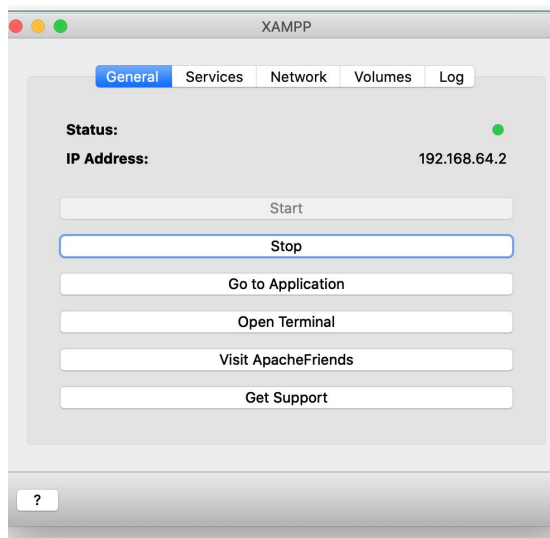
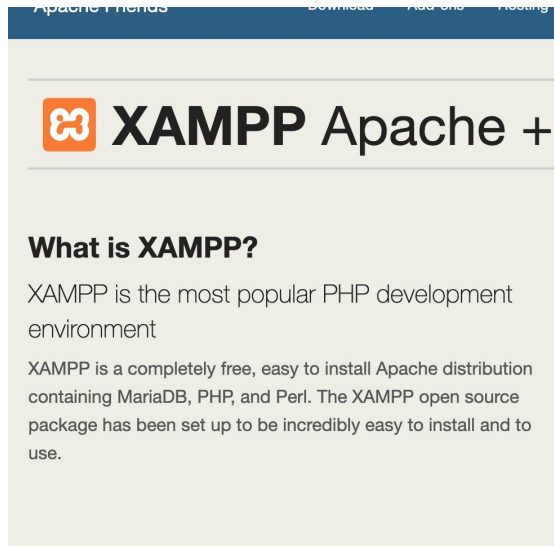
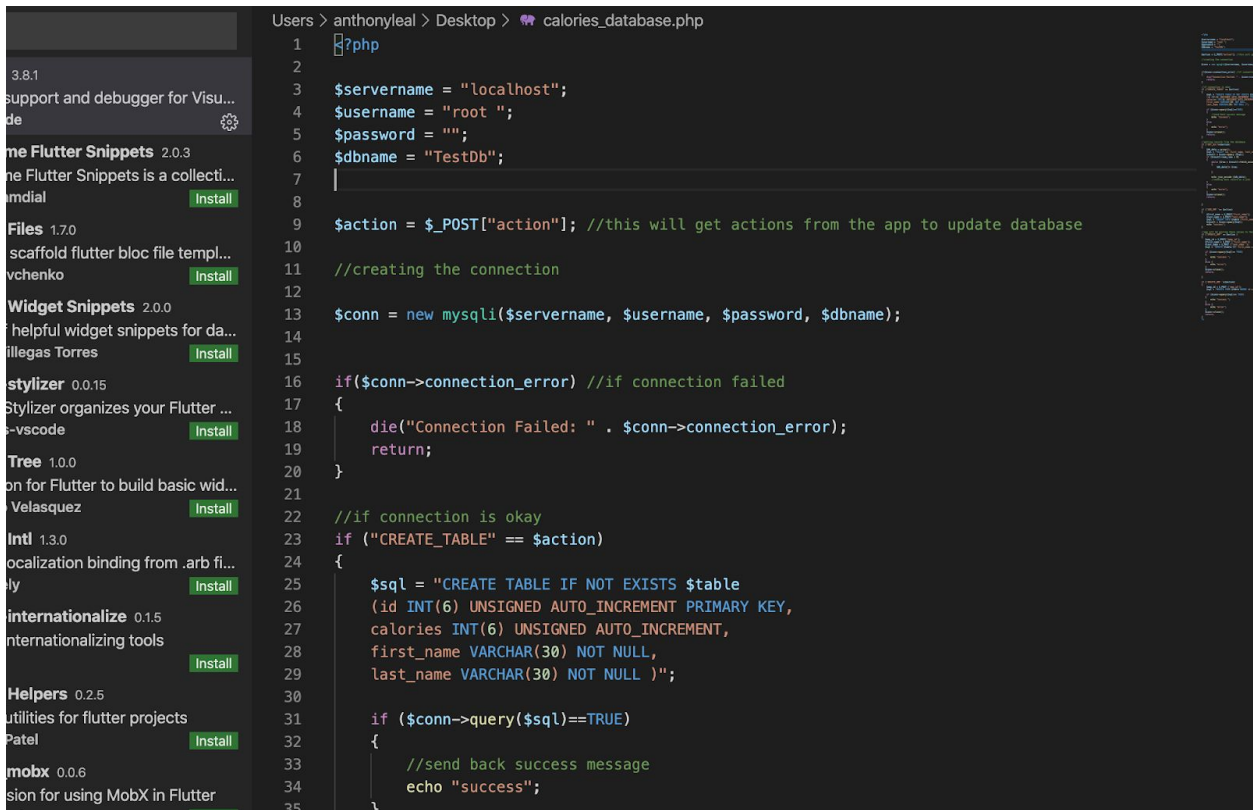
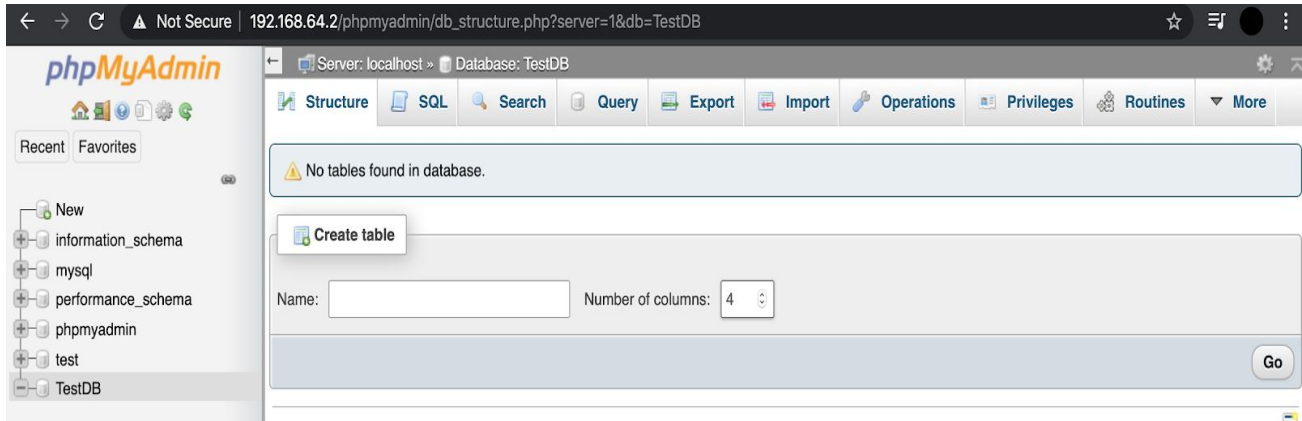


Back End Database (Anthony Leal)





```

6  class FoodDatabase {
7      // DB info
8      static final _databaseName = "food.db";
9      static final _databaseVersion = 1;
10     static final table_name = 'food';
11
12     // DB columns
13     static final columnId = '_id';
14     static final columnName = 'name';
15     static final columnCalories = 'calorie';
16
17     // Singleton ~~ only one instance
18     FoodDatabase._privateConstructor();
19     static final FoodDatabase instance = FoodDatabase._privateConstructor();
20
21     // Database reference
22     static Database _database;
23     Future<Database> get database async {
24         if (_database != null) return _database;
25         // initialize database if it is not created yet
26         _database = await _initDatabase();
27         return _database;
28     }
29
30     // Opens/Creates database
31     _initDatabase() async {
32         Directory documentsDirectory = await getApplicationDocumentsDirectory();
33         String path = join(documentsDirectory.path, _databaseName);
34         return await openDatabase(path,
35             version: _databaseVersion,
36             onCreate: _onCreate);
37     }
38
39     // SQL code to create the database table_name
40     Future _onCreate(Database db, int version) async {
41         await db.execute('''
42             CREATE TABLE $table_name (
43                 $columnId INTEGER PRIMARY KEY,
44                 $columnName TEXT NOT NULL,
45                 $columnCalories INTEGER NOT NULL
46             )
47         ''');
48     }
49 }

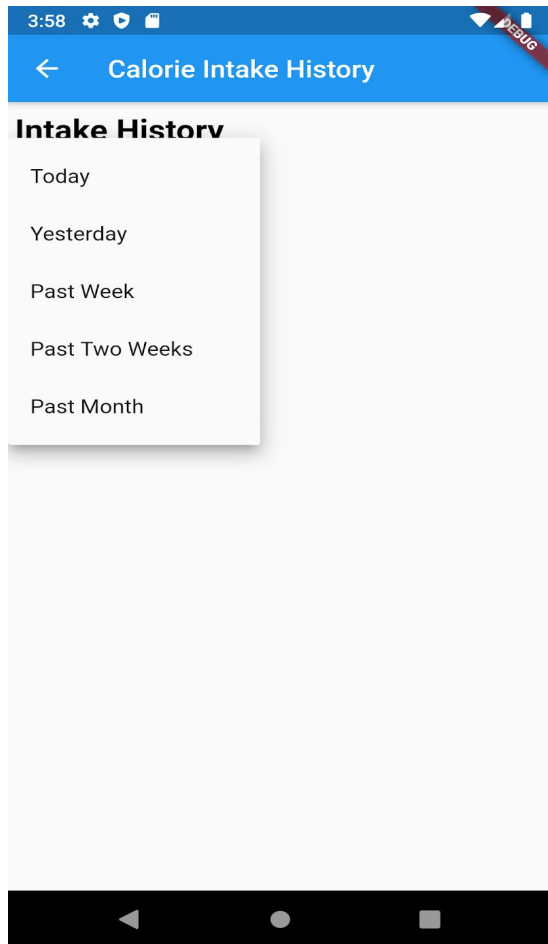
```

Intake History Page

Code forming the page and components:

```
void openFoodHistoryPage(BuildContext context) {
  final _historyPeriod = ['Today', 'Yesterday', 'Past Week', 'Past Two Weeks', 'Past Month'];
  String currentPeriodSelected = _historyPeriod[0];

  Navigator.push(context, MaterialPageRoute(
    builder: (BuildContext context) {
      return Scaffold(
        appBar: AppBar(
          title: const Text('Calorie Intake History'),
        ),
        body: Column(
          children: <Widget> [
            Container(
              child: Text('Intake History',
                style: TextStyle(
                  fontSize: 24,
                  fontWeight: FontWeight.bold,
                  decoration: TextDecoration.underline,
                  color: Colors.black
                )
              ),
            ),
            padding: EdgeInsets.only(top: 10, left: 5)
          ],
          DropdownButton<String>(
            items: _historyPeriod.map( (String menuItem) {
              return DropdownMenuItem<String> (
                value: menuItem,
                child: Text(menuItem)
              );
            } ).toList(),
            onChanged: (String newPeriodSelected) {
              //To be implemented in a stateful class
              //in a future iteration
            },
            value: currentPeriodSelected
          )
        ]
      ),
    );
}
```



Calorie Log Page

- Started with initial mockups of main page
- Used mockups to design main page of app
- Screen allows users to add and delete items from today's calorie log
- Encompasses basic functionality of the app, and is working correctly and usable right now

Main Page Mockup 2

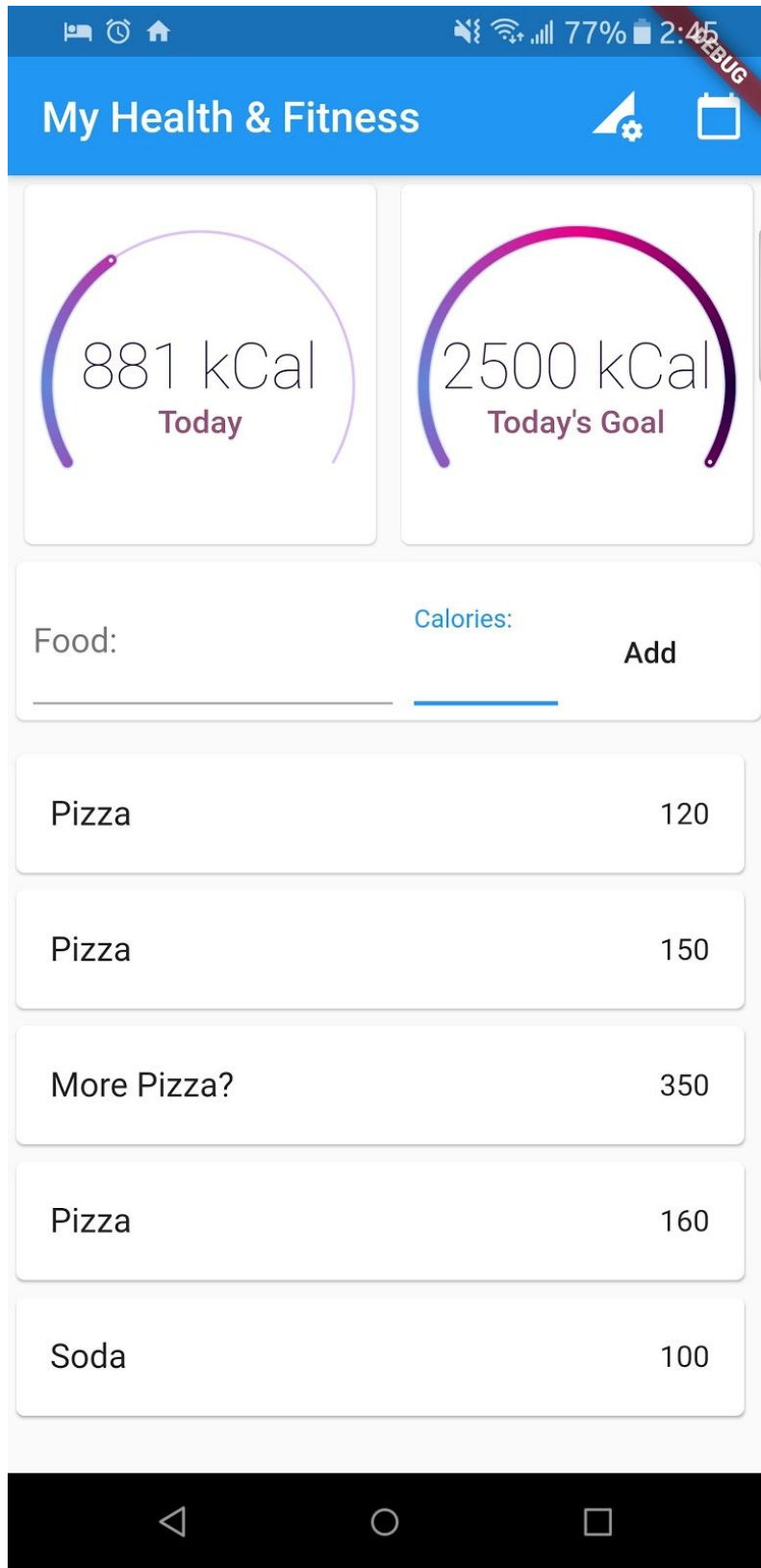
800 Cal 2-5	
<u>Pizza</u>	<u>100</u> Add
.....	
Soda	100
Sandwich	150

Main Page Mockup 1

1400 / 2000

Toast	200
Orange Juice	100
Sandwich	150
Soda	90
✓...	
Pizza	100

Add



Splash Screen

- Only shows first time user opens app and submits information using boolean value
- Input user information such as name, age, height and weight
- Planned to be shared with a future user information widget within the app that allows the user to edit information

```

1 import 'package:flutter/material.dart';
2 import 'package:shared_preferences/shared_preferences.dart';
3 import 'package:reborn_next_job02/ui/LoginScreen.dart';
4 import 'package:reborn_next_job02/ui/splashScreen.dart';
5
6 class SplashState extends StatefulWidget {
7
8
9     SplashState({Key key}) : super(key: key);
10
11     @override
12     _SplashState createState() => _SplashState();
13 }
14
15 class _SplashState extends State<SplashState> {
16
17     Future checkFirstSeen() async {
18         SharedPreferences prefs = await SharedPreferences.getInstance();
19         bool _seen = (prefs.getBool('seen') ?? false);
20         if (_seen) {
21             Navigator.of(context).pushReplacement(
22                 new MaterialPageRoute(builder: (context) => new LoginScreen()));
23         } else {
24             prefs.setBool('seen', true);
25             Navigator.of(context).pushReplacement([
26                 new MaterialPageRoute(builder: (context) => new urlScreen());
27             ]);
28         }
29     }
30
31     final _formKey = GlobalKey<FormState>();
32
33     @override
34     Widget build(BuildContext context) {
35         return Form(
36             key: _formKey,
37             child: Column(
38                 crossAxisAlignment: CrossAxisAlignment.start,
39                 children: <Widget>[
40                     TextFormField(

```

```
37 children: <Widget>[
38   TextFormField(
39     decoration: const InputDecoration(
40       hintText: 'Enter your name',
41     ),
42     validator: (name) {
43       if (name.isEmpty) {
44         return 'Please enter a valid name';
45       }
46       return null;
47     },
48   ),
49   TextFormField(
50     decoration: const InputDecoration(
51       hintText: 'Enter your age',
52     ),
53     validator: (age) {
54       if (age.isEmpty) {
55         return 'Please enter a valid age';
56       }
57       return null;
58     },
59   ),
60   TextFormField(
61     decoration: const InputDecoration(
62       hintText: 'Enter your height',
63     ),
64     validator: (height) {
65       if (height.isEmpty) {
66         return 'Please enter a valid height';
67       }
68       return null;
69     },
70   ),
71   TextFormField(
72     decoration: const InputDecoration(
73       hintText: 'Enter your weight',
74   ),
```

```

78     }
79     return null;
80   },
81   ),
82   Padding(
83     padding: const EdgeInsets.symmetric(vertical: 16.0),
84     child: RaisedButton(
85       onPressed: () {
86         if (_formKey.currentState.validate()) {
87           }
88       },
89       child: Text('Submit'),
90     ),
91   ),
92 ],
93 ),
94 );
95 }
96 }

```

This example shows a Form with one TextFormField to enter an email address and a RaisedButton to submit the form. A GlobalKey is used here to identify the Form and validate input.

Enter your email

Submit