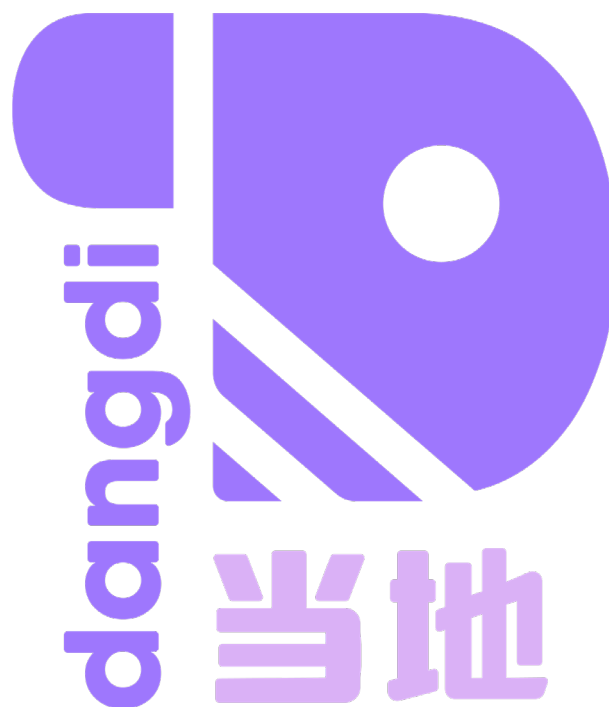


PRAKTIKRAPPORT MED REFLEKSJONER

WEBUDVIKLER



PETER RUD FRANKILD 7A

SINOLAB APS 15. JANUAR 2024 - 22. MARTS 2024





INDHOLDSFORTEGNELSE

1.0	FORSIDE	1
2.0	INDHOLDSFORTEGNELSE	2
3.0	ANSØGNINGER OG KONTRAKT	3
3.1	ANSØGNINGER	3
3.2	KONTRAKT	3
4.0	OPGAVERNE OG MÅLSÆTNING	4
4.1	OPGAVERNE	4
4.2	MÅLSÆTNING	4
5.0	OPFYLDTE JEG MINE MÅL	5
6.0	KONKLUSION	6
7.0	BIBLIOGRAFI	7
8.0	APPENDIKS	8

3.0 Ansøgninger og kontrakt

3.1 Ansøgninger

Efter at have deltaget i rekrutteringsdagen på KEA Guldbergsgade 29, fik jeg en god forståelse af hvilke virksomheder, der tilbyder praktikophold til studerende på den digitale linje. Det var en værdifuld oplevelse, talspersonerne, var både unge og voksne, som viste engagerede og sympati. Arrangementet havde repræsentanter fra både den private sektor, herunder webbureauer og formidlere, og offentlige tjenester, der arbejder med samfund og udvikling.

Jeg valgte at kontaktede en lille virksomhed, der tilbyder en kulturhistorisk digital platform. Der var nogle oplagte muligheder for at ændre deres platform fra ren JavaScript til et Python-framework, hvor kravspecifikation og teknologi ville blive sat på prøve. Efter flere samtaler besluttede talspersonen at arbejde videre med en anden kandidat. I den efterfølgende periode arbejdede jeg tæt sammen med Gitte Jul, som er praktikansvarlig, og Berit Bergmann, som er studievejleder, for at sikre mig en praktikplads.

I den efterfølgende processen valgte jeg at søge praktik hos virksomheder som potentielt arbejdede med kultur formidling. Heldigvis fik jeg et positivt svar fra Sinolab ApS, en lovende startup virksomhed med base i København. Selv om jeg var i Toronto, Canada, på det tidspunkt var jeg jublende over at blive inviteret til en samtale med Jacob hos Sinolab, hvilket jeg ivrigt accepterede.

Den 4. januar havde jeg fornøjelsen at mødes med Sinolab

for første gang. Under min samtale med Jacob, ejeren af virksomheden, havde jeg vi en god samtale om vores fælles interesser og hvordan vi kunne drage fordel af dem. Jeg nævnte min erfaring med at studere i Tyskland og et valgfag på fjerde semester i Design Across Borders, som kunne være nyttige, når Sinolab arbejde med webudviklere i flere lande. Vi talte også om forskellige aspekter rejsende værdsætter, herunder indkvartering, attraktioner, fællesskab og kommunikation på sociale medier og i det virkelige liv. Foretrækker de at shoppe i butikker eller at handle med henblik på eksport? Nyde de at se på kunst eller få viden om teknologi? Hvad med data på mobiltelefonen? Det kan være utroligt nyttigt, til at finde billig transport, foretage søgninger og planlægge dagen. Derudover kan mobil data styrke mulighederne for at kommunikere og engagere sig i kulturelle og samfundsmæssige begivenheder. Da vi blev enige om vilkårene, delte jeg straks et kontraktudkast med Jacob.

3.2 kontrakt

Vi skal arbejde med kravspecifikationer, UX, Dart programmering og app udviklingsplatformen FlutterFlow i Xcode og Android Studio. Jacob introducerede mig venligt til min arbejdsplads på Symbion Østerbro, som er et kontormiljø med fokus på digital industri og programmering. Samme dag havde jeg fornøjelsen af at møde to ansatte fra Sinolab, Xingling som er uddannet Erhvervsjournalist, og Wenhao Softwareingeniør fra DTU, sammen med andre har de udviklet Dangdi appen til App store og Google Play, samt hjemmeside, med Landing page og skrevet artikler.

Samme dag færdiggjorde vi kontrakten med et læringsmål

baseret på ideen om, at hvis vi skriver få læringsmål, men bevarer en bred vifte af faglige muligheder. Dette er hvad, vi valgte at arbejde videre med som læringsmål i kontrakten.

- Kravspecifikation af funktionalitet og teknologi til en App med form input.
- App'en skal bliver udviklet som en del af Dangdi App platform, bl.a. backend API med cascade output i Flutterflow.
- Figma UX prototype, Funktionel prototype
- Forstå og arbejde med Flutterflow, dart programmering.

4.0 Opgaverne og Målsætning

4.1 Opgaverne

Under praktikopholdet havde jeg mulighed for at arbejde med flere forskellige softwaregrænseflader, herunder Webflow og Flutterflow. Webflow et modulatororienteret no-code software, der nemt genererer visuelle imponerende sider ud fra templates og JQuery microanimationer. Du kan eksportere koden eller redigere den i Custom Code Editor. FlutterFlow er på den anden side specifikt designet til udvikling af mobileapplikationer i et no-code miljø, rettet mod en cross operativsystem deployment som, web, android og ios application.

Til at starte med gennemgik vi min opsætning og delte ressourcer fra FlutterFlow, Github, Postman, Figma, Ghost MySQL og Trello. I min første opgave skulle jeg udvikle en onepage tilføjelse til Dangdi.life siden. Jeg fik et beskrivende Word dokument af min kollega Min Content management.

4.2 Målsætning

Jeg valgte at udvikle tre frontend JavaScript filer, en til listevisning af FAQ json objekter og en til søgefunktionen. Den sidste JavaScript fil var fetch af Ghost MySQL database på AWS.

```

js > JS article.min.js > ...
1  fetch("./json/ghostCredential.json")
2  .then((response) => response.json())
3  .then((jsonData) => {
4      const ghostApiUrl = `${jsonData.url}/ghost/api/content/posts?key=${jsonData.key}`;
5      const rootDiv = document.getElementById("root");
6      const unorderedList = document.createElement("ul");
7      unorderedList.setAttribute("class", "unorderedList");
8      fetch(ghostApiUrl)
9          .then((response) => {
10             return response.json();
11         })
12         .then((data) => {
13             //Create template
14         })
15         .catch((err) => {
16             const errorMessage = document.createElement("marquee");
17             errorMessage.textContent = "Arg.. API is not working!";
18             rootDiv.appendChild(errorMessage);
19         });
20     });
    
```

1) Fra article.js - API fetch af Ghost MySQL databasen

Vores hovedmål var at skabe en interessant landingsside, der opfylder kravspecifikationer og UX. Søge-funktionaliteten og listevisningen skulle fetches fra en enkelt json fil med boolean true false af fire kategorier for at få konsistens uden duplikater. Søgningen ved hjælp af filter() funktionen og ved at konvertere søgefeltet, både i issues og issue_text til små bogstaver. Efter test blev alle JavaScript og css filer komprimeret til én linje i \$terser xxx.js -o output.min.js.

```

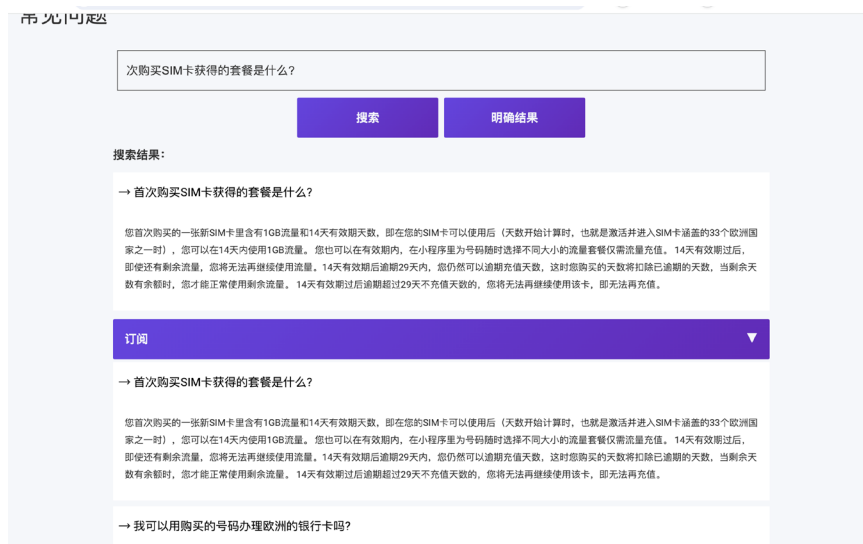
24  var filterObjs = faqissues.filter((function (item) {
25      const searchText = searchTerm.toLowerCase();
26      return item["issue"].toLowerCase().includes(searchText)
27      || item["issue_text"].toLowerCase().includes(searchText)
28  }));
    
```

2) Fra search.js - filter()

Den funktionelle prototype blev udviklet ved hjælp af Figma.

Derudover blev FAQ json data placeret i den rette form i et Google sheet.

Der blev brugt tid på at sætte projektet op i Flutterflow, Android studie og VS Xcode med tilhørende download og installation af mobile emulators. Flutter har en smart opsætning af CRUD til Firebase, no-SQL db fra Google og DNS fra Sendgrid af email subscription, authorisation, API m.m. Udviklingen af Dangdi.life, omfattede signup, login og logout samt CRUD funktioner ved hjælp af Dart FlutterFlow framework med localhost MAMP MySQL database.



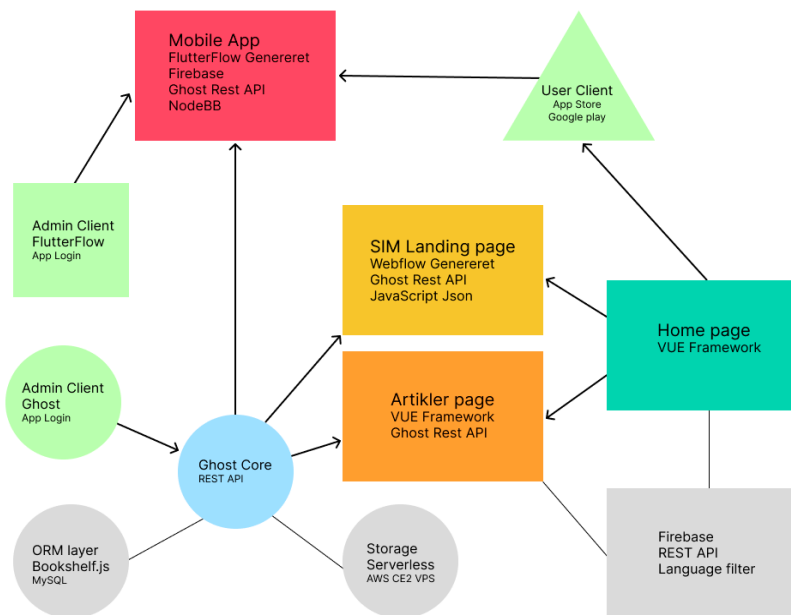
3) Udskrift i browser af søge funktion og list af FAQ Dangdi.life landing page

Under et andet projekt forsøgte jeg at udvikle Flutter framework som en web applikation med local MySQL fra MAMP. Det var oplagt at lægge mest muligt SQL kode funktionalitet i phpMyAdmin. SQL koden omfatter bl.a. CREATE SELECT INSERT INTO UPDATE DELETE, med den mere komplicerede brug af TRIGGER effekt f.eks. INSERT, som trigger en ny

column efter en JavaScript funktion, er eksekveret f.eks. hashed password eller en api-key, som så tilhører den nye bruger.

5.0 Opfyldte jeg mine mål

De mål, vi satte i starten af perioden, er blevet overgået og de små problemer, der opstod, blev evalueret, prioriteret og løst. Programmering og research, blev tilrettelagt mellem Wenhao og mig. Her kom daglige noter og ugentligt todo'er og done'er i Trello mig til hjælp, da jeg nemt kunne fortsætte, hvor jeg slap.



4) Dangdi's Digitale Platform. Centralt display på Homepage og Mobile App

Den digitale platform Dangdi.life er hjertet af Sinolab og omfatter udviklingen af en Web applikation i VUE, som er blevet produceret og deployed. Hjemmesiden giver information om, hvem målgruppen er, og hvordan de bruger platformen. Der

er touchpoint med links til de lande, hvor Dangdi's services kan tilgås, og en landingsside, der forklare om Dangdi's samarbejdspartnere. Formidling både i Europa og Kina.

Jeg fik erfaring i at udvikle hjemmesider i Webflow. Webflow's fordel ligger hovedsagligt i at gøre det nemt at udvikle hjemmesider, som er lingvistisk opdateret live i skyen. Brugen af templates kan virke inspirerende, men det er værd at nævne at datamængden hurtigt kan stige til utilsigtede mængder.

Jeg valgte at udvikle et fullstack Flutter framework i Dart programmering sprog, der inkluderer en dynamisk blog med registrering, login og dashboard. Dette skulle give en komplet forståelse af Dart programmering og flutterFlow framework. Jeg brugte Mamp Apache til at opsætte en MySQL database med to tabeller, users og blog.

Der blev også tid til at researche på implementeringen af MitID fra Signcat i Holland. For at aktivere 2-faktor funktionaliteten skal du blot tilføje domæne og indsætte en kort code snippet på index.html siden, der omdirigerer til hjemmesiden. Jeg søgte indsigt i opsætningen af Ghost på AWS EC2 PVS Linux Ubuntu med MySQL database.

Hos Sinolab arbejder vi i et fælles kontormiljø, hvor teamet bag Dangdi.life mødes hver frokost til en buffet. Det er en god mulighed for at diskutere arbejdet og give feedback. Det var også her mange møder blev arrangeret til senere på dagen. Når teamet omtales, er det værd at bemærke at der er flere eksterne partnere, der tager sig af AWS backend opsætningen og valg af frontend udviklingen, her fik jeg nogle gode svar på komplicerede sammenhænge.

6.0 Konklusion

Afslutningsvis kan jeg konkludere at det har været en fornøjelse at arbejde i et lille team hvor forventningerne er høje både til processen og produktet, hvilket har virket meget inspirerende. Arbejdsprocessen har været effektiv i forhold til at opfylde de planlagte krav fra kontrakten og vigtigheden i at mindske tidsforbrug under udviklingsfasen gennem brug af templates.

Et kendt problem for en startup virksomheder er økonomien. Derfor skal der tages hensyn og udvikles ud fra en fælles interesse. Konceptet skal præsentationen med fælles værdier og økonomi for øje, fundraising i daglig tale.

7.0 Bibliografi

- <https://dangdi.life>
- <https://trello.com/invite/b/XOGHlbyJ/ATTIa6d6a5977b5612bd072505f3e95cc15804084832/sinolab-praktikophold>
- https://docs.google.com/spreadsheets/d/1E9tTDa_KTKCylbrNa_fINuTjceP91NBvHZCwto80ZUc/edit?usp=sharing
- <https://www.figma.com/file/MyiAZ0u9ni1TYvDQsy0qP7/Dangdi?type=design&node-id=0%3A1&mode=design&t=kp7IHSh7lu10EUec-1>
- <https://getpostman.com>
- <https://developer.apple.com/xcode/>
- <https://developer.android.com/develop>
- <https://developer.android.com/guide/topics/manifest/manifest-intro>
- <https://intercom.help/flutterflow/en/>
- <https://docs.flutter.dev>
- <https://github.com/FlutterFlow/flutterflow-issues/issues/2021>
- <https://ghost.org/docs/content-api>
- <https://sendgrid.com>
- <https://firebase.google.com>
- <https://nodebb.org/>
- <https://uncss-online.com/>
- <https://signicat.com/>
- <https://ghost.org/docs/content-api/>
- <https://dart.dev/get-dart>
- <https://www.hacksplaining.com/>
- <https://www.youtube.com/@FlutterFlow/videos>
- <https://bookshelfjs.org/>

8.0 Appendiks

phpMyAdmin Triggers: Trigger hashed password:

```
before_insert_user_hashed_password
/*SET NEW.password = generateHashedPassword(NEW.password)
```

Trigger api key:

```
before_insert_user_api_key
SET NEW.api_key = generateApiKey()
```

phpMyAdmin functions: generate ApiKey:

```
BEGIN
    DECLARE hex CHAR(32);
    SELECT LOWER(SUBSTRING(MD5(UUID()), 1, 32)) INTO hex;
    RETURN hex;
END
```

generate Hashed Password:

```
BEGIN
    DECLARE salt CHAR(22);
    DECLARE hashedPassword VARCHAR(255);

    -- Generate a random salt
    SET salt = SUBSTRING(MD5(UUID()), 1, 22);

    -- Hash the password using BCRYPT with the generated salt
    SET hashedPassword = CONCAT('$2y$', '10$', salt, '$', MD5(inputPassword));

    RETURN hashedPassword;
END
```


phpMyAdmin - username, password, email, api_key, created_at, updated_at

✓ Viser rækkerne 0 - 4 (5 i alt, Forespørgsel tog 0.0082 sekunder.)

```
SELECT * FROM `users`
```

Profiling [Redigér indlejret] [Ret] [Forklar SQL] [Fremstil PHP-kode] [Opdater]

Vis alle | Antal rækker: 25 | Filtrer rækker: Søg i denne tabel | Sort by key: Ingen

+ Indstillinger

		id	username	password	email	api_key	created_at	updated_at				
<input type="checkbox"/>	Ret	<input type="checkbox"/>	Kopi	<input type="checkbox"/>	Slet	1	sample_user	hashed_password	sample@email.com	77526e0b39b9ecc89a30817c7394aa89	2024-03-06 13:16:05	2024-03-06 15:44:54
<input type="checkbox"/>	Ret	<input type="checkbox"/>	Kopi	<input type="checkbox"/>	Slet	2	Asample_user	hashed_password	A@aa.com	sample_api_key	2024-03-06 15:17:54	2024-03-06 15:17:54
<input type="checkbox"/>	Ret	<input type="checkbox"/>	Kopi	<input type="checkbox"/>	Slet	5	user	\$2y\$10\$JPeD73ymWNuJXJh4RcR1xOe0xX/knUWR8LvWqNx67p...	pete33t5@stud.kea.dk	c1b288b3d56336dc1099bad99c223715	2024-03-06 16:28:10	2024-03-06 16:28:10
<input type="checkbox"/>	Ret	<input type="checkbox"/>	Kopi	<input type="checkbox"/>	Slet	13	username	\$2y\$10\$b011507d32ab0eabff520c\$5f4dcc3b5aa765d61d83...	email@example.com	6df5733ffe526904d7e58d6cc47b8da4	2024-03-07 00:01:11	2024-03-07 00:01:11
<input type="checkbox"/>	Ret	<input type="checkbox"/>	Kopi	<input type="checkbox"/>	Slet	70	user	\$2y\$10\$Q3CT/e/5tqJdfPuoMDOBeZZg2jnExrgxMath9MIPHr...	p@pe.com	f2387158892076f47c5e23b9f824efab	2024-03-11 11:25:26	2024-03-11 11:25:26

Med det markerede: Ret Kopi Slet Eksportér

Desktop Postman - login med email, password, api_key, username

Home Workspaces API Network

Team Workspace

Overview **POST** http://localhost:8888/u **GET** http://localhost:8888/u

http://localhost:8888/userdata.php?email=p@pe.com&password=11%23%23aaa&api_key=sample_api_key&username=user

GET http://localhost:8888/userdata.php?email=p@pe.com&password=11%23%23aaa&api_key=sample_api_key&username=user

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Key	Value	Description
<input checked="" type="checkbox"/>	email	p@pe.com
<input checked="" type="checkbox"/>	password	11%23%23aaa
<input checked="" type="checkbox"/>	api_key	sample_api_key
<input checked="" type="checkbox"/>	username	user
<input type="checkbox"/>	Key	Value
<input type="checkbox"/>		Description

Body Cookies Headers (10) Test Results

Status: 200 OK Time: 7 ms Size: 461 B Save as example

Pretty Raw Preview Visualize HTML

1

Postbot Runner Start Proxy Cookies Trash

Flutter framework opsat i htdocs med php API filer connected til phpMyAdmin

```
blog.php
1 <?php
2 header("Access-Control-Allow-Origin: *");
3 header("Access-Control-Allow-Methods: POST, GET, OPTIONS");
4 header("Access-Control-Allow-Headers: *");
5
6 if ($_SERVER['REQUEST_METHOD'] === 'POST') {
7     $title = $_POST['title'] ?? '';
8     $content = $_POST['content'] ?? '';
9     $imageError = $_FILES['image']['error'];
10
11     if (!empty($title) && !empty($content) && $imageError === 0) {
12         $host = 'localhost';
13         $user = 'root';
14         $password = 'root';
15         $database = 'userdata';
16
17         $db = mysqli_connect($host, $user, $password, $database);
18
19         if (!$db) {
20             die("Connection failed: " . mysqli_connect_error());
21         }
22
23         $title = mysqli_real_escape_string($db, $title);
24         $content = mysqli_real_escape_string($db, $content);
25         $imageName = $_FILES['image']['name'];
26         $imageTmpName = $_FILES['image']['tmp_name'];
27
28         $uploadDirectory = 'uploads/';
29         $destination = $uploadDirectory . $imageName;
30
31         if (move_uploaded_file($imageTmpName, $destination)) {
32             $query = "INSERT INTO blog (title, content, image_name, image_path) VALUES ('$title', '$content', '$imageName', '$destination')";
33             $result = mysqli_query($db, $query);
34
35             if ($result) {
36                 echo json_encode(array('message' => 'Data inserted successfully.));
37             } else {
38                 echo json_encode(array('error' => 'Failed to insert data.));
39             }
40         } else {
41             echo json_encode(array('error' => 'Failed to move uploaded file.));
42         }
43
44         mysqli_close($db);
45     } else {
46         echo json_encode(array('error' => 'Title, content, and image are required.));
47     }
48 } else {
49     echo json_encode(array('error' => 'Invalid request.));
50 }
51
```

PROBLEMS 83 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
User is not authenticated
User is authenticated
Application finished.
(base) admin@adminins-MacBook-Pro-Peter htdocs % tree
zsh: command not found: tree
(base) admin@adminins-MacBook-Pro-Peter htdocs % flutter run -d chrome
```

Flutter dashboard i dart programmering med Uri.parse() til API

```

lib > dashboard.dart > ...
1  import 'dart:convert';
2  import 'dart:typed_data';
3  import 'package:flutter/material.dart';
4  import 'package:http/http.dart' as http;
5  import 'package:image_picker/image_picker.dart';
6  import 'package:shared_preferences/shared_preferences.dart';
7  import 'login.dart';
8
9  class Dashboard extends StatefulWidget {
10     @override
11     _DashboardState createState() => _DashboardState();
12 }
13
14 class _DashboardState extends State<Dashboard> {
15     TextEditingController _titleController = TextEditingController();
16     TextEditingController _contentController = TextEditingController();
17     late String _image = ''; // Initialize _image with an empty string
18
19     @override
20     void initState() {
21         super.initState();
22         checkAuthentication();
23     }
24
25     void checkAuthentication() async {
26         SharedPreferences prefs = await SharedPreferences.getInstance();
27         String? token = prefs.getString('authToken');
28
29         if (token == null || token.isEmpty) {
30             Navigator.pushReplacement(
31                 context,
32                 MaterialPageRoute(builder: (context) => LoginScreen()),
33             );
34         }
35     }
36
37     @override
38     Widget build(BuildContext context) {
39         return Scaffold(
40             appBar: AppBar(
41                 title: Text('Dashboard'),
42             ), // AppBar
43             body: SingleChildScrollView(
44                 child: Column(
45                     crossAxisAlignment: CrossAxisAlignment.stretch,
46                     children: [
47                         TextFormField(
48                             controller: _titleController,
49                             decoration: InputDecoration(
50                                 labelText: 'Title',
51                             ), // InputDecoration
52                         ), // TextFormField
53                         SizedBox(height: 20),
54                         TextFormField(
55                             controller: _contentController,
56                             decoration: InputDecoration(
57                                 labelText: 'Content',
58                             ), // InputDecoration
59                             maxLines: null,
60                         ), // TextFormField
61                         SizedBox(height: 20),
62                         ElevatedButton(
63                             onPressed: getImage,
64                             child: Text('Pick Image'),
65                         ), // ElevatedButton
66                         SizedBox(height: 20),
67                         _image.isEmpty
68                             ? Image.memory(Uint8List.fromList(base64Decode(_image)))
69                             : Text('No image selected.'),
70                         SizedBox(height: 20),
71                         ElevatedButton(
72                             onPressed: submitData,
73                             child: Text('Submit'),
74                         ), // ElevatedButton
75                     ],
76                 ), // Column
77             ), // SingleChildScrollView
78         ); // Scaffold
79     }
80
    
```

```

80
81     Future<void> submitData() async {
82         final title = _titleController.text;
83         final content = _contentController.text;
84
85         final url = Uri.parse("http://localhost:8888/blog.php");
86
87         try {
88             var request = http.MultipartRequest('POST', url)
89                 ..fields['title'] = title
90                 ..fields['content'] = content;
91
92             if (!_image.isEmpty) {
93                 // Save image to server
94                 var response = await http.post(
95                     Uri.parse('http://localhost:8888/blog.php'),
96                     body: {'image': _image},
97                 );
98
99                 if (response.statusCode == 200) {
100                     // Image saved successfully, get the image URL
101                     var imageUrl = response.body;
102
103                     // Add image URL to the request fields
104                     request.fields['image_url'] = imageUrl;
105                 } else {
106                     throw Exception('Failed to save image to server!');
107                 }
108             }
109
110             var response = await request.send();
111             if (response.statusCode == 200) {
112                 ScaffoldMessenger.of(context).showSnackBar(
113                     SnackBar(content: Text('Data saved successfully')),
114                 );
115                 _titleController.clear();
116                 _contentController.clear();
117                 setState(() {
118                     _image = '';
119                 });
120             } else {
121                 ScaffoldMessenger.of(context).showSnackBar(
122                     SnackBar(content: Text('Failed to save data')),
123                 );
124             }
125         } catch (error) {
126             print('Error: $error');
127         }
128     }
129
130     Future<void> getImage() async {
131         final picker = ImagePicker();
132         final pickedFile = await picker.pickImage(source: ImageSource.gallery);
133
134         if (pickedFile != null) {
135             final bytes = await pickedFile.readAsBytes();
136             setState(() {
137                 _image = base64Encode(bytes);
138             });
139         } else {
140             print('No image selected.');
```