LONDON METROPOLITAN UNIVERSITY

Module Code & Module Title

CS6PO5 Final Year Project

Assessment Weightage & Type

25% Project Interim Report

2019-20 Autumn

Student Name:

London Met ID:

College ID:

Assignment Due Date:

Assignment Submission Date:

Internal Supervisor:

External Supervisor:

Word Count:

I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded.

Abstract

As of January 2020, the project has reached a halfway mark for the development phase. This interim report serves to document the project progress up to date. The report starts with mentions of technology used to create the system and brief summary of exploration and comparison of similar applications already available. The major portion catalogues the progress made since initiation till date with appropriate screenshots to prove the advancement in development.

The progress is then compared while describing any deviation from the original plans. The planned Gantt chart is available in the appendix. The report also mentions the major tasks yet to be accomplished and plans to complete them in accordance with the work table.

Contents

Figures

2020 January 7th

Introduction

The project software to be documented in this report is a flutter android application and a web application based on Laravel. The two applications work in tandem to create a food ordering system designed specifically for the college canteen.

The original intention in creating such application is to organize the meals and incorporate the college cafeteria into ever expanding sector of information technology such that its users need to spend as little amount of time as possible ordering and waiting for the food. In Addition, the system provides alternative payment methods for those critical moments such as when wallets are forgotten back home.

The system will have dynamic pages capable of automatically reloading according to the changes to the database. This allows the staff to be instantaneous notified when new orders are place. The staff can in turn the users when the meals are ready through application notification. The system will also provide suggestions to the clients based on their past activities where as statics and order predictions for the staff.

one of the most disappointing features prevalent in most of the application is the email verification that takes an eternity to verify, and until the customers can neither order nor view the menu. This system will not face such problems for a simple fact that the all users are also somewhat related to the college itself, and all users are created by the administrator beforehand. This will not require a verification to be sent.

The ongoing survey conducted locally within the college suggests that the students spend more time than they wished waiting for their meals to be prepared or deciding what to order. Having an application to notify user when the meals are ready means not having to wait for your number to be called out. The Staff can also better manage time and resources when orders are sorted and categorized.

2020 January 7th

Background

Flutter is a cross platform framework for dart programming language and can be used to create fluid mobile as well as desktop application. The framework can render and refresh the pages at 60 frames per second to produce a dynamic and seamless _____

Similarly, Laravel is also a framework package for PHP with a large online community and libraries. When used with Eloquent ORM, the powerful combination simplifies a lot of programming and routings between pages consequently speeding up page loads.

The intended Software being built with integration between the two frameworks has a very specific target of users thus similar applications are rare. But then again, in some ways it can be compared to generic food ordering app out there. As such some were taken as research models, namely the food-mandu app and foodmario, two of the most prevalent local apps

Firstly, the Foodmandu app, disregarding the option to choose the origin restaurant of the food, most of the functionalities are similar. The user selects a food item and it moves to a “basket” as opposed to a ”tray” in our case.

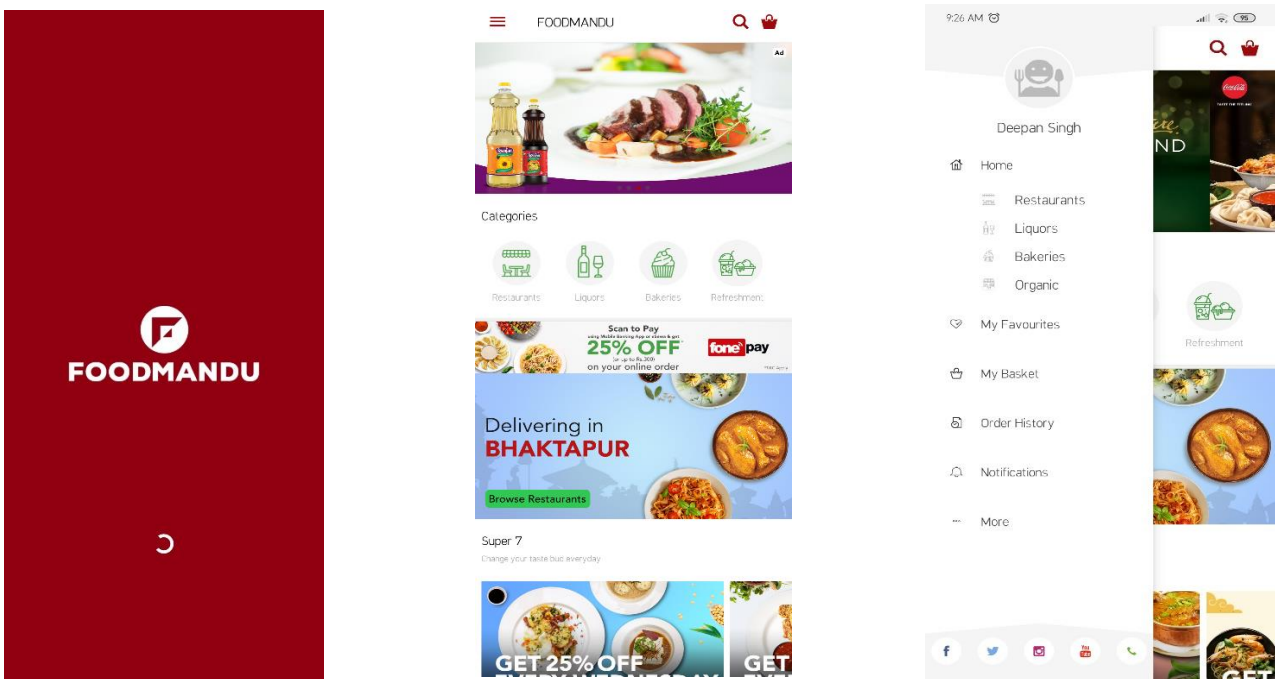


Figure 1: Foodmandu app

(From left: splash screen, home page, option panel)

However, If the user wanted to check or correct their orders, they would have to move to a different page entirely and return back to the original page to order more. In Contrast, the tray will be available at any given moment in out intended application. The user can add to the item by tapping on it or remove the item by holding the item directly from the side panel.

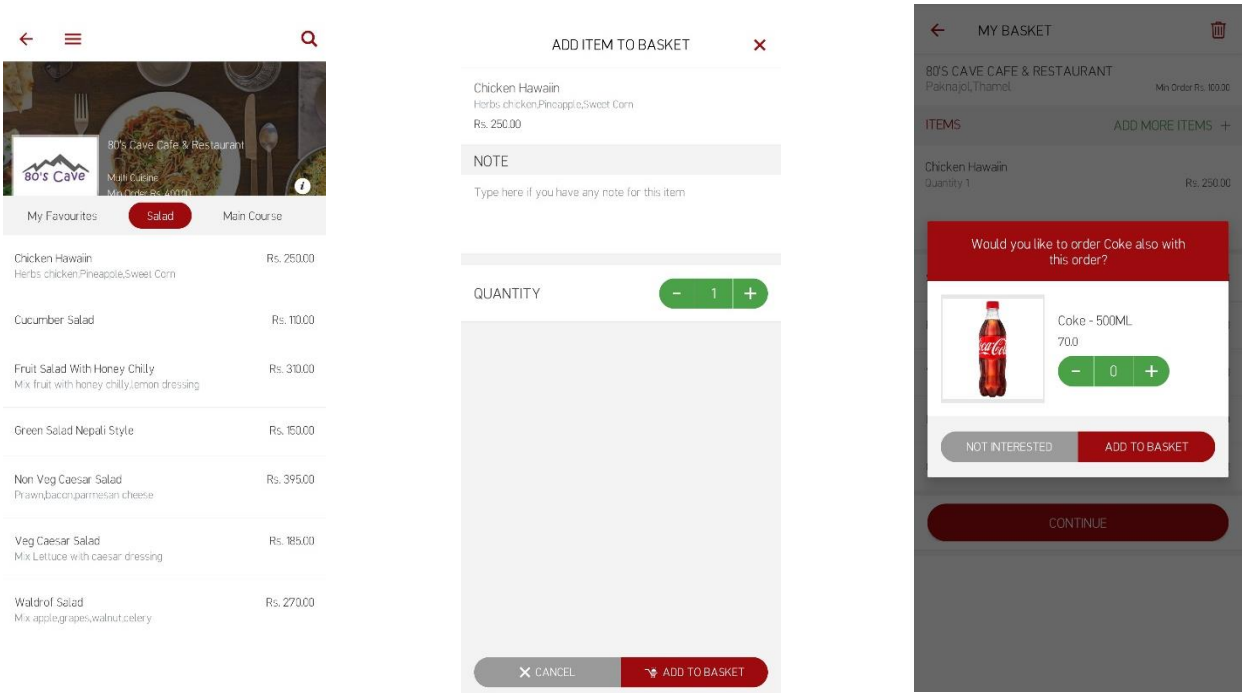


Figure 2: Foodmandu app
(from left: menu, basket, suggestions)

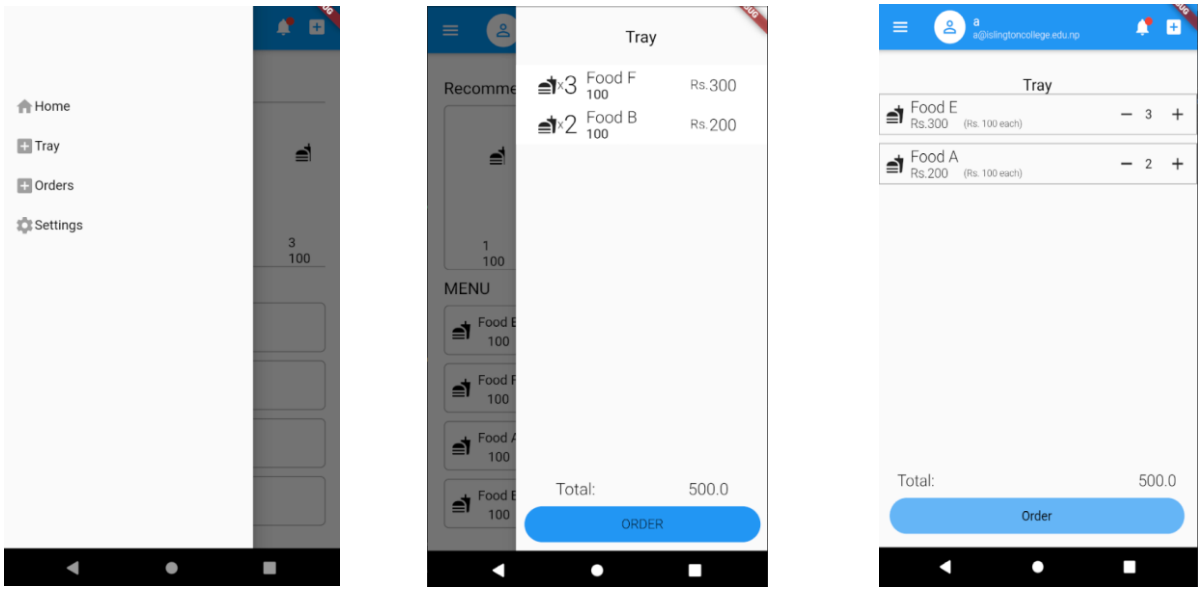


Figure 3: app in development
(from left: option panel, tray panel, tray page)

Development to date

In terms of milestones, Entity diagrams have been created representing the necessary database tables finalized through repetitive iteration of trial and error. According to the diagrams, the databases were created before generating a Laravel project. A Data Model was created for each table in the database. APIs were inserted to access the database tables. An initial iteration of the system, albeit without a designed User interface, was coded capable of using the APIs to request and receive data.

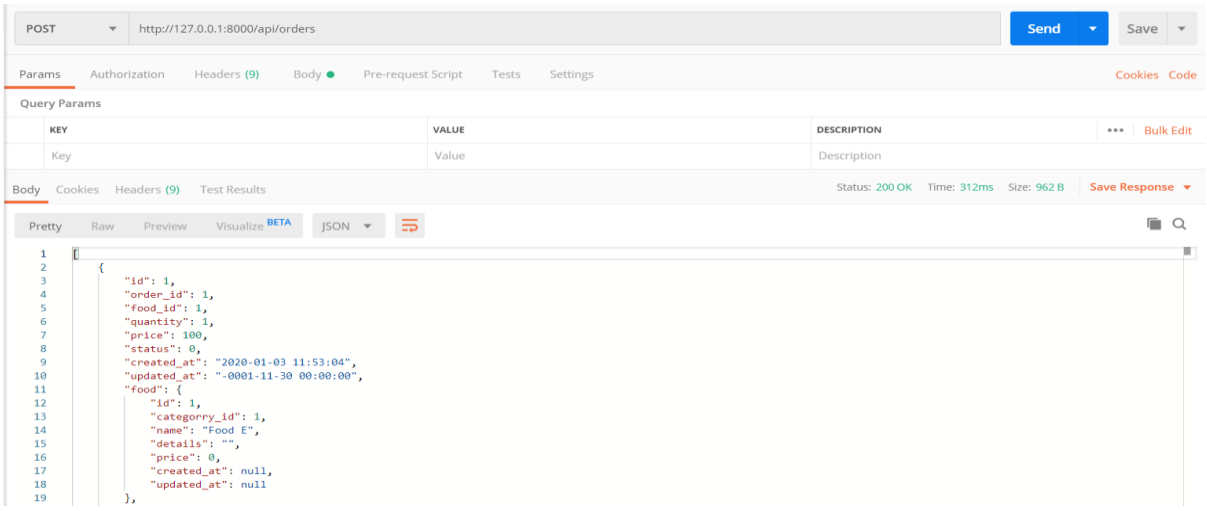


Figure 4: API demo in Postman - orders

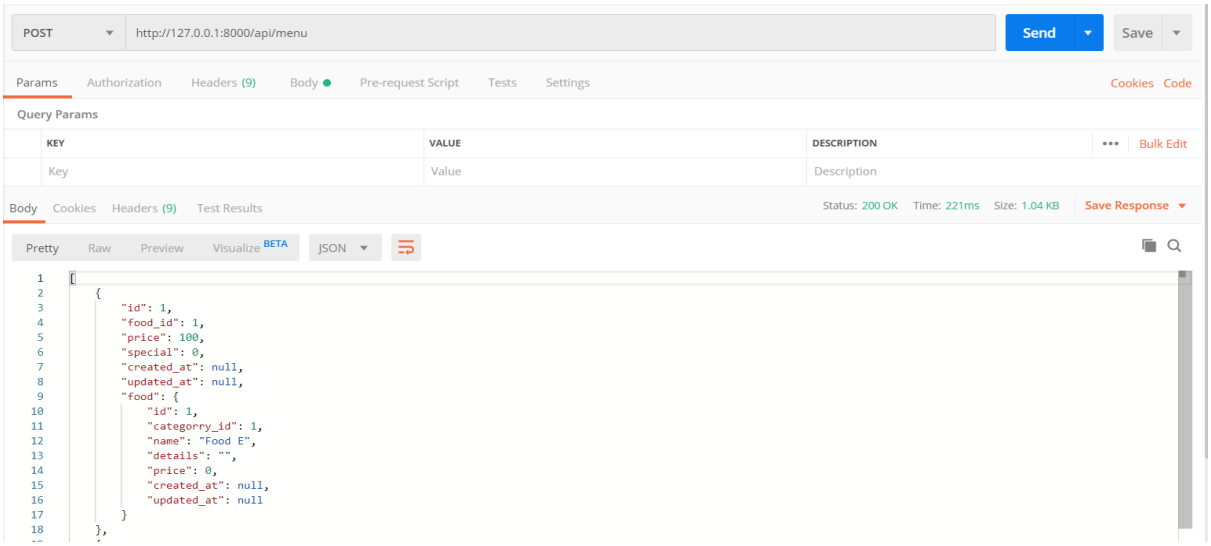


Figure 5: API demo in Postman - menu

The API themselves can be tested through the URLs, but a Postman software is used to manage the requests and visualize the data for better understanding. The server sends and receives data in json format as shown in above screenshots which the application parses to display in proper user interface.

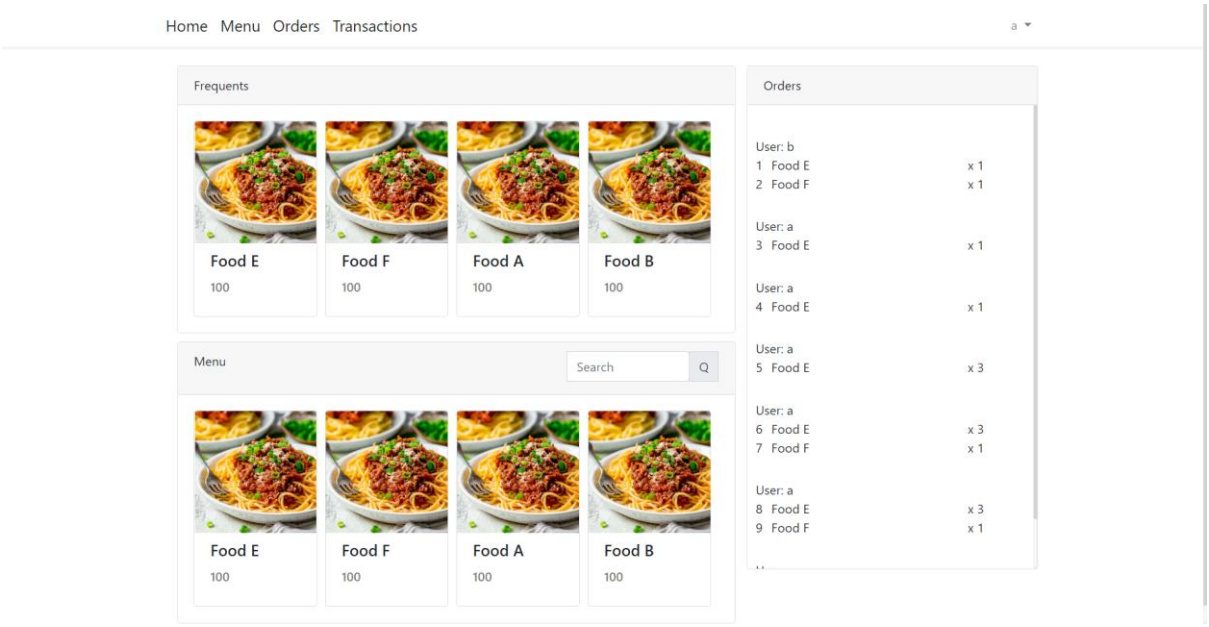


Figure 6: web home page

As depicted in the above picture, the web application till date can show available menu and meals ordered from the mobile counterpart. The top middle section lists the most ordered items to make them easily accessible for the staff when it is next ordered. The right side panel shows all orders and should be able to update dynamically when a new order is places. The bottom sections has menu for the day. Rather than showing all food items in the database, it shows only a select few as pre defined by the staff.

The initial iteration of the mobile application is capable of some basic features like displaying the available menu for the day then collecting the user selected items in a “tray” to push the order list to the server via the numerous API provided. The application is authenticated by a token dynamically generated at each login. The application can store this token as a shared preference to remember to user for next login. At each start of the application, the token is referenced with the server with login API and when successful get the menu also with an API.

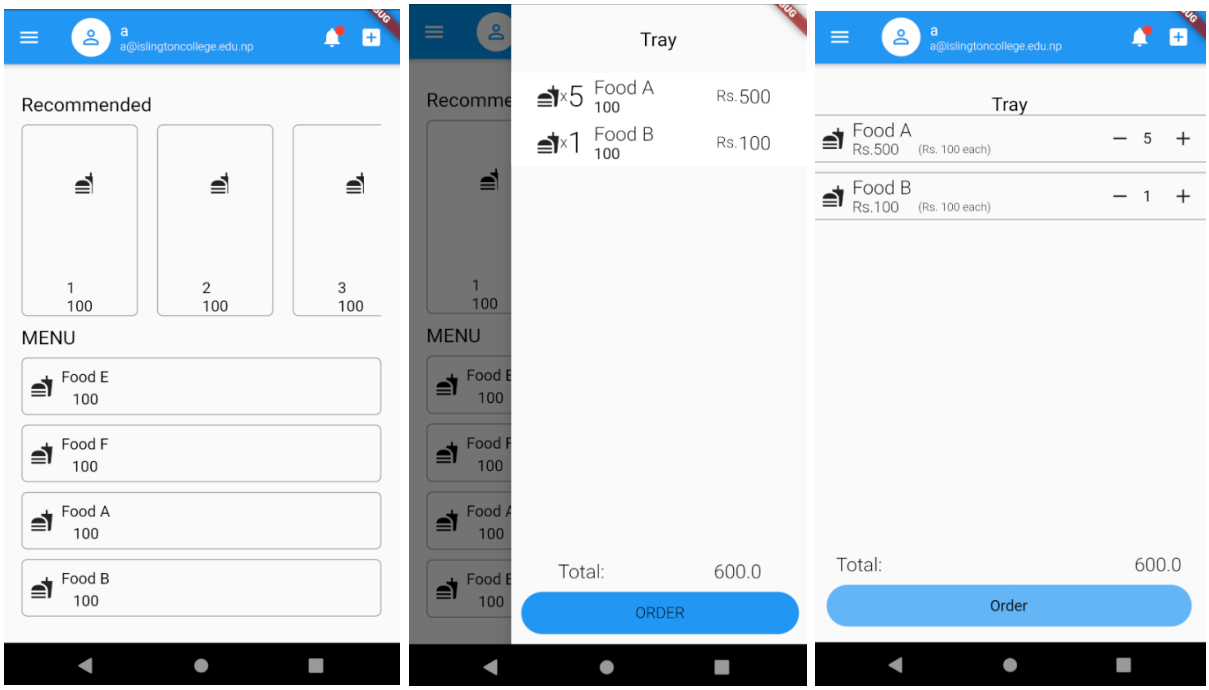


Figure 7: App in development(from left: home, side panel, tray)

The top section of the first page shows the recommended items which is generated based on past user activity, the staff suggestions or the items cancelled by other users. The lower section on the same page shows the menu of the day. When user taps any of these they will be added to the tray that can be accessed from the right-side panel. Further tapping the item on the tray adds to the quantity in the item. More detailed controls can be found in the tray page itself. When the meals are ordered, the list of items are sent to the server through another API. Each API is protected by the previously mentioned token which can also be used to identify the requesting user.

The application can also receive notification when the meals are ready with the implementation of google cloud messaging services. User can keep track of the progress of the meal and alerted when ready. The users can navigate through pages from the options panel at the left side the screen which can be toggled by a swipe right gesture on screen or the button on the top left side of the screen.

As of the moment, vue.js is being implemented with Ajax as a mediator to reload pages dynamically with the database. However, there are some issues where the pages do not reload or crashes entirely. The transition of a page can not be depicted in a static document however relevant code can be found in the appendix.

2020 January 7th

Analysis of progress

In comparison to the time allocation and Gant chart in the proposal, the project is slightly lagging behind. Although it might be wrong to attribute the tardiness of the project to other modules and obligations like interns and assignment, I hope to catch up and more in the free time I get after the completion of my intern. The deviation from the original plans are listed below. Gantt chart is available in the Appendix for comparison.

The project was initiated in October 2019. As planned, the requirements were gathered and use cases were drawn numerously till satisfaction. Once finalized, the diagrams were expanded into relational diagrams and Data modules. The required number of UI pages were estimated, and a basic layout was planned.

November 2019 started exactly as planned, most of data planning was laid out and a general out line and wireframes were completed. However not as much was done for web application other than the API. The middle ware authentication was implemented to verify each request through these API whereas a general auth was implemented for web login.

December 2019 should have started with half way completion of initial iteration of the mobile application, but major pages only started in December. Pages to show the menu and orders are developed in terms of functionality in mobile as well as the web application.

By the end of 2019, an initial iteration for the system has been finished. Although the mobile aspect has been somewhat resolved, much is lacking in the web portion. However, the Eloquence and hot reloading aspect has been somewhat implemented at an earlier iteration.

Future work

If the project is to be completed as per the allotted time two more iterations are to be completed by the end of February 2020. The second iteration will include a viable User interface design for both flutter application as well as the Laravel web app. The Menu in the mobile application will be categorized in to tabs for easier findings of the items. Similarly, a search feature is to be added to the web menu section. Although the application can receive notification, the module to send has yet to be added in the web section. One Signal libraries will be integrated into the system in order to send notification form the web.

The third iteration is expected to implement some extra features like delegation of cancelled orders and offline data for faster page loads. The option to sort orders according to user or meals can be included into the system which will be especially helpful for the staff members.

In March 2020, a prediction algorithm will be integrated initially with dummy data and should be able to predict number of items that might be ordered in any particular day and suggest items to the users based on their past activities.

Bibliography

Flutter Community, 2019. *Flutter Documentation.* [Online]
Available at: https://flutter.dev/docs
[Accessed 12 October 2019].

FoodMandu pvt ltd, 2018. *FoodMandu.* [Online]
Available at: https://foodmandu.com/
[Accessed Oct 15 2019].

FoodMario, 2019. *FoodMario Customer App.* Kathmandu: Food Mario.

Gore, A., 2018. *Vue.js Developers - The Ultimate Vue.js & Laravel CRUD Tutorial.* [Online]
Available at: https://vuejsdevelopers.com/2018/02/05/vue-laravel-crud/
[Accessed 28 December 2019].

Laravel LLC, 2019. *Laravel Documentation.* [Online]
Available at: https://laravel.com/docs/6.x
[Accessed 12 October 2019].

MealNepal, 2019. *Online food delivery in Nepal: Top 5 services | Meal Nepal.* [Online]
Available at: http://mealnepal.com/online-food-delivery-in-nepal-top-5-services/
[Accessed 15 Oct 2019].

PUSHER inc, 2019. *PUSHER - HANDLE HTTP REQUESTS IN A LARAVEL VUE.JS APP WITH AXIOS.* [Online]
Available at: https://pusher.com/tutorials/laravel-vue-axios
[Accessed 3 january 2020].

w3school, 2019. *AJAX Introduction | w3school.* [Online]
Available at: https://www.w3schools.com/xml/ajax_intro.asp
[Accessed 27 12 2019].

Appendices

Gantt Chart

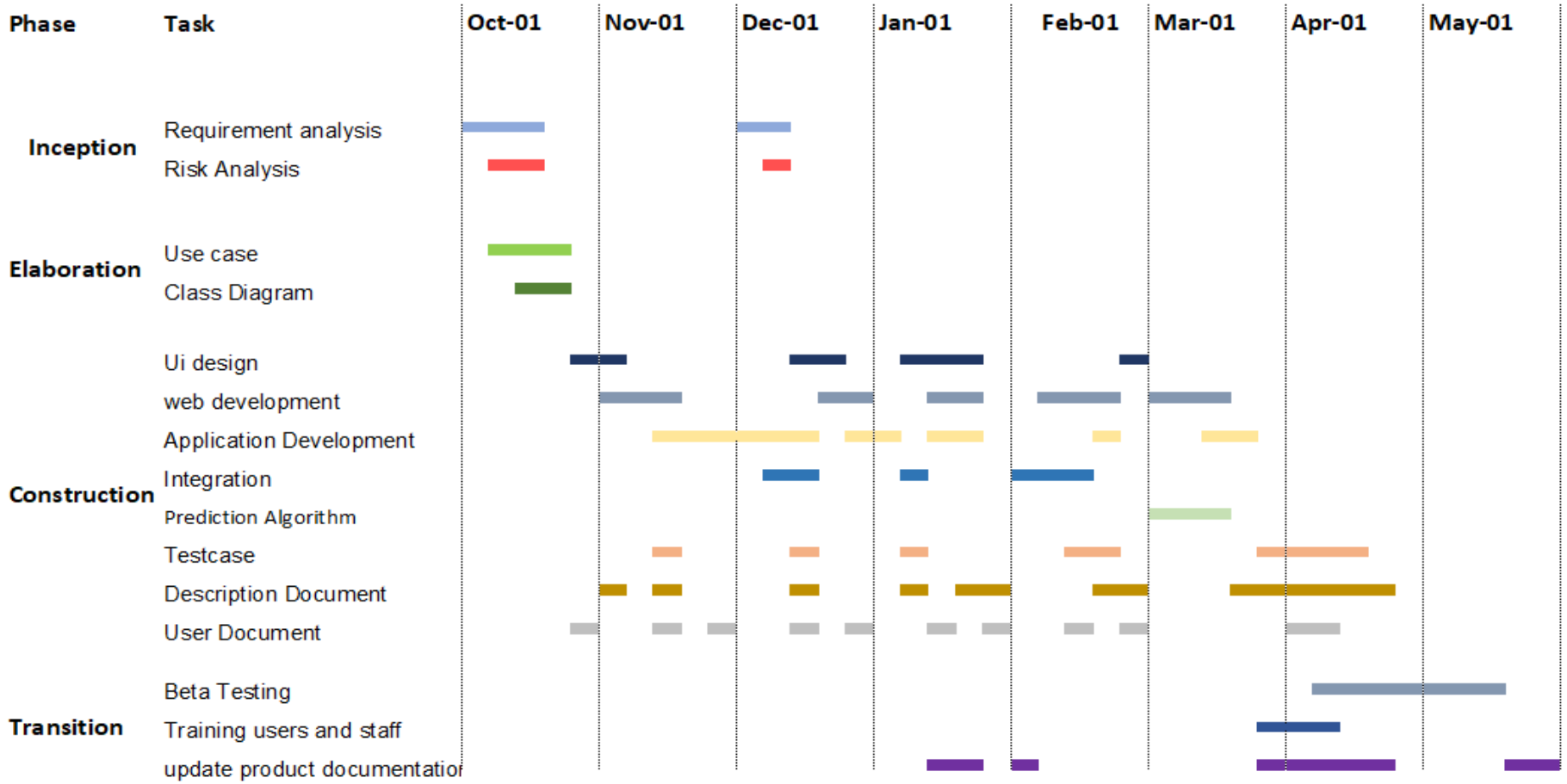


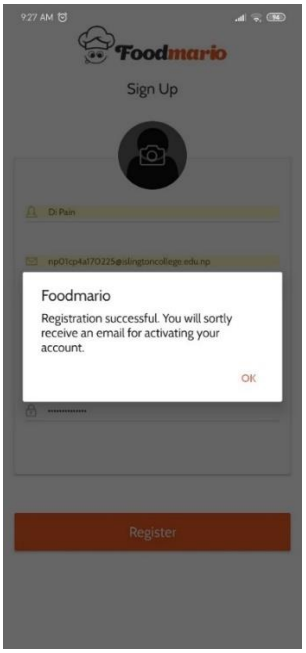
Figure 8: Gantt Chart

2020 January 7th

Survey conducted

https://docs.google.com/forms/d/1FqvnvpJm7acxwjztEAuw-AUGVcrct1GjI1pUcQE_3Zk

Other Researched Applications



Food Mario has a decent user interface but somewhat slow service. In comparison to foodmandu app, the altogether load was slower and the verification for registration took 2 days. The meals could be paid for with online medium or payment on delivery system. Since these applications also handled delivery the require constant stream of current location to work properly.

Git Repository

Flutter: https://github.com/DiPain/fyp

Laravel: https://github.com/DiPain/fypBackend

Major Codes

Navi.dart

```
class Navi extends StatefulWidget {
  Navi({Key key, this.title, this.index=1}) : super(key: key);
  final String title;
  int index;

  @override
  _NaviState createState() => _NaviState( this.index);
}

class _NaviState extends State<Navi> {
  int index;
  _NaviState(this.index){
    print('=====Navi=====');
    print(DbServer.token);
  }

  @override
  Widget build(BuildContext context) {
    DbServer().getProfile().then((value){
      if(value==null){
        Fluttertoast.showToast(
          msg: "No Internet Connection",
          gravity: ToastGravity.BOTTOM,
        );
      }
    });
    List pages = [
      ['Home', Icons.home, Home(),],
      ['Tray', Icons.add_box, Tray(),],
      ['Orders', Icons.add_box, Orders(),],
      ['Settings', Icons.settings, Settings(),],
    ];

    final GlobalKey<ScaffoldState> _scafKey = new GlobalKey<ScaffoldState>();

    return ChangeNotifierProvider(
      builder: (_)=>NaviBack(),
      child: Consumer<NaviBack>(
        builder: (context, naviBack,_){
        final naviBack = Provider.of<NaviBack>(context);
```

```
        return Scaffold(
          key: _scafKey,
          drawer: Drawer(
            child: Container(
              child: ListView.builder(
                itemCount: pages.length+1,

                itemBuilder: (context, i){
                  if(i==0){
                    return Container(
                      height: 100,
                      width: 100,
                      child: SizedBox(),
                    );
                  }else{
                    return Container(
                      margin: EdgeInsets.only(left: 0),
                      child: FlatButton(
                        child: Row(children: <Widget>[
                          Icon(pages[i-1][1], color: Colors.grey,),
                          Txt(pages[i-1][0]),
                          Expanded(
                            child: SizedBox(),
                          )
                          ]),
                        onPressed: (){
                          naviBack.changeTab(i-1);
                          print('should change');
                          Navigator.popAndPushNamed(context, '/newroute');
                        },
                      ),
                    );
                  }
                },
              ),
            ),
          ),
          appBar: AppBar(
```

```
                padding: EdgeInsets.all(0),
                margin: EdgeInsets.all(0),
                child: Row(
                  crossAxisAlignment: CrossAxisAlignment.center,
                  children: <Widget>[
                  Container(
                    width: 40,
                    height: 40,
                    decoration: BoxDecor(
                      radius: 22
                    ),
                    child: Icon(Icons.person_outline, color: Colors.blue,),),
                    SizedBox(width: 10,),
                  Column(
                    crossAxisAlignment: CrossAxisAlignment.start,
                    mainAxisAlignment: MainAxisAlignment.center,
                    children: <Widget>[
                      Txt(DbServer.name,color: Colors.white,),
                      Txt(DbServer.email, fw: FontWeight.w300,size: 12,color:
Colors.white,),
                    ],
                  )
                ],),
              ),
              actions: <Widget>[
                Container(
                  width: 30,
                  child: FlatButton(
                    padding: EdgeInsets.all(0),
                    child: Stack(
                      children: <Widget>[
                        Icon(Icons.notifications, color: Colors.white,),
                        Container(
                          height: 10,
                          width: 10,
                          margin: EdgeInsets.only(left: 15),
                          decoration: BoxDecor(radius: 5, color: Colors.red, b
orderColor: Colors.transparent),
                        ),
                      ],
                    ),
                    onPressed: (){
                      // Navigator.push(context,
                      //   MaterialPageRoute(builder: (context)=>AllEvents()))
                    },
                  ),
                ),
```

```
            margin: EdgeInsets.only(right: 15,left: 10),
            child: FlatButton(
              padding: EdgeInsets.all(0),
              child: Icon(Icons.add_box, color: Colors.white,),
              onPressed: (){
                // Navigator.push(context,
                //    MaterialPageRoute(builder: (context)=>Tray()));
                _scafKey.currentState.openEndDrawer();
              },
            ),
          )
        ],
      ),

      endDrawer: Drawer(

        child: Container(
          child: ChangeNotifierProvider(
            builder: (_)=>TrayBack(),
            child: Consumer<TrayBack>(
              builder: (context, trayBack,_){
                final trayBack = Provider.of<TrayBack>(context);
                trayBack.getTotal();
                return Container(
                  child: Stack(
                    children: <Widget>[
                      ListView.builder(
                        itemCount: DbServer.tray.length+1,
                        itemBuilder: (context, intex){
                          if(intex==0){
                            return Container(
                              height: 70,
                              child: FlatButton(
                                onPressed: (){
                                  naviBack.changeTab(1);
                                  Navigator.popAndPushNamed(context, '/n

                                },
                                child: Txt('Tray', size: 20),
                              ),

                            );
                          }else{
                            return Container(
                              padding: EdgeInsets.only(right:20, left: 1


                                decoration: BoxDecor(radius: 0),
                                child: FlatButton(
                                  onPressed: (){
```

```
                                    naviBack.changeTab(naviBack.index);
                                    trayBack.addQty(intex-1, true);},
                                  child: Row(children: <Widget>[
                                    Icon(Icons.fastfood),
                                    Txt('x',fw: FontWeight.w300),
                                    Txt(DbServer.tray[intex-
1][1].toString(), size: 30,fw: FontWeight.w300,),
                                    SizedBox(width: 10,),
                                    Column(
                                      crossAxisAlignment: CrossAxisAlignme
nt.start,
                                    children: <Widget>[
                                      Txt(DbServer.menu[DbServer.tray[inte
x-1][0]]['food']['name'],fw: FontWeight.w300, size: 20,),
                                      Txt(DbServer.menu[DbServer.tray[inte
x-1][0]]['price'].toString() )
                                    ],),
                                    Expanded(child: SizedBox(),),
                                    Txt('Rs.', fw:FontWeight.w300),
                                    Txt((DbServer.menu[DbServer.tray[intex
-1][0]]['price']*DbServer.tray[intex-
1][1]).toString(), size: 20,fw:FontWeight.w300 ),

                                  ],),
                                )
                              );
                            }
                          },
                        ),
                        Positioned(
                          bottom: 10,
                          left: 10,
                          child: Column(
                            children: [
                              Row(
                                children: <Widget>[
                                  Txt('Total:', fw: FontWeight.w300,size:
20,),

                                  SizedBox(width: MediaQuery.of(context).s
ize.width-300,),

                                  Txt(trayBack.total.toString(),fw: FontWe
ight.w300,size: 20,)

                                ],
                              ),
                              SizedBox(height: 10,),
                              Container(
                                height: 44,
                                width: MediaQuery.of(context).size.width-
```

```
                                    radius: 22),
                                      child: Center(child: Txt('ORDER', fw: Fo
ntWeight.w300,))),
                                ]
                              ),
                            )
                          ]
                        ),
                      );
                    }
                  )
                )
              ),
            ),
            body: pages[naviBack.index][2]);
              // bottomNavigationBar: bottomNavBar(naviBack),
          }
        )
    );
  }
}
```

Home.dart

```
class Home extends StatefulWidget {
  Home({Key key,}) : super(key: key);

  @override
  _HomeState createState() => _HomeState();
}

class _HomeState extends State<Home> {
  bool loaded= false;

  @override
  Widget build(BuildContext context) {
    DbServer().getProfile().then((value){
      if(value==null){
        Fluttertoast.showToast(
          msg: "No Internet Connection",
          gravity: ToastGravity.BOTTOM,
        );
      }
    });

    return Scaffold(
      appBar: AppBar(
        actions: <Widget>[
          Container(
            child: FlatButton(
              child: Stack(
                children: <Widget>[
                  Icon(Icons.notifications, color: Colors.white,),
                  Container(
                    height: 10,
                    width: 10,
                    margin: EdgeInsets.only(left: 15),
                    decoration: BoxDecor(radius: 5, color: Colors.red, borderC
olor: Colors.transparent),
                  )
                ],
              ),
              onPressed: (){
                // Navigator.push(context,
                //   MaterialPageRoute(builder: (context)=>AllEvents()));
              },
            ),

          )
        ],
      ),
```

```
        padding: EdgeInsets.only(left: 20, right: 20, top: 20),
        child: Column(
          children: <Widget>[
            ChangeNotifierProvider(
              builder: (_)=>HomeBack(),
              child: Consumer<HomeBack>(
                builder: (context, logBack,_){
                  final homeBack = Provider.of<HomeBack>(context);
                  if(!loaded){
                    loaded=true;
                    homeBack.update();
                  }
                  return Column(
                    crossAxisAlignment: CrossAxisAlignment.start,
                    children:<Widget>[
                      SizedBox(height: 10,),
                      Txt('Recommended', size: 20,),
                      SizedBox(height: 10,),

                      SingleChildScrollView(
                        scrollDirection: Axis.horizontal,
                        child: Row(
                          children: homeBack.recommended,
                        ),
                      ),

                      SizedBox(height: 10,),
                      Txt('MENU', size: 20),
                      SizedBox(height: 10,),

                      Container(
                        height: MediaQuery.of(context).size.height-444,

                        child: SingleChildScrollView(
                          child: Column(
                            children: homeBack.menu,
                          ),
                        ),
                      )
                    ]
```

Home_back.dart

```
class HomeBack with ChangeNotifier{
  List<Widget> menu=[SizedBox()];
  List<Widget> recommended=[SizedBox()];
  update(){
    DbServer().getMenu().then((val){
      if(val!=null){
        if(val.length!='failed'){
          menu=[];
          for (var each in val){
            print(each);
            menu.add(Container(
              width: double.infinity,
              height:  60,
              decoration: BoxDecoration(
                border:Border.all(
                  color: Colors.grey,
                  width: 1,
                ),
                borderRadius: BorderRadius.all(Radius.circular(6)), ),
                child: FlatButton(
                  padding: EdgeInsets.all(0),
                  onPressed: (){
                    DbServer().addtoTray(each['foodId'], each['quantity']);
                    Fluttertoast.showToast(
                      msg: 'added foodId',
                      gravity: ToastGravity.CENTER
                    );
                   },
                  child: Row(
                    children: <Widget>[
                      Container(
                        height: 40,
                        width: 40,
                        child: Icon(Icons.fastfood)),
                      Column(
                        mainAxisAlignment: MainAxisAlignment.spaceAround,
                        children: <Widget>[
                          SizedBox(),
                          SizedBox(),
                          Txt(each['food_id'].toString()),
                          Txt(each['price'].toString()),
                          SizedBox(),
                      ],)
                    ],),
                ),
              ));
              menu.add(SizedBox(height: 10,));

              recommended.add(Container(
```

```
              width: 120,
              height:  200,
              decoration: BoxDecoration(
                border:Border.all(
                  color: Colors.grey,
                  width: 1,
                ),
                borderRadius: BorderRadius.all(Radius.circular(6)), ),
                child: FlatButton(
                  padding: EdgeInsets.all(0),
                  onPressed: (){
                    DbServer().addtoTray(each['foodId'], each['quantity']);
                    Fluttertoast.showToast(
                      msg: 'added foodId',
                      gravity: ToastGravity.CENTER
                    );
                   },
                  child: Stack(
                    children: <Widget>[
                      Positioned(
                        top: 40,
                        child:Container(
                        height: 40,
                        width: 40,
                        child: Icon(Icons.fastfood))),
                      Column(

                        mainAxisAlignment: MainAxisAlignment.end,
                        crossAxisAlignment: CrossAxisAlignment.start,
                        children: <Widget>[
                          SizedBox(),
                          SizedBox(),
                          Txt(each['food_id'].toString()),
                          Txt(each['price'].toString()),
                          SizedBox(),
                      ],)
                    ],),
                ),
              ));
              recommended.add(SizedBox(width: 20,));
            }
          }
        }
        notifyListeners();
    });
  }
}
```

Order.dart

```
class Orders extends StatefulWidget {
  Orders();
  @override
  _OrdersState createState() => _OrdersState();
}

class _OrdersState extends State<Orders> {
  bool updated = false;
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Container(
        child: Column(
          children: <Widget>[
            SizedBox(height: 30,),
            Txt('Orders', size:20),
            Container(
              child: ChangeNotifierProvider(
                builder: (_)=>OrderBack(),
                child: Consumer<OrderBack>(
                  builder: (context, orderBack,_){
                    final orderBack = Provider.of<OrderBack>(context);
                    if(!updated){
                      updated=true;
                      orderBack.update();
                    }
                    return Column(
                      children: orderBack.tray,
                    ); }
                ))
            ),],)
      ),
      floatingActionButton: Container(
        height: 44,
        width: double.infinity,
        decoration: BoxDecor(
          color: Colors.blue[300],
          radius: 40
        ),
        margin: EdgeInsets.only(left:30),
        child: Center(
          child: FlatButton(
            child: Txt('Order'),
            onPressed: (){},
          ),
        ),),
    ); }
}
```

Order_back.dart

```
class OrderBack with ChangeNotifier{
    List<Widget> tray=[SizedBox()];

    update(){
      if(DbServer.tray.length>0){
        tray=[];
      }
      for (var item in DbServer.tray) {
        tray.add(
          Container(
            decoration: BoxDecoration(
              border:Border.all(
                color: Colors.grey,
                width: 1,
              ),
            ),
            child: Row(
              children: <Widget>[
                Container(
                  height: 40,
                  width: 40,
                  child: Icon(Icons.fastfood)),
                Column(

                  mainAxisAlignment: MainAxisAlignment.spaceAround,
                  children: <Widget>[
                    SizedBox(),
                    SizedBox(),
                    Txt(item[0].toString()),
                    Txt(item[0].toString()),
                    SizedBox(),
                ],)
              ],
            ),

          )
        );
        tray.add(SizedBox(height: 10));
      }
    }
}
```

Web Application

Api.php

```php
<?php

use Illuminate\Http\Request;

Route::middleware('auth:api')->get('/user', function (Request $request) {
    return $request->user();
});


Route::post('login', 'UserController@login');
Route::post('logout', 'UserController@logout');
Route::post('profile', 'UserController@profile');

Route::middleware('auth:api')->post('/menu', 'MenuController@menu');
Route::middleware('auth:api')->post('/recommend', 'MenuController@recommend');
Route::middleware('auth:api')->post('/orders', 'OrderController@show');
Route::middleware('auth:api')->post('/order', 'OrderController@placeOrder');
Route::middleware('auth:api')-
>post('/payment', 'OrderController@orderDetails');
Route::middleware('auth:api')-
>post('/transactions', 'OrderController@orderDetails');
Route::middleware('auth:api')-
>post('/history', 'OrderController@orderDetails');
Route::middleware('auth:api')-
>post('/updateInfo', 'OrderController@orderDetails');
```

HomeOrders.vue

```
<template>
  <div id="app">
    <div class="heading">
      <h1>Orders</h1>
    </div>
    <ordered-item
      v-for="orderedItem in orderedsItems"
      v-bind="orderedItem"
      :key="orderedItem.order_id"
      @cook="cook"
      @ready="ready"
    ></crud-component>
    <div>
      <button @click="ready()">Ready</button>
    </div>
  </div>
</template>
<script>
  function OrderedItem({id, order_id, food_id, quantity, price}) {
    this.id = id;
    this.order_id = order_id;
    this.food_id = food_id;
    this.quantity = quantity;
    this.price = price;
  }

  import OrderedItem from './OrderedItem.vue';

  export default {
    data() {
      return {
        orderedsItems: []
      }
    },
    methods: {
      read() {
        window.axios.post('/orderedItems').then(({ data }) => {
          data.forEach(orderedItem => {
            this.orderedItems.push(new OrderedItem(orderedItem));
          });
        });
      },
      ready(id) {
        // To do
      },
      cook(id) {
        // To do
      }
    },
```

```
    components: {
        OrderedItem
    },
    created(){
      this.read();
    },
  }
</script>
```

UserContrpller.php

```php
<?php

namespace App\Http\Controllers;

use Illuminate\Support\Facades\Hash;
use Illuminate\Http\Request;
use Illuminate\Support\Str;
use App\User;

class UserController extends Controller
{

    public function index()
    {
        //
    }

    public function create()
    {
        //
    }
    public function store(Request $request)
    {
        //
    }

    public function show($id)
    {
        return User::where([
            'id'=>$id
        ])->first();
    }


    public function edit($id)
    {
        //
    }

    public function update(Request $request, $id)
    {
        //
    }

    public function destroy($id)
    {
        //
    }
    public function profile(Request $request){
```

```
        $profile = User::where([
            'api_token'=>$request->api_token
        ])->first();
        if($profile){
            $profile['success']='true';
            return $profile;
        }else{
            return response()->json([
                'success'=>'false',
                'error'=>'wrong Api token']);
        }
    }

    public function generateToken(){
        $api_token = Str::random(60);
        $result = User::where(["api_token"=>$api_token ])->first();
        if($result){
            $api_token = generateToken();
        }
        return $api_token;
    }

    public function login(Request $request){
        $api_token = $this->generateToken();
        $email=$request->input('email');
        $pass=$request->input('password');
        $result = User::where(["email"=>$email ])->get();
        if($result->isNotEmpty()){
            $id=$result[0]->id;
            $passwor = $result[0]->password;
            if(Hash::check($pass, $passwor)){
                $here= User::where('id',$id)-
>update(array('api_token'=>$api_token));
                if($here ==1){
                    return response()->json([
                        'result'=>'success',
                        'id'=>$id,
                        'token' => $api_token
                    ]);
                }else{
                return response()->json( ['result'=>'fu'] );
                }
            }else{
                return response()->json( ['result'=>'password'] );
            }
        }else{
            return response()->json( ['result'=>'email'] );
        }
    }
```

```
    public function logout(Request $request){
        $api_token = $request->input('api_token');
        if($api_token!=''){
            $result = User::where('api_token',$api_token)-
>update(array('api_token'=>''));
            if($result){
                return response()->json(['success'=>'true' ]);
            }
            else{
                return response()->json([
                    'success'=>'false',
                    'error'=>'wrong token'
                ]);
            }
        }else{
            return response()->json([
                'success'=>'false',
                'error'=>'blank'
            ]);
        }
    }
}
```

LogController.php

```php
<?php
namespace App\Http\Controllers;

class logController extends Controller
{
    //===function to get first date of given month
    private function getFirst($start){
        if($start[7]=='-'){
            $start = substr($start,0,8);
        }else{
            $start = substr($start,0,7);
        }
        $start=$start.'1';
        return $start;
    }
    //===function to get last date of given month
    private function getLast($start){
        $end = '';
        foreach(range(0,6) as $j){
            $end = $end.$start[$j];
        }
        if($end[-1]!='-'){
            $end=$end.'-';
        }
        $end = $end.date('t',strtotime($start)).' 23:59:59';
        return $end;
    }

    private function getTilToday($start){
        if (substr($start,0,7)== Date('Y-m')){
            return substr(date('Y-m-d'),0,10);
        }else{
            return $this->getLast($start);
        }
    }

    public function uploadImage(Request $request){
        $api_token = $request->api_token;
        if($api_token!=''){
            $emp = User::where(["api_token"=>$api_token ])->get();
            if($emp->isNotEmpty()){
                $id=$emp[0]->id;
                $imag = Image::where(['id'=>$id])->first();
                $name=$emp[0]->f_name;
                $old_filename=$imag->path;
                $byteimage = $request->image;
                $filename =$name.'_'.date('Y_m_d_H_i_s').'.png';
                $decoded = base64_decode($byteimage);
                $im = imageCreateFromString($decoded);
```

```
                if (!$im) {
                    die('Base64 value is not a valid image');
                }
                $img_file = 'profiles/'.$filename;
                imagepng($im, $img_file, 0);
                DB::table('images')->where('id', $id)-
>update(['image' => $filename]);
                unlink('profiles/'.$old_filename);
                return 'done';
            }
        }
    }

    public function checkNotification(Request $request){
        $api_token = $request->input('api_token');
        if($api_token!=''){
            $emp = User::where(["api_token"=>$api_token ])->get();
            $id = $emp[0]->id;
            if($emp->isNotEmpty()){
                $result = Notification::where('user_id',$id)-
>orderBy('created_at', 'DESC')->limit(5)->get();
                return response()->json(
                    [
                        'success'=>'true',
                        'data'=>$result]);
            }else{
                return response()->json([
                    'success'=>'false',
                    'error' => 'expired token'
                ]);
            }
            }else{
            return response()->json([
                'success'=>'false',
                'error' => 'empty token'
            ]);
        }
    }

    public function logout(Request $request){
        $api_token = $request->input('api_token');
        if($api_token!=''){
            $result = User::where('api_token',$api_token)-
>update(array('api_token'=>''));
            if($result){
                return response()->json(['success'=>'true' ]);
            }
            else{
                return response()->json([
                    'success'=>'false',
```

```
                    'error'=>'wrong token'
                ]);
            }
        }else{
            return response()->json([
                'success'=>'false',
                'error'=>'blank'
            ]);
        }
    }

    public function changePass(Request $request){
        $api_token = $request->input('api_token');
        $old = $request->input('old');
        $nuu = $request->input('nuu');
        if($api_token!=''){
            $result = User::where(["api_token"=>$api_token ])->get();
            if($result->isNotEmpty()){
                if(Hash::check($old, $result[0]->password)){
                    $result = User::where('api_token',$api_token)-
>update(array('password'=>password_hash($nuu, PASSWORD_DEFAULT)));
                    if($result){
                        return response()->json([
                            'success'=>'true',
                        ]);
                    }
                }else{
                    return response()->json([
                        'success'=>'false',
                        'error'=>'wrong old password'
                    ]);
                }

            }else{
                return response()->json([
                    'success'=>'false',
                    'error'=>'token Mismatch'
                ]);
            }
        }else{
            return response()->json([
                'success'=>'false',
                'error'=>'empty token'
            ]);
        }
    }

    public function checkExists(Request $request){
        $api_token = $request->input('api_token');
        if($api_token!=''){
```

```
            $result = User::where(["api_token"=>$api_token ])->get();
            return $result;
            if($result->isNotEmpty()){
                return 'true';
            }else{
                return 'false';
            }
        }
    }

    public function notifyUsers($title, $message, $image, $isTest , $players)
    {
        $app_id = "08725583-aaf5-4dfd-8d6d-6abc738f5539";
        $rest_api_key = "ODY4ZTE0ZmQtODQ5ZS00MDE5LThhMjYtYTI0MmM4ZjUwOTUw";
        $heading = array(
        "en" => $title
        );
        $content = array(
        "en" => $message
        );
        $fields = array(
        'app_id' => $app_id,
        'data' => array('just'=>'because'),
        'contents' => $content,
        'headings' => $heading,
        'large_icon' => $image,
        );

        if(count($players)>0){
            $fields['include_player_ids'] = array('1f539bf3-3ddc-4c69-afb4-
21890ba8dd24');
        }else{
            if ($isTest) {
                $fields['included_segments'] = array("Test Users");
            } else {
                $fields['included_segments'] = array("Active Users", "Inactive
 Users");
            }
        }
        $fields = json_encode($fields);
        print("\nJSON sent:\n");
        print($fields);
        $ch = curl_init();
        curl_setopt($ch, CURLOPT_URL, "https://onesignal.com/api/v1/notificati
ons");
        curl_setopt($ch, CURLOPT_HTTPHEADER, array(
        'Content-Type: application/json; charset=utf-8',
        'Authorization: Basic ' . $rest_api_key
        ));
        curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
```

```
        curl_setopt($ch, CURLOPT_HEADER, false);
        curl_setopt($ch, CURLOPT_POST, true);
        curl_setopt($ch, CURLOPT_POSTFIELDS, $fields);
        curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);

        $response = curl_exec($ch);
        curl_close($ch);

        $return["allresponses"] = $response;
        $return = json_encode($return);

        print("\n\nJSON received:\n");
        print($return);
        print("\n");
    }

    public function oneUser($id){
        $emps = User::where('user_id', $id)->get();
        return absences($emp['joined_date'],date('Y-m-d 23:59:59'), $id);
    }

    public function show(){
        $emps = User::select()->get();
        $abs = [];
        foreach($emps as $emp){
            if($this->absences(date('Y-m-d'),date('Y-m-
d 23:59:59'), $emp['id'])){
                array_push($abs,[$emp['name'],$emp->id]);
            }
        }
        return view('detail')->with('absentees',$abs);

    public function setPlayId(Request $request){
        User::where('api_token', $request->api_token)-
>update(['player_id'=>$request->playerId]);
    }
    public function passReset(Request $request){
        $headers = "From: ndpstha@gmail.com" ;
        mail('ndpstha@gmail.com', 'feather webs password reset', 'yolololololo',
$headers);
    }
}
```