authentication services. It allows real-time data synchronization & is widely used for dynamic applications.

- No

Conclusion: Thus we successfully deployed our e-commerce PWA using Github pages, leveraging its free & easy to setup for static web hosting.



App Manifest

[snapit_pwa/manifest.json](#)

Errors and warnings

- ⚠ Richer PWA Install UI won't be available on desktop. Please add at least one screenshot `form_factor` set to `wide`.
- ⚠ Richer PWA Install UI won't be available on mobile. Please add at least one screenshot `form_factor` is not set or set to a value other than `wide`.

Identity

Name SnapShop

Short name SnapShop

Description A QR-based shopping app

Computed App ID https://swayamraut8.github.io/snapit_pwa/index.html ⓘ Learn

Note: `id` is not specified in the manifest, `start_url` is used instead.

To specify an App ID that matches the current identity, set the `id` in `manifest.json` to `/snapit_pwa/index.html` ⓘ.

Presentation

Start URL index.html

Aim: To study & implement the google lighthouse PWA analysis tool to test the progressive web App (PWA) function

Theory: google lighthouse is an open source tool by google used to audit web applications based on various parameters including performance PWA compliance, accessibility & best practices. It helps developer analyze & optimize web apps to function like native applications.

key audit metrics

- i) performance - Measures loading speed, rendering time & responsiveness
- ii) PWA compliance - Ensures the app meets PWA standards such as service workers, offline functionality & manifest
- iii) accessibility - evaluates the app's usability differently - users, including screen-reader compatibility
- iv) best practices - check HTTPS usage, security policies & deprecated code elements

Light house provides actionable insights to improve website efficiency, ensuring a better user experience & higher engagement

Conclusion :- Thus we successfully used the google lighthouse PWA analysis tool to evaluate & optimize the PWA improving its performance, accessibility & compliance with best practices

METRICS

Exp

First Contentful Paint

0.2 s

Largest Contentful Paint

0.2 s

Total Blocking Time

0 ms

Cumulative Layout Shift

0

Speed Index

0.2 s

 View Treemap

10-

How service workers use IndexedDB

opening the database:

let db;

let request = indexedDB.open('My database', 1);

request.onsuccess = function (event) {

db = event.target.result;
};

creating a store & adding Data

request.onupgradeneeded = function (event) {

let db = event.target.result;

let store = db.createObjectStore('users', {keypath: 'id'});

store.add({id: 1, name: 'John Doe', age: 25});
};

fetching data in service worker

let transaction = db.transaction('users', 'readonly');

let store = transaction.objectStore('user');

let getUser = store.get(1);

getUser.onsuccess = function () {

console.log('get user result');
};

cache.open('app-cache')

return cache.addAll(['1', 'Index.html'])

});

});

});

ii) Activation phase

The old service worker is replaced with new one
request-serves cached files & types data

Q4) Explain the use of indexed DB in the service worker for storage

Ans: use of Indexed DB in service worker for data storage:
DB is a nosql database that stores large amount
structured data like JSON objects. It helps in
online saving & retrieving data efficiently

why use IndexedDB in service workers?

- i) offline support - store data when offline & sync it to
- ii) efficient storage - saves structured data like user settings, cart items or form inputs
- iii) faster access - retrieves data quickly without using a network request
- iv) persistent data - data remains saved even after the browser is closed

my difference:

responsive adopts dynamically to all screens
 fluid resizes smoothly but may not be fully optimized
 adaptive loads different layout based on device-type

describes the lifecycle of service workers, including registration, initialization & activation phases

lifecycle of service workers. A service worker is a script that runs in the background and helps a web app work offline, local faster & send push notifications. Its lifecycle as 3 main phases

registration phase: The browser registers the service worker using JavaScript

if ("serviceWorker" in navigator) {

navigator.serviceWorker.register

. then c) \Rightarrow console.log ("service registered")

. catch (error) \Rightarrow console.log ("register failed", error);

installation phase

The service worker downloads necessary files (HTML, CSS, JS) and

if successful, it moves to the activation phase

code eg

self.addEventListener ('install' event \Rightarrow

performance	fast with service workers	fast with new installation
updates	automatic	Manual
development cost	lower	higher

Q2) Define responsive web design and explain its importance in the context of progressive web apps. Apps compare contrast responsive, fluid, & adaptive web design approaches.

Soln:- definition of responsive web design
 Responsive web design (RWD) is a technique that web pages adjust automatically to different screen devices. It ensures good user experience on mobiles (tablets) & desktops without needing separate

Importance of responsive design in PWAs:

- I) better user experience
- II) faster load time
- III) SEO benefits
- IV) cost effective

Comparisons:

Approaches

Responsive

How it works

CSS media queries
& flexible grids

pros

flexible

works on all devices

cons

can't
comp.

separate teams for iOS & android

3) consistent UI: flutter - renders everything
in its own engine, ensuring a UI
across devices.

Q16. Discuss how flutter framework differs from traditional approaches & why it has gained popularity in developer community?

Ans. flutter uses a single codebase for multiple platforms, unlike traditional native development that requires separate code for iOS (Swift) & android (Kotlin). It does not rely on platform-specific UI components but instead renders everything using its own Skia engine, ensuring consistency. Unlike React Native, which uses a JavaScript bridge, flutter compiles directly into ARM code, offering better performance. Hot reload feature allows developer to see changes instantly, making development faster & more efficient.

flutter has gained popularity due to its fast development cost efficiency & cross platform support. Businesses prefer it as it reduces development costs while delivering high performance apps. Its customizable widget system ensures a smooth user experience.

Q2 a) Describe the concept of the widget tree in flutter. Explain the widget composition. How complex UIs are built using this concept?

Describe the concept of the widget tree in flutter
explain the widget composition is used to build
complex UI.

In flutter everything is a widget (e.g. button,
text, layouts etc) These widgets are arranged in a
hierarchical structure known as the widget tree. The
widget tree determines the UI.

Widget composition to build complex UI:
Flutter encourages a composition-based approach
rather than inheritance.
Instead of creating large, monolithic widget, developer
will create small, reusable widget that are combined to
form complex UIs.

- A column widget can hold multiple Text & button
widget, creating a structured widget

~~Provide an~~ of commonly used widgets & their roles in
building a widget tree

1) Structural widget

Scaffold: provide basic structure of a screen

Container: used for layout styling

Column & Row: used for vertical & horizontal layout

2) Interactive widget

Text field: for user input

Elevated button: clickable buttons

iii) styling widget
• padding: adds spacing around widget
• align, center: adjust alignment

iv) list & scrollable widget

- listview: scrollable widget
- gridview: provide / display items in grid

en. simple widget Tree

```
scaffold (  
  appBar: AppBar (title: Text ("flutter App"),  
  body: Column (children: [  
    Text ("welcome to flutter !"),  
    ElevatedButton (onPressed: () {  
      print ("click Me");  
    }, child: Text ("click Me"))  
  ]));  
);
```

Q3a) Discuss the importance of state management in flutter

Ans: The importance of state management in flutter applies to handling dynamic data that changes over time. In flutter, the UI reinitializes when the state changes, so proper state management helps in maintaining the state and ensuring the app remains interactive & responsive. It also improves code maintainability & better UI behaviour.

Compare and contrast the different state management in flutter approaches available in flutter, such as `setstate`, `provider` & `Riverpod`, provide scenarios where each approach is suitable.

comparison of state management approaches in flutter approach description suitable scenarios `setstate` basic state management by calling `setstate()` to update UI. small apps, simple UI updates (e.g., toggling a switch) `provider` uses `inherited widget` to efficiently manage state across the widget tree. ~~medium sized apps~~ using global state sharing (e.g., user authentication). `Riverpod` more scalable than `provider` with improved dependency injection & state handling. large, complex apps requiring modular & scalable state management (e.g. e-commerce apps)

Q3. Q-20 - Explain the process of integrating `firebase` with a flutter application.

Discuss the benefits of using `firebase` as a backend solution.

Integrating `firebase` with flutter & its benefits.

on: Integration process:

Setup `firebase` console.

Create a `firebase` project.

Register the app for android & iOS.

Download & add `google-services.json` (android) or `google-service-info.plist` (ios).

Install `firebase` dependencies.

yaml

dependencies:

firebase-core: latest-version

firebase-auth: latest-version

cloud-functions: latest-version

Initialize firebase in flutter

void main() async {

 WidgetsFlutterBinding.ensureInitialized();
 await Firebase.initializeApp();

 runApp(MyApp());

Benefits:

No need to manage servers & backend - as-a-service
Wide authentication, database & cloud for
scalable & cost-effective

Q4b. Highlight the firebase services commonly used
in flutter development & provide brief overview
Ans: commonly used firebase services in flutter
synchronization service in flutter
firebase authentication, firebase cloud firestore, sign-in (Email, Facebook)
cloud storage, NoSQL database for some data syncing services
manage files (image, videos) cloud messaging
notifications, firebase analytics, app usage analysis

Data synchronization in firebase:

firestore allows real time data sync using snapshot listeners

in for real-time listeners in firestore

dart

~~1. Firebase firestore.instance.collect('message').snapshot
listen ((snapshot) {
for (var doc in snapshot.docs) {
print (doc.data());
}
});~~

✓

MPL assignment 2

04/05
Page 01
Date

Define progressive web app (PWA) & explain its significance in modern web development. Discuss the key characteristics that differentiate PWAs from traditional mobile apps.

A progressive web app (PWA) is a type of web application that combines the best features of both traditional web pages & native mobile apps. It leverages modern web technologies such as service workers, manifests, & responsive design, to deliver a fast, reliable & engaging user experience across different devices.

Significance of PWAs in modern web development.

PWAs have gained popularity in modern web development due to their ability to provide app-like experiences without requiring installation from an app store.

Their significance includes cross-platform compatibility - works seamlessly across desktops, tablets & mobile devices.

Improved performance: caching & efficient loading reduce page load times.

Offline functionality - service workers enable offline access, enhancing user engagement.

Difference in PWA & Traditional Apps

features

PWA

Installation

direct from browser

Internet req.

work offline with
caching

Traditional app

download from app
store

usually require
internet

Explain the key features & 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, & desktop from a single codebase.

Key features of flutter:

~~Single codebase: unicorns, run on multiple platforms (iOS, Android, web, desktop)~~

~~Dart programming language: uses dart, which is optimized and ahead-of-time (AOT) compiled~~

~~Hot reload: instantly reflects changes in the app on the device without restarting, making development faster & more efficient~~

~~Rich widget library: provides a vast collection of customizable widgets that support material design & Cupertino styles for a native look & feel~~

Advantages of using flutter:

~~faster development time: hot reload & a single code base reduce development effort & time~~

~~cost effective: since developers write one codebase for multiple platforms, it reduces costs associated with maintaining~~