Irakli Lomidze

G00275525

B.Sc. (Hons) in Software Development

Hiking Ireland Application Developed With Ionic 5

Final Year Project
Supervised by: Dr. Dominic Carr

Department of Computer Science and Applied Physics Galway-Mayo Institute of Technology (GMIT)

Contents

Abstract

For my final year project, I wanted to make an app that would assist people who are Hiking and outdoor enthusiastic in planning their trip. Hiking is healthy way to live in life and very therapeutant and relaxing experience. As a project developer i want to design a free web application where the user can enter data and store it, users planning their next trip and adding in the list. There is also Google maps section where user can find locations and save trips. I have created this application with a client and server pulling data from databases. I used three main technologies in developing this application.

- Ionic 5 – For presentation
- NodeJS – For logic
- Firebase – For Database

Authors:

- **Irakli Lomidze**

Chapter 1 Introduction

When choosing this project, I wanted to make sure I was going to develop something that could be used by anyone in their everyday life. I wanted a project that also showed existing skill that I have gain trough in 4 years as a student and improve on that by adding new set of skills that would help my overall development as software developer. With this criteria's in mind I began research to discover new was of developing software.

After doing some brainstorming on what application I wanted to develop, I eventually decided on app called Hiking Ireland, as the name suggests this application main components will be based on outdoor activities. Looking through already existing Hiking application categories main one that catch my eye was Gaia GPS which has similar functionalities as what I wanted my application to have, It gave me a lot of ideas about what my app might need to include, as well as what else I could do in my own version.

As the main focus and feature on this application is google maps API I knew I had research how to implement this in my application, my ide was for user to use Google maps to record users location and user to navigate and drop marker between two locations. For this feature I choice to have two separate pages for user to navigate to.

Focus on my version of application was to have more easier and simpler user expectance so that anyone could download app and use straight away without any issues.

This meant I was going to have simple layout for the app and easy on the eye, when application started user will be greeted by five main tabs/pages that are easy to fallow.

Chapter 2

Context

The general context of this application revolves around people who love outdoor Hiking activities. Users of this application will be able to keep track on their hiking trips, store information about trips, record location, search places and display information, capture multimedia and change user settings this are core application features.

- Home
- Record
- Places
- Capture
- Settings

2.1 Objectives

The main objective of this application is to help users who to discover unique outdoor spots in Ireland. Also wanted to make it easier for users, who are finding it difficult to keep track of their trips and places.

The following is the list of core pages in this application and their main functions.

1. **Login/Create account page -** A login and a registration page are available in the application. The login page allows users to safely log into the application, while the register page allows users to safely register with application. The user has access to additional functionality that are not available to unregistered users until signed in.

2. **Home page** – Home page is base for navigation for the application, when successfully logged in home page will display users list of trips.

3. **Record** – Function of this page is to record your desired location and save it in database.

4. **Places** – Function of this page is that user can search locations, dropped the pin between user current location to where users wishes to go to.

5. **Capture** – Function of this page is that user can compute multimedia and upload.

6. **Settings** – Function of this page is that user can edit user details or log out from the application

The following is the list of Personalized Features that app comes with.

1. **Upload** – Function of this page is that user can upload pictures from their devise.

2. **My Favorites** – Function of this page is for users to see their favorite places that they have added to 'Favorites' list.

3. **Trip History** – Function of this page is to display trips they have saved, saved trips can be viewed by user.

4. **Calories Calculator** – This feature was added as an extra feature for the app, user can feel in the form and their calories will be calculated.

The following is the list of General Features that app comes with.

1. **Login** – Page for user to login in the application.

2. **Register** – Page for user to register in the application**.**

3. **News/Events** – Page for user to see up to date news.

4. **Covid-19** – Page for user to see up to date Covid case information.

5. **Home** – Page for user to return on main home page.

2.2 Project Links

Repository Link: https://github.com/ika25/hikingapp2.0

2.3 Chapters Review

This paper is divided into chapters that cover anything from project preparation to solution design and implementation. The following subsections will give you a quick summary of each chapter in the project.

2.3.1 Methodology Used

In this chapter, I'll go through some of the methodologies I used in the development of the application. Agile, version control and testing will all be covered in this section.

2.3.2 Technology Review

In the Technology Review I will review technologies that I used and helped me in development of this application, this includes front end and back end technologies and development tools.

2.3.3 System Design

I'll go over the system architecture and functionality of this application in depth in System Design. I'll go into why these innovations were chosen and how they were incorporated into the device design.

2.3.4 System Progression and Evolution

The robustness, testing, Results V Objectives, and limits of the system will be evaluated in this chapter.

2.3.5 Project Conclusion

In the end, I'll summarize the project in terms of its aims and objectives. The application will be reviewed, and potential future progress will be discussed.

Chapter 3

Methodology Used

In this chapter of the report the methods in which features were designed developed and tested will be described, the theory behind the project and each of its individual features will be explained as well as the research and steps taken during the development phase of the feature in question, as well as the approach taken to solve any problems that were encountered. First of all, the general research and methods will be explained then the actual way in which features were implemented will be explained.

3.1 Approaching the problem

The problem was approached by breaking down the overall project down into smaller individual components that were later combined to from functional system. The breakdown of the project is as fallows.

- Hardware
  - Mobile device
- Software
  - Deployment environment set up
  - Server Application and Database Construction
  - Desktop application for viewing of data stored on database.

As a person with Agile software mythology, I learned that It was best to tackle projects was feature-by feature- instead of in big phases. With this approach a goal oriented module approach was taken, small goals that were flexible were established and the above features were never developed from start to finish individually as some aspects of one feature progressing gradually which in result allowed the developer to jump from one part of the system to another.

What this resulted in was small blocks of functionality that could easily be adapted to the main project, removed or modified. The design process of each of the blocks started with a goal, research on how this goal should be achieved and then implementing the knowledge into a small project which would later be merged with the main project. This type of approach continued until the completion of the systems, this practice was adopted due to the fact that it made bug fixing easier and it also added an element of safety due to the fact that it wasn't implementing new functionality to the main project but first being thoroughly tested before the additions was applied and this method made sure that new code wouldn’t jeopardize the functionality of the main code and even if it were to cause problems it was easy to extract and fix.

3.2 Version Control

I used GitHub during the project's life cycle. GitHub is a version control hosting service. I made a GitHub repository because I considered it to be a very useful tool in the research and creation of this app. Despite the fact that I mostly used GitHub to manage source code.

GitHub also allowed me to share my work on a weekly basis with my project advisor. You can see on the repository when new items are committed and exactly what has been modified. The ability to roll back the project if necessary was another feature I used with GitHub. Github monitored the whole project development and I can return to any of the previous commitments at any time, which was very helpful.

3.3 Testing

I needed to decide how I would test the application because I did not choose to use a framework for example J-unit to test it. I did, however, decide to use white and black box testing.

3.3.1 White Box Software Testing Technique

The tester has access to the software's internal workings during white box testing. White Box Testers have a complete and comprehensive understanding of the internal composition of software and usually are software developers. I thought I was playing this role, and thus I performed the software tests. I've written down a number of characteristics I want to test and the expected result. If I tested the log-in system, for example, the expected result would be that the user logged in and his user page is displayed. Then I'd test it and follow the way the program goes through the code.

3.3.2 Black Box Software Testing Technique

Black box testing is a technique for testing a portion of a program without having a thorough understanding of the software's internal workings. The internal program design, structure, or implementation are unknown to a black box testing system. I asked my friends to test some of the software's features. The web app was hosted on the Google cloud platform, so I could send them a link to ask for certain features. Again, I'd have made a list of anticipated results and asked them to recommend any modifications if they found any issues. This was helpful to me because I often skipped something small.

3.4 Development Timeline & Tasks

This section I will explain each of the fallowing tasks I completed and research done for this project gave me chance to learn how to complete certain task on this project, creating architecture for this project that would help me to integrate all of the features together. Bellow you will find research that was done on each task and timeline for project development for each part of the project.

3.4.1 Creating Pages

When starting development first I have decided on the number of pages my application would have, I created application pages that I was going to use and on top of that I added few extra pages for extra features that I was going to have include in application if I had enough time to do it, for example calorie counter and step counter. By creating pages that give my application base to work with.

3.4.2 Menu Navigations

Next step was to implement menu navigation this would connect all the pages in my application, I also added side menu In application that would be separate from main menu application features. After completing this task now menu navigations were responsive.

3.4.3 Registration and login

In this task I implemented the registration and login functionality, for this I did research on Firebase real time database and how to set up user authentication in database. This stage I have successfully step up authentication with firebase.

3.4.4 Back End

Task here was to implement NodeJS server and set up API routes along with firebase database to be able store data in database such as creating deleting and updating data.

3.4.5 Home page

On home page task I started to implement function for user to add hiking spots, after that I implemented CRUD function so that user could create, rea, update and delete their Hiking spot.

3.4.5 Google Maps API

In this task I started research to have to connect Google maps API to my application after doing research I generated google Maps API key which helped to connect my application to google maps from where I could have Geolocation and google places set up and working.

3.4.6 Settings

In settings I wanted to implement for user to update their profile details if they wanted and log out from application. This was successfully implemented.

3.4.7 Adding Extra Features

In this task I started to implement first extra feature into my application, with calorie calculator I was able to find formula online to have to calculate calories for person and implemented into the application. This was done by user having to fill in form and at the end it would generate their daily calorie intake. Next extra feature was user to capture media with their devise this was relatively easy to implement as there is Cordova plugin for this feature. I also added news api and covid-19 api for live updated information.

3.4.8 Favorites

Task here was to pull information from home page where user has created their hike spot and added to their favourites so that user could see them in their favourites page, I was able to complete this task.

3.4.9 Record and Places

In this task I went back to Goole maps implementation and improved feature I intended to have in this application, on record page I was able to step up google maps to record users location and store in firebase database, and with places page for user to search destination A and drop marker to Destination B by doing so user would be able to see distance between two destinations. This was implemented and by end of application development was looking more suitable to good standard.

Chapter 4

Technology Review

In this part, I'll go over the application's overall design and architecture. This will be accomplished by using code snippets to aid in the comprehension of the specification. The presentation, details, and logic sections of the technology review chapter will be divided into three sub sections. Other technologies I used that didn't fit into the three key subsections will be listed in a separate subsection.

4.1 Presentation

4.1.1 Ionic

Ionic is a free front-end SDK that allows you to build hybrid mobile applications that use web technologies. Like HTML, SCSS, and Typescript. It uses Cordova and Ionic Native to include mobile-optimized web technology components as well as native APIs. Angular also contributes significantly to an Ionic application's efficiency. It comes with its own command line interface tool that is extremely useful for scaffolding and developing an application, particularly in avoiding boilerplate code and thus saving time. [6] I recommend going to the following site for a more in-depth look at the Ionic structure.[7]

4.1.2 Ionic User Interface

Ionic comes with a range of ready-to-use components that can be easily incorporated into our application's code to greatly enhance the layout. The following website provides links to these elements. [8]

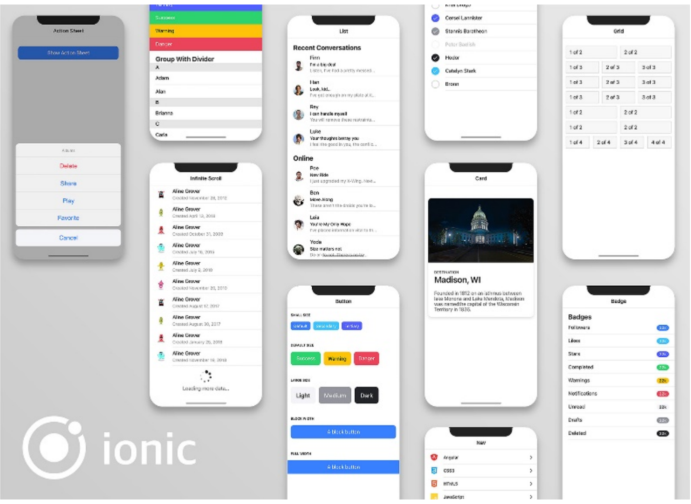


Figure 4.1.2: Ionic

4.1.2 Cordova

Cordova Apache is an open source framework for developing mobile devices. Using the native development language for each mobile platform, it allows cross-platform development using popular web technologies like HTML5, CSS3, and JavaScript. Applications run within a platform-specific wrapper that uses standard APIs to access system sensors, data, and network status. Cordova is a platform for developing mobile applications,for a variety of platforms by embedding the browser inside the app.

In fact, your app is a small browser that only displays one website: your app. To speed up the download, all resources can be stored in the application's delivery package, and you can also download from the server if appropriate.

On the mobile device by default, Cordova only offers the most basic browser features; however, you can expand the number of functions available in the browser by using plugins. Each plugin has a single user interface that can be accessed from any computer using a browser. Cordova is used in my system to implement various plugins on various Android/IOS devices. [1]

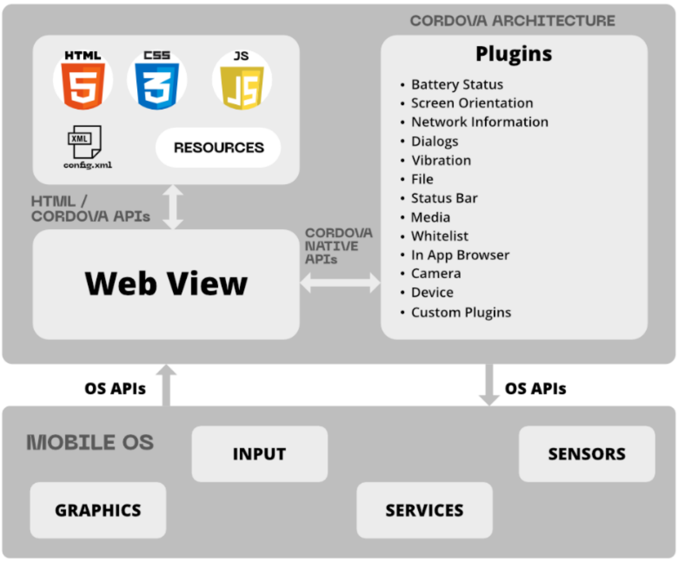
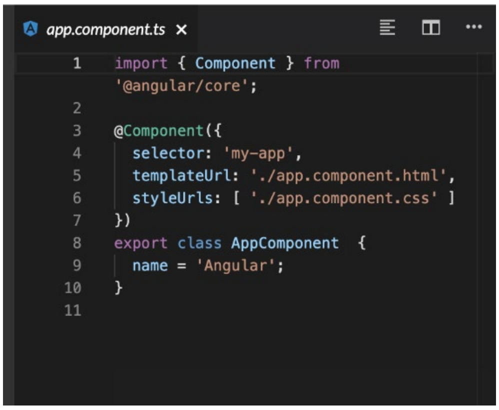


Figure 4.1.2: Cordova

4.1.3 Angular

Angular is a Google platform for developing client software. Its first goal is to build single-page applications using SPA solutions. AngularJS is the successor to another AngularJS project in this regard. Angular, on the other hand, is not a modified version of AngularJS, but a whole new system. Two-way binding, for example, in Angular 10 allows you to dynamically modify data in one room, such as models, routing, and so on. The fact that Angular uses Typescript as its programming language is one of its most notable features. However, Typescript isn't the only choice. You can write Corner applications in languages like Dart or JavaScript if you want. [2]

However, Angular continues to use Typescript as its primary programming language. The components are one of the most important aspects of the application. The part is in charge of how the presentation is shown on the screen.



```
import { Component } from
'@angular/core';

@Component({
  selector: 'my-app',
  templateUrl: './app.component.html',
  styleUrls: [ './app.component.css' ]
})
export class AppComponent  {
  name = 'Angular';
}
```

Figure 4.1.3: Angular Component Example

The term "export key" refers to a class that can be used in other modules. In the same class, only one variable is specified, and it stores a string as a value. To designate a class as a component, use the @Component decorator. If the @Component decorator was not added to the AppComponent class, it would not be considered a component. @Component is a decorator that takes an object and configures it to describe the framework, how to communicate with it, and how to interpret it as an argument.

4.2 Logic

4.2.1 NodeJS

Figure 4.2.1: NodeJS

NodeJS is a server-side platform based on the JavaScript Engine in Google Chrome. Ryan Dahl created NodeJS in 2009, and the most recent version is 16.0.0. NodeJS is a framework for easily creating fast and scalable network applications based “on Chrome's JavaScript run-time. NodeJS uses an event-driven, non-blocking I/O model that makes it lightweight and” [13] effective, making it ideal for data-intensive real-time applications that run across multiple devices, as well as for our application. [3]

4.2.2 npm

Figure 4.2.2: npm

The package manager included with NodeJS is called NodeJS Package Manager. Rather than downloading each library separately, we can now use a single command to install all at once.

npm install - When you run the command the package.json file is searched for in the current folder by npm If a library from the list is found, it will be installed. [5]

4.3 Data Section

The data section displays all of the data that has been saved and allows you to store, view, update, and delete specific items. I only have one database for my application. User details, Google maps, history, images, and other data are all stored in this database.

4.3.1 Firebase

Figure 4.3.1: Firebase

“Firebase is a mobile and web app development platform developed by Firebase Inc., which was acquired by Google in 2014” [5]. "Building, developing, and growing your software" is what Firebase is all about. It comes with features “like analytics, databases, messaging, and crash reporting, so you can turn easily and focus on your customers. One of the fastest-growing application development” [5] services is Firebase.

Some of the factors are as follows:

1. You don't have to start from scratch when creating a back-end. Firebase is a ready-to-use back-end that includes a database. You just need to include the Firebase SDK in your app and you're ready to go.
2. It is in real time. If you're a developer, you already know how important it is to have a real-time backend/database in today's app industry. When you consider real-time operations, things like chat, news feeds, ratings, and bookings become quite easy.
3. Authentication operations that are simple. Login/register operations are one of the first things that a user-facing program requires. Firebase does this with ease and with very little coding effort.
4. You get lots of extra features built in, such as push alerts, analytics, and so on.
5. It's free, but there's a limit on how much you can use it.

[5]

4.4 Additional

4.4.1 Google Maps API

Figure 4.4.4: Google Maps API

With static and dynamic maps, you can create personalized, “agile experiences that bring the real world to your users. [6]” With high-quality directions and real-time traffic alerts, provide your users with the best way to get from point A to point B. To establish more detailed itineraries, determine the path the user takes.

Places provides users with rich location data for over 100 million locations, allowing them to explore the globe. Allow them to use addresses to locate unique locations.

4.4.2 Typescript

Figure 4.4.2: TypeScript

Typescript can be compiled in JavaScript and is backwards compatible. In reality, after compilation, a Any latest browser can run a Typescript application, and it can also be used in combination with the NodeJS server platform. The Apache license applies to the code of the compiler that converts Typescript to "JavaScript is a scripting language that allows you to createThe ability to expressly implement static types, as well as support for classes and connecting modules, as in conventional object-oriented languages." [7] differentiate Typescript from JavaScript, all of which are intended to speed up development, improve readability, refactoring, and reuse of code, and check for errors during development and compilation.

4.4.3 HTML

Figure 4.4.3: HTML

HTML stands for hypertext mark-up language, and it's a programming language for making web pages. HTML is used to generate the web page, and it includes all of the required elements. A link, a table, a numbered or unnumbered list, photos, text divided into paragraphs and sections, and section headings can all be found on a simple HTML page. [9] define section headings to a simple HTML page. On the HTML tab, you can also use text fields, buttons, select options from a list, check-boxes. HTML5 allows you to add video and audio files to a tab, draw on a canvas, and make quick animations with new tags.

4.4.4 SCSS

Figure 4.4.4: CSS

Sassy Cascading Style Sheets (SCSS) is a CSS superset. So, first and foremost, I'll define CSS, since it covers a lot of what Sccss does.' CSS actually aid in separating the structure and content of the page “from its appearance. Each code factor defines both the content and the appearance of the page if the page is entirely written in HTML.” [8] page's content, but also how it is displayed. For example, not only is there a text “Hello” in this or that location, but it is also highlighted in bold and red. All works a little differently when CSS code is used. Only the order of the page's content elements, as well as their classes, are represented using HTML. In the CSS file, the corresponding classes are written. A set of properties is allocated to each of them. When we assign a class to an HTML element, it inherits all of the properties of that class. The amount of code that has to be repeated is significantly reduced as a result of this. When a site has a lot of pages, therefore, CSS is needed.

4.4.5 JavaScript

Figure 4.4.5: JavaScript

‘JavaScript is a scripting or programming language that allows you to implement complicated parts on web pages, if a web page does anything than just sit there and show static details for you to look at, such as showing timely content changes, interactive charts, animated 2D/3D graphics, scrolling video, and so on.’ [10]

JavaScript is almost certainly involved. It's the third layer of a three-layer cake of standard web technologies, the first two of which (HTML and CSS) have been discussed extensively elsewhere in the Learning Field.

4.5 Software and tools used

4.5.1 Visual Studio Code

Figure 4.5.1: Visual Studio Code

Microsoft has made a number of development tools available. However, Visual Studio Code, which runs on Linux, OS X, and Windows, is probably the best thing they've ever published. Without giving up all of the benefits of a full-featured VisualStudio IDE. Microsoft wanted to restructure the programmers' main toolkit, starting with the most significant, the code editor.

Although Visual Studio Code is primarily an editor, it also includes integrated development environments (IDEs) that rely on extensions. Visual Studio Code also supports a variety of languages, including JavaScript, Typescript, C sharp, and others. Visual Studio can also be used to develop ASP.NET 5 or NodeJS web projects, as well as communicate with package managers such as npm and debug them.

Code fragments, refactoring, navigation, multi-windowing, and git support are just a few of the features accessible. Visual Studio Code is, in some respects, much more convenient than the complete version of Visual Studio and has less hardware specifications.

4.5.2 GitHub

Figure 4.5.2: GitHub

GitHub is a project hosting and collaboration platform that allows users to host, upload, and collaborate on projects. It is free to use in Open Source projects. Repositories are GitHub projects that are hosted on the platform. It is here that users can access their projects, download a version of the project to their system, and make contribution to the exciting projects. Collaborating on projects, can be difficult because other people's work can intervene with yours and cause problems. To avoid this, GitHub's version control handles this for you by allowing you to undo changes at any time during the project. To split job trees, use branches, and to combine branches together, use merge. For programmers and developers, GitHub is a fantastic social networking site. in addition to being a hosting project. Users may follow each other, subscribe to project notifications, like them, give them feedback, and so on. These features enable users to keep up to date on projects that they are involved in, as well as communicate with colleagues and employees. These days its good to have GitHub profile if want to apply for work it acts as portfolio of your work and helps employers to see what your skills are.

GitHub isn't just for software creation and programming. It's also used in a variety of other programs. Open source guides, documentation programs, educational tools, and other collaborative projects between users.

[https://docs.github.com/en]

4.5.3 Microsoft Word

Figure 4.5.3: Microsoft Word

I choice between LaTeX and Microsoft word to use for writing my dissertation document. My choice was Microsoft Word, I was introduced LaTeX only in first semester in one module and due to busy college schedule, I didn’t have enough time to get comfortable with it. Where Microsoft word is something I have used in years and even though am not expert in word I still have better knowledge on have to use it more sufficient and less time-consuming way. It was much easier to create, edit and read document in word. It has lot of useful features that are very easy to pick up and use it. Newest versions of Microsoft word even has PDF file converter built in its easy as one button click to convert word document into PDF file which was required for final Dissertation document delivery.

[https://www.techwalla.com/articles/list-of-ms-word-features]

Chapter 5

System Design

I will go over the overall design and architecture of the Application in this section. To demonstrate this I will attach code snippets and visual graphs to help you understand the application's architecture. The System Design chapter will be divided into four parts. Firebase Database, Firebase Authentication represent by Data, The NodeJS server represents logic, and Presentation is represented by the Ionic 5 App and Application deployment.

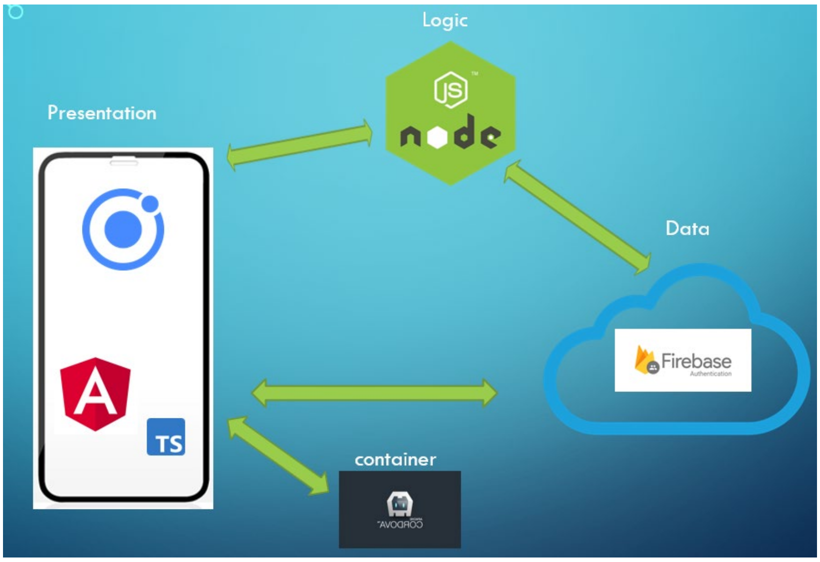


Figure 5: System Design

5.1 Data

The data structure represents all of the data that has been processed and allows you to store, view, edit, and delete specific data. For this application I am using Firebase which has all the database functions built in, instead of using few different databases, this way it’s much easier to keep track and manage database for application.

5.1.1 Firebase Authentication

Since Firebase is well-known in the Ionic community as a popular database for login and registration, I chose it to manage user accounts. Since it provided pre-built features including password reset, email, and password authentication. This free Authentication API services appealed to me.

5.1.2 Firebase real time Database

I chose to use Firebase real time Database to handle database requirements of the application, it provides cloud database to store and sync data from client and server-side development. In my database structure you can see how user data is stored. There is many resources to follow for Firebase which made things easier.

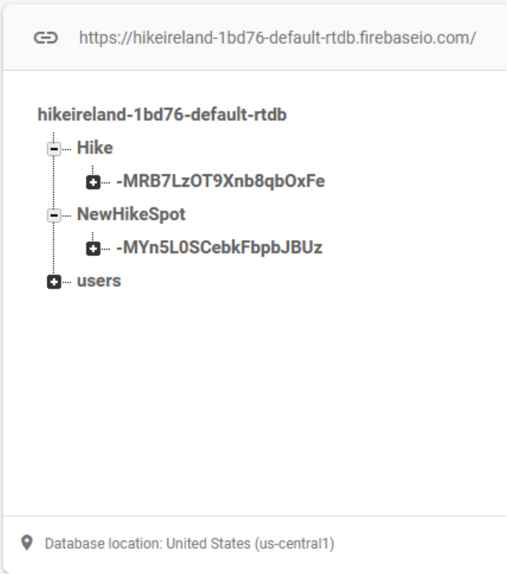

https://hikeireland-1bd76-default-rtdb.firebaseio.com/
hikeireland-1bd76-default-rtdb
Hike
-MRB7LzOT9Xnb8qbOxFe
NewHikeSpot
-MYn5L0SCebkFbpbJBUz
users
Database location: United States (us-central1)

Figure 5.1.2: Database Structure

5.1.3 Firestore

As I already mentioned firebase has many nice features and they are always improving, one of the newest feature is Firestore Database their newest database which is used for mobile app development. I took advantage to learn bit more of this new feature by adding in my application. In Firestore I will be storing user data such as images and Google Maps locations history saved by user.

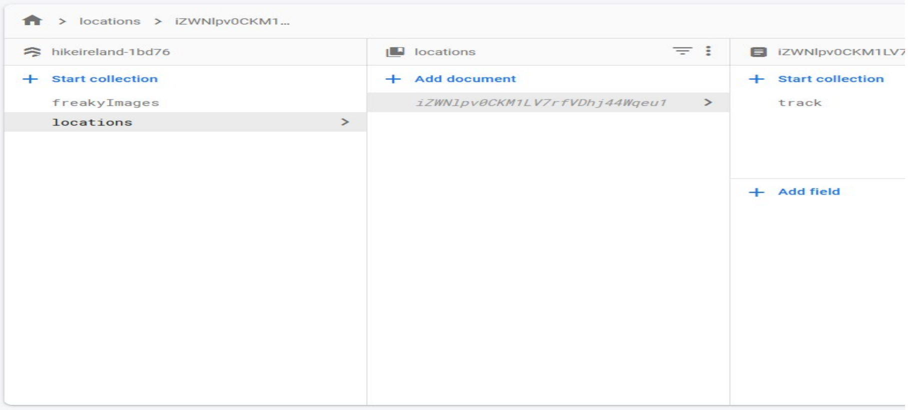


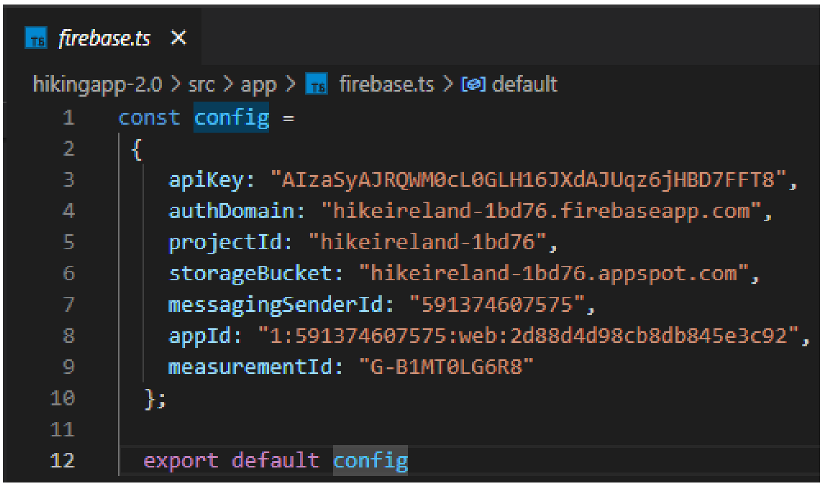
Figure 5.1.3: Firestore

5.2 Logic

We program the application's ability to manipulate data in this section. These operations include data generation, storage, upgrading, and deletion. I use NodeJS to manage our business logic. NodeJS was an obvious choice for project architecture because it not only works well with Firebase but also integrates well with Ionic for presentation. NodeJS was more than capable for application due to its lightweight nature and scalability due to its non-blocking I/O calls that enabled tens of thousands of concurrent connections. Because of their similarities, using Javascript for application logic and type-script for providers in the presentation made transitioning between the two a breeze.

5.2.1 Connection to Data

To set up basic data connection between Firebase and application. I add project on firebase this can be done on their website. After adding project Firebase configuration keys was issues these keys evoke ionic and firebases consensus.

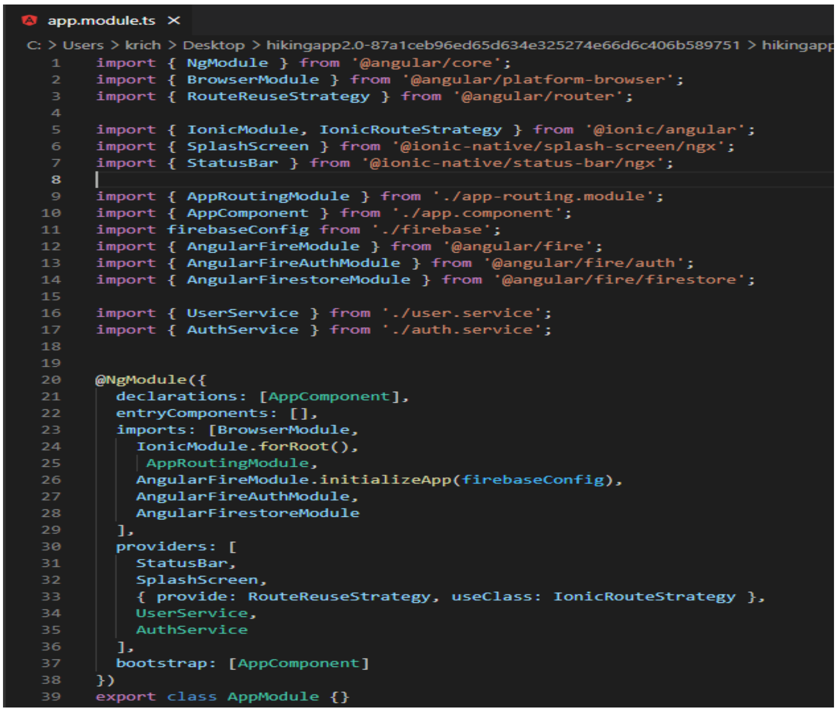


```
const config =
 {
   apiKey: "AIzaSyAJRQWM0cL0GLH16JXdAJUqz6jHBD7FFT8",
   authDomain: "hikeireland-1bd76.firebaseapp.com",
   projectId: "hikeireland-1bd76",
   storageBucket: "hikeireland-1bd76.appspot.com",
   messagingSenderId: "591374607575",
   appId: "1:591374607575:web:2d88d4d98cb8db845e3c92",
   measurementId: "G-B1MT0LG6R8"
  };

  export default config
```

Figure 5.2.1: Connection to Data

5.2.2 Routing and Accessing data

Next step is to add firebase modules in Application module.

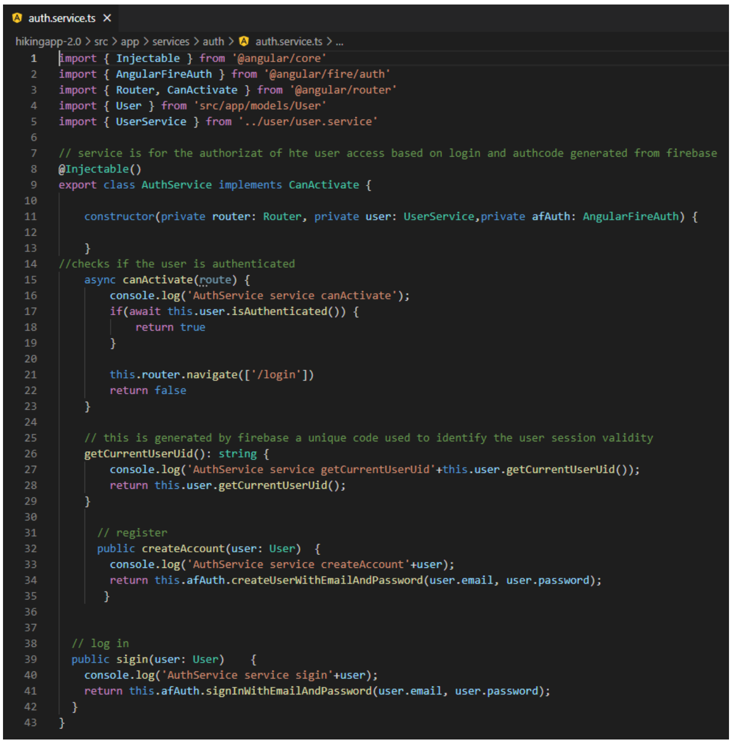

app.module.ts
C: > Users > krich > Desktop > hikingapp2.0-87a1ceb96ed65d634e325274e66d6c406b589751 > hikingapp

```typescript
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { RouteReuseStrategy } from '@angular/router';

import { IonicModule, IonicRouteStrategy } from '@ionic/angular';
import { SplashScreen } from '@ionic-native/splash-screen/ngx';
import { StatusBar } from '@ionic-native/status-bar/ngx';

import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
import firebaseConfig from './firebase';
import { AngularFireModule } from '@angular/fire';
import { AngularFireAuthModule } from '@angular/fire/auth';
import { AngularFirestoreModule } from '@angular/fire/firestore';

import { UserService } from './user.service';
import { AuthService } from './auth.service';


@NgModule({
  declarations: [AppComponent],
  entryComponents: [],
  imports: [BrowserModule,
    IonicModule.forRoot(),
     AppRoutingModule,
    AngularFireModule.initializeApp(firebaseConfig),
    AngularFireAuthModule,
    AngularFirestoreModule
  ],
  providers: [
    StatusBar,
    SplashScreen,
    { provide: RouteReuseStrategy, useClass: IonicRouteStrategy },
    UserService,
    AuthService
  ],
  bootstrap: [AppComponent]
})
export class AppModule {}
```

Figure 5.2.2: Routing

And now we have successfully added firebase database to my application and is ready for further development.

User Authentication

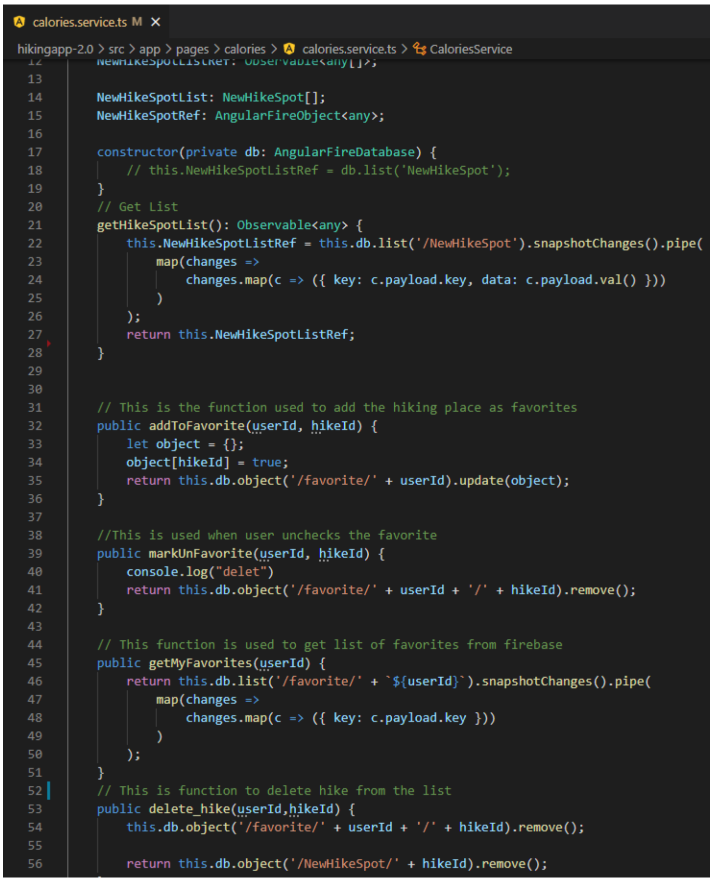

auth.service.ts
hikingapp-2.0 > src > app > services > auth > auth.service.ts > ...

```typescript
import { Injectable } from '@angular/core'
import { AngularFireAuth } from '@angular/fire/auth'
import { Router, CanActivate } from '@angular/router'
import { User } from 'src/app/models/User'
import { UserService } from '../user/user.service'

// service is for the authorizat of hte user access based on login and authcode generated from firebase
@Injectable()
export class AuthService implements CanActivate {

    constructor(private router: Router, private user: UserService,private afAuth: AngularFireAuth) {

    }
//checks if the user is authenticated
    async canActivate(route) {
        console.log('AuthService service canActivate');
        if(await this.user.isAuthenticated()) {
            return true
        }

        this.router.navigate(['/login'])
        return false
    }

    // this is generated by firebase a unique code used to identify the user session validity
    getCurrentUserUid(): string {
        console.log('AuthService service getCurrentUserUid'+this.user.getCurrentUserUid());
        return this.user.getCurrentUserUid();
    }

      // register
      public createAccount(user: User)  {
        console.log('AuthService service createAccount'+user);
        return this.afAuth.createUserWithEmailAndPassword(user.email, user.password);
       }


  // log in
  public sigin(user: User)    {
    console.log('AuthService service sigin'+user);
    return this.afAuth.signInWithEmailAndPassword(user.email, user.password);
  }
}
```

Figure 5.2.3: Auth

In here you can see how user authentication is implemented and working with Firebase.

Create, Update and Delete


calories.service.ts M ×
hikingapp-2.0 > src > app > pages > calories > calories.service.ts > CaloriesService

```
  NewHikeSpotListRef: Observable<any[]>;

  NewHikeSpotList: NewHikeSpot[];
  NewHikeSpotRef: AngularFireObject<any>;

  constructor(private db: AngularFireDatabase) {
      // this.NewHikeSpotListRef = db.list('NewHikeSpot');
  }
  // Get List
  getHikeSpotList(): Observable<any> {
      this.NewHikeSpotListRef = this.db.list('/NewHikeSpot').snapshotChanges().pipe(
          map(changes =>
              changes.map(c => ({ key: c.payload.key, data: c.payload.val() }))
          )
      );
      return this.NewHikeSpotListRef;
  }


  // This is the function used to add the hiking place as favorites
  public addToFavorite(userId, hikeId) {
      let object = {};
      object[hikeId] = true;
      return this.db.object('/favorite/' + userId).update(object);
  }

  //This is used when user unchecks the favorite
  public markUnFavorite(userId, hikeId) {
      console.log("delet")
      return this.db.object('/favorite/' + userId + '/' + hikeId).remove();
  }

  // This function is used to get list of favorites from firebase
  public getMyFavorites(userId) {
      return this.db.list('/favorite/' + `${userId}`).snapshotChanges().pipe(
          map(changes =>
              changes.map(c => ({ key: c.payload.key }))
          )
      );
  }
  // This is function to delete hike from the list
  public delete_hike(userId,hikeId) {
      this.db.object('/favorite/' + userId + '/' + hikeId).remove();

      return this.db.object('/NewHikeSpot/' + hikeId).remove();
```

Figure 5.2.4: Create, Update, Delete

In here you can see example of how Create, delete and update function is working with Firebase.

5.2.3 Connecting to Presentation

I'll take a brief look at how we're linking our logic and presentation parts. I've configured my application server to run on port 8100 in the logic. Our servers can send and receive traffic to our presentation part via this port. The presentation will be able to submit http requests to the server using the IP address, port number, and routing address. The routing address will decide what action the server takes with the info.

5.3 Presentation

I'll be looking at five areas in the Presentation Part section. I'll go into how my Ionic 5 app is designed, how our providers communicate with our NodeJS server and Data databases, a quick tour of each of our Applications pages to demonstrate how they look and work, a quick discussion of Pipes in the app, and finally how I used SCSS to style the application.

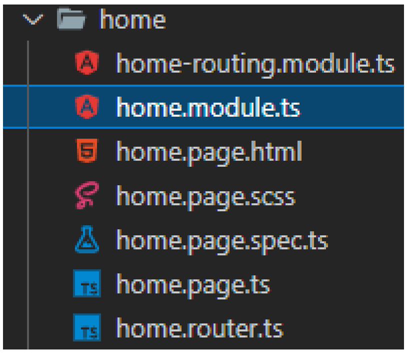
5.3.1 Ionic 5 Application Structure

I will give brief explanation to main folders that I have in Application folder.

- Node-Modules – in here we have all the packages that are required to run and develop this application
- src – This folder contains most of coding for this application. For example Typescript, Angular, JavaScript, CSS.
    - App – In here we have main base set up for the application, in here you will find app components for pages and creating application menus, tabs and structure.
    - Model – In here we create model for the user.
    - Pages – In here we have all the application presentation code goes. Here we go through page designs and how they connect to the application
    - Services – In here we have functions for user, news and covid-19.
    - Assets – In this folder we any images/icons.
- www – In here you find Cordova plugins that was used for this application, index.html and root component where the application will load.
- Theme – In here we setup overall design and look for application using GCSS.

6 Pages

In Ionic 5 pages are broken down in 6 parts



home
home-routing.module.ts
home.module.ts
home.page.html
home.page.scss
home.page.spec.ts
home.page.ts
home.router.ts

Figure 5.3.2: Pages Broken Down

- Home-routing,module.ts - When a user navigates to a URL, routing,module.ts is used to decide which part should be displayed or which module should be loaded.

- home.module.ts – This serves as way to describe the HomeModule with properties.

- home.page.html – In here html file creates page layout. Creating buttons, forms and lists.

- home.page.scss – SCSS page design.

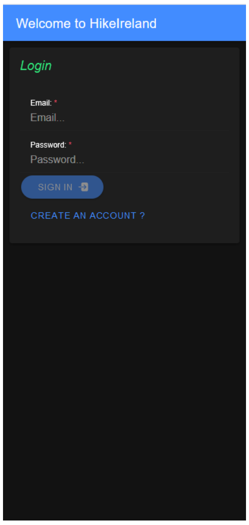
- home.page.spec.ts - This file is for unit test, this is generated when creating new application, I later found out that you can skip this file by ng new ng-app-name --skip-tests. I didn't use unit testing In this project,it would've been great to know how not to include them from the start to avoid extra files.

- home.page.ts – In here we code functions in bring in imports that needed for functionality.

- home.router.ts - contains all routings defined during navigation of the application.

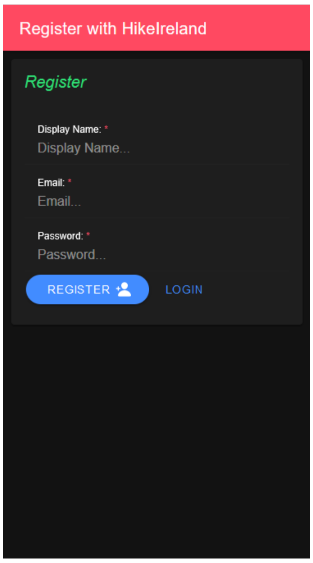
6.1 Login Page

This will be the first page the user sees when they open the app. They will use an email address and password to log in to their account. Validation is conducted before entering the email and password to ensure that the email is in the correct format, and that the password contains the correct characters and length. In here, Firebase Authentication is used to query the Firebase API to see whether the user exists in our Firebase Database.



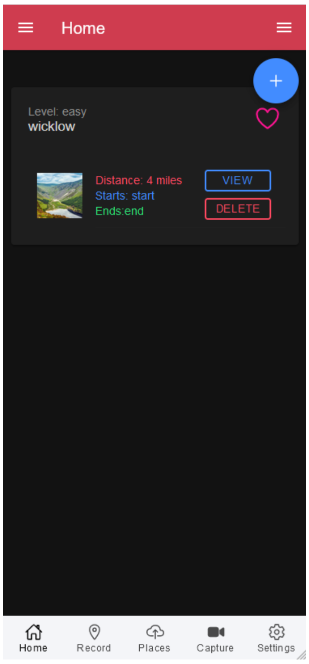
6.2 Register Page

When a user visits the register page, they will be asked to provide an email address and a password to create an account. The email and password fields are validated in the same way as they are on the login page. When a user clicks 'Register,' the Firebase API saves the user's information and returns a session key, allowing the user to access our app.



6.3 Menu Bar

The menu bar is accessible ones user is logged in and can navigate freely through the application by using application options.

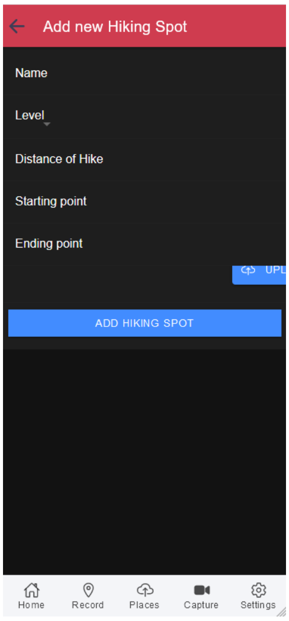


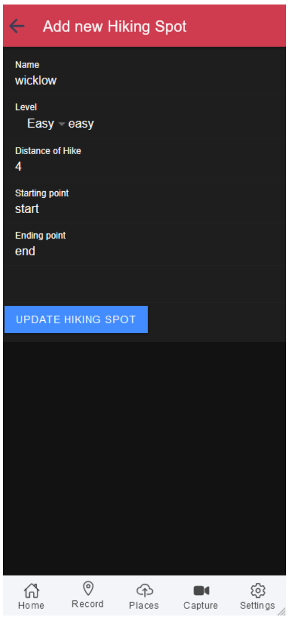
Level: easy
wicklow

Distance: 4 miles
Starts: start
Ends:end

6.5 Home Page

In home page user can add new Hike spot by clicking +, view or update their list, delete and favourite their Hike spot. These functions are not in separate pages for this part I chose to use prompt function where user is directed to form to where user can make alterations.





Name
wicklow

Level
Easy - easy

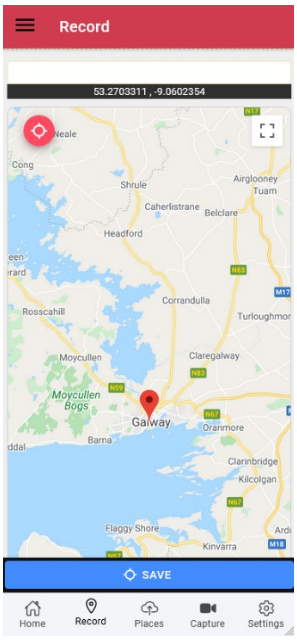
Distance of Hike
4

Starting point
start

Ending point
end

6.4 Record Page

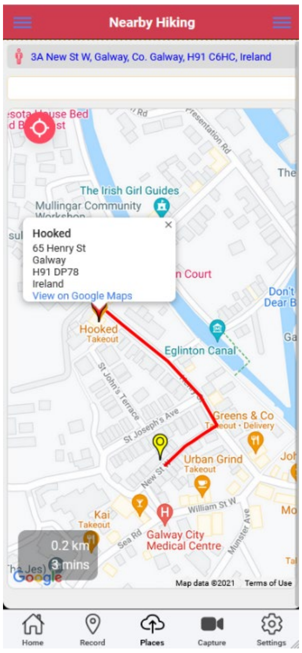
In this page user can save their location, this location will be stored in Firebase database, user will be able to see saved place. I will showcase this feature further down.



53.2703311 , -9.0602354

6.6 Places Page

In this page user can select their location and location they want to go to, when user selects both locations marker will present in redline between locations, user also be able to walking distance and time it would take to get to destination.



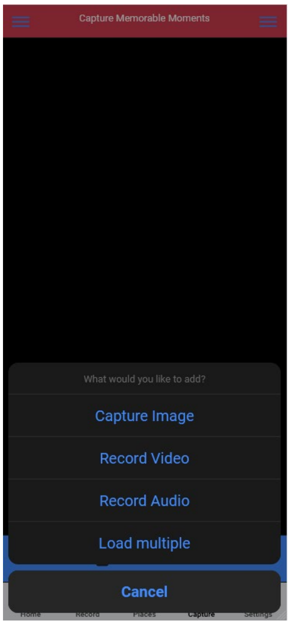
3A New St W, Galway, Co. Galway, H91 C6HC, Ireland

Hooked
65 Henry St
Galway
H91 DP78
Ireland
View on Google Maps

0.2 km
3 mins

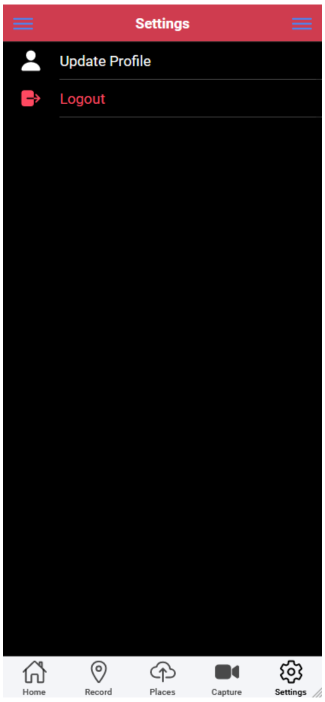
6.7 Capture Page

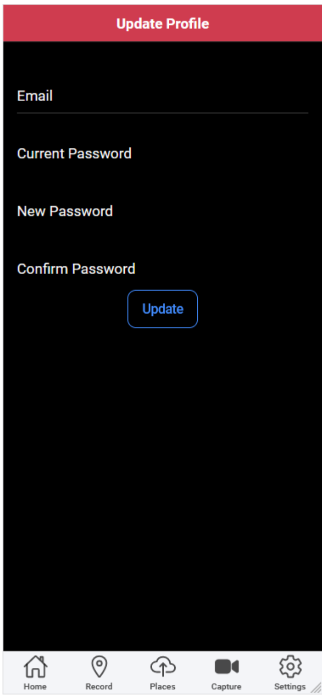
In this page user will be able to capture media elements and save them in the page.



6.8 Setting Page

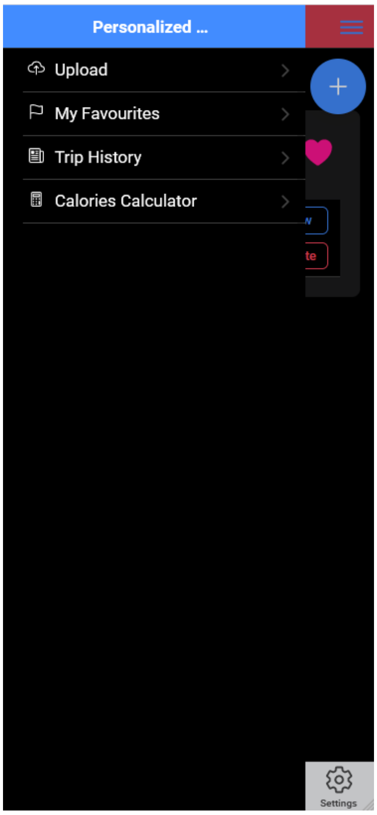
In this page user can update they login information and log out from application. Again, when clicked on update page this will prompt new page.

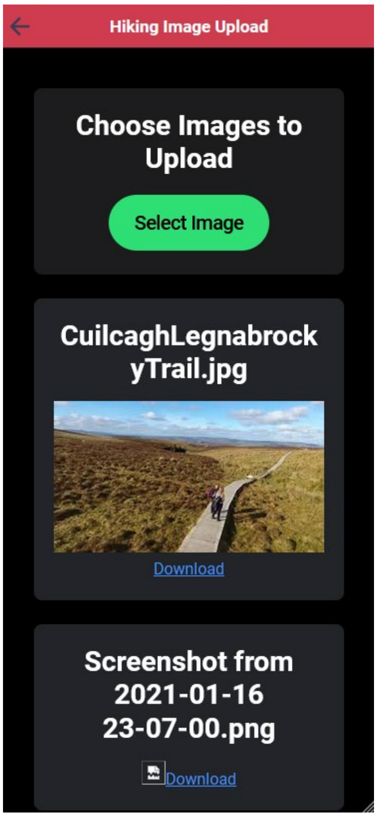




6.9 Personalized Page

Side menu on the left has pages for Upload, Favourites, Trip history and calorie calculator. Upload, favourite and Trip history is saved in firebase database. When user click to saved history trip map opens o show where saved location was. Calorie calculation was extra feature that was added when user feels in the form amount of calorie will be generated for the user.



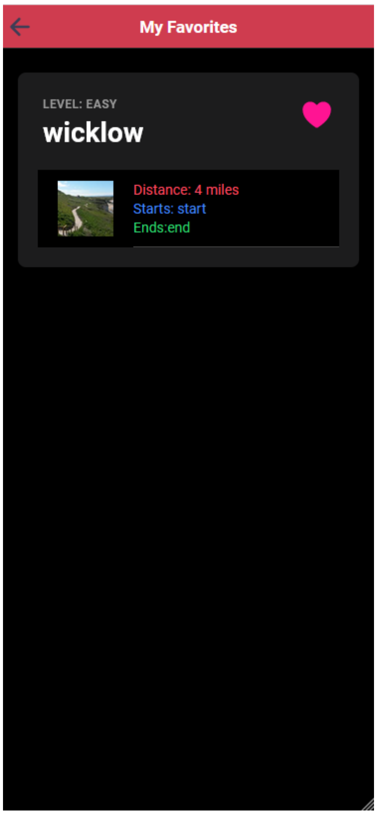


CuilcaghLegnabrockyTrail.jpg

Download

Screenshot from 2021-01-16 23-07-00.png

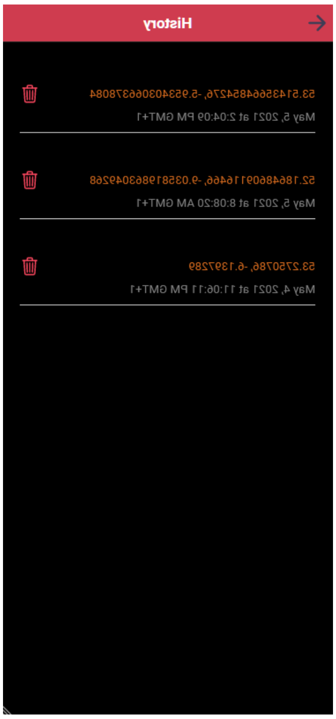
Download



LEVEL: EASY

wicklow

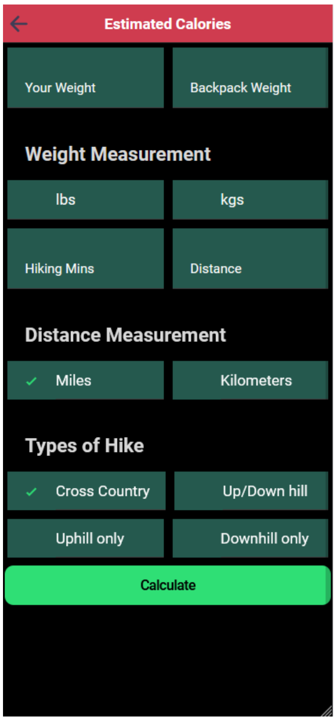
Distance: 4 miles
Starts: start
Ends:end



53.51435664854276, -5.953403066378084
May 5, 2021 at 2:04:09 PM GMT+1

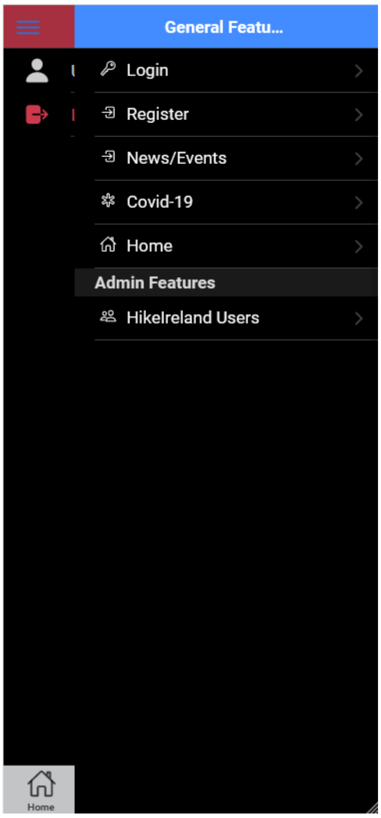
52.18648609116466, -9.035819863049268
May 5, 2021 at 8:08:20 AM GMT+1

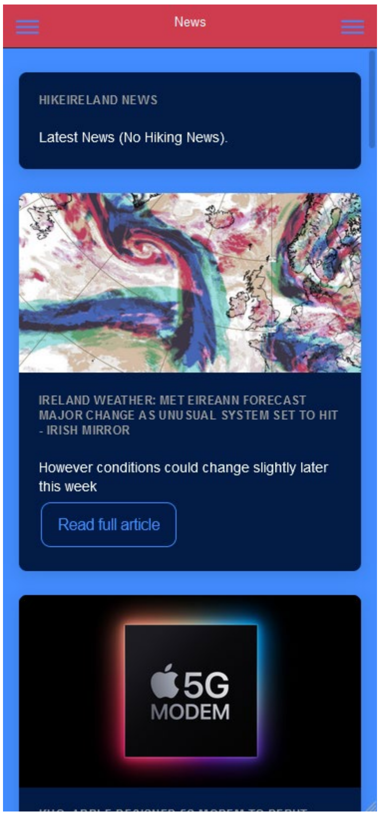
53.2750786, -6.1397289
May 4, 2021 at 11:06:11 PM GMT+1



6.10 General Features Page

Side menu on the right represents pages for login, registration for different user to login or new user to register, News page has live news update for user to catch latest news, Covid-19 page show cases Covi-19 cases in Ireland up to date. Home page will bring back to home page and Hike Ireland users page is admin feature that will show users that are registered in the application.



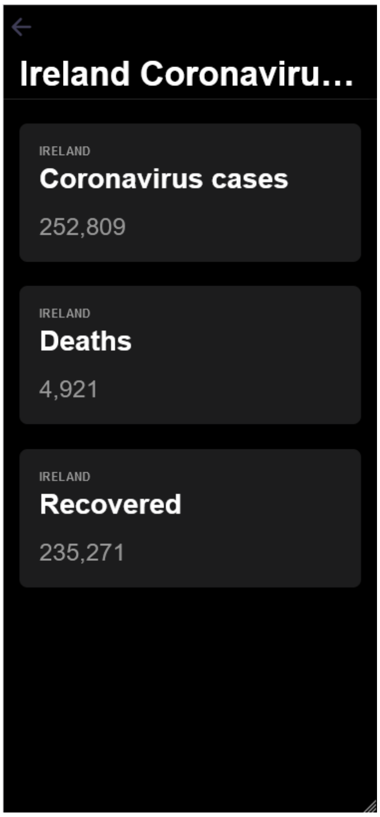


HIKEIRELAND NEWS

Latest News (No Hiking News).

IRELAND WEATHER: MET EIREANN FORECAST MAJOR CHANGE AS UNUSUAL SYSTEM SET TO HIT - IRISH MIRROR

However conditions could change slightly later this week



Ireland Coronaviru...

IRELAND
Coronavirus cases
252,809

IRELAND
Deaths
4,921

IRELAND
Recovered
235,271

6.1.1 Deployment and Hosting

Right now I have decided to host project on my own machine locally, to run application I used Visual Studio Code build in command prompt to navigate between folders and execute application deployment.

When I open project folder in Visual studio code I CD into project folder

- C:\Users\krich\Desktop\GMIT 20-21\hikingapp2.0> cd .\hikingapp-2.0\

And to run application I simple run

- C:\Users\krich\Desktop\GMIT 20-21\hikingapp2.0\hikingapp-2.0> ionic serve

After all the steps completed correctly application should open in your browser on localhost:8000

Chapter 7

System Evaluation

In this chapter I will evaluate the application in four key areas: Robustness, Testing, Limitations< Result & Objectives.

7.1 Robustness

Because of the time constraints, I was unable to thoroughly test the application for robustness; unfortunately, load testing and stress testing were not possible. For the time being, I have not encountered any errors, bugs, or application failure when checking my application.

7.2 Testing

I believe the Application has met the standards I set out at the start of this project through continuous white box testing and routine black box testing. In order to meet these goals, I conducted testing using white box testing in the following manners – Ionic serve testing and Device testing.

7.2.1 Ionic Serve

I used this method of testing during application development because it allowed me to test application in browser while carrying working, to run application in browser I simple cd to root of the application folder and run the commands:

- ionic serve
    - This method of testing was very helpful in the application development because it provided live reload server. What this means is if I made any change in any part of the code, the changes would automatically apply and reload in the browser.
- Ionic serve –lab
    - When this command is run application, changes are previewed in iOS, android and windows platforms.

7.2.2 Device Testing

For iOS testing I used capacitor plugin to build application for iOS device, this helped to give me idea on how application would run on iOS device. But in my circumstances, this wasn't best option for me. Instead I decided to build an Android APK file that I could install on android devise which made testing easier and better for me.

7.3 Limitations

Some of the limitations in the application

7.3.1 Service Limitations

In this application I am using Firebase authentication as a free service. because currently application is only developed as a prototype and plus as a student free price tag is appealing. Application is also hosting locally on my own machine which for this time suits this application as I am are only testing its features and building it as a prototype.

7.3.2 Testing on Apple device

I had apple device available if I wanted to test iOS version of the application. But as we know apple can be very strict with data laws and some cases you need their OS systems to fully test applications, it can be difficult for windows system users to take full advantage of their development tools. For that reason, I tested iOS version of application using capacitor, this was useful, but I still had some issues some of the time. After doing some research to find better solution I saw that lot of people online had similar issues. That's why testing on android devise was much easier.

7.3.3 Results & Objectives

When I first started to set objectives for this application at the start of development, I was bit afraid that I wouldn't have enough time to complete them, but in the end I think I have completed objectives I set out for myself.

In regards if application helped any Hike lovers in assisting their needs can't be conformed as application not yet been lunched on ah app store for people to use as I think this is still working progress.

My foal from the start was to make my application easy and simple to use for anyone who doesn't have that technology know how, as some applications out on the market can be bit doting to use for some people as they are more complex. I believe there can be market for my application once it is on offer to the public.

Chapter 8

Conclusion

I can honestly say this being individual project it was one of the hardest if not hardest task I have took on in my four years of college, especially with worldwide pandemic happening and having to learn to live with it, this can be mentally challenging this defiantly put lot more strain on me and my approach to things had to be adjusted to needs of learning from home and doing online classes.

The scale and complexity of this project has given me better understanding, knowledge and experience of taking on such big project for the future. With scale of this kind of project had its ups and downs, like all project developers in my position have faced problems and so did I but I tried to stay positive and find solution one task at the time. To be honest after my experience as individual developer on this project, next time I would probably try and work in group project.

At the beginning my main objective was to create an application that will be helpful for people who prefer easy to use application.

List of tasks that have been accomplished:

- User can log into their account
- User can create new account
- When user is logged in user can add data to area they wished to.
- User can delete their data
- User can search destinations on google maps
- User can track their location on google maps
- User can save their location from google maps
- User can see their saved location later on if they wish to
- User can calculate their daily calorie intake.
- User can capture multimedia

Working on Ionic 5, NodeJS and Firebase technologies I was able show case my skills gained in my time in GMIT for last four years of studying and to gain experience and learn new ones which will help me out in the future.

8.1 Future Development

I see myself improving and building this application to its full potential by adding some technical upgrades to rich its full potential and adding business plan.

7.1.1 App Development

Get application on the app store – My aim is to get this application on app store soon as possible. First, I would try to have HikeIreland application on android store as its must easier to publish android app. If application gets good feedback than I would consider having application published on Apple store, it's more difficult to publish apps on apple store than Android store.

Marketplace/Business – Again depending on success of application I would consider having low cost monthly subscription fee to be added to the application. Add ads in application to get revenue back from ads market.

Chapter 9 Appendix

Project Source Code Link:

https://github.com/ika25/hikingapp2.0

Bibliography

[1] “Cordova” - https://cordova.apache.org/

[2] “Angulat” - https://angular.io/docs

[3] “NodeJS” - https://nodejs.org/en/about/

[4] “npm” - https://www.npmjs.com/about

[5] “Firebase” - https://enappd.com/blog/firebase-with-ionic-4-hosting-auth-and-db-connection/58/

[6] “Google Maps API” - https://developers.google.com/maps/documentation

[7] “TypeScript” - https://en.wikipedia.org/wiki/TypeScript

[8] “SCSS” - https://skillcrush.com/blog/css/

[9]“HTML” - https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/HTML_basics

[10] “JavaScript” - https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/JavaScript_basics

[11] “Ionic 5” - https://ionicframework.com/blog/announcing-ionic-5/

[12] “project structure” - https://ionicframework.com/docs/v3/intro/tutorial/project-structure/

[13] “Chrome” - https://www.readkong.com/page/porting-mobile-ios-app-to-android-klimb-as-6132143

[14] “Angular Routing” - https://angular.io/tutorial/toh-pt5

[15] “ionic deployment” - https://ionicframework.com/docs/v3/intro/deploying/

[16] “GitHub Project Link” - https://github.com/ika25/hikingapp2.0