Project Report
Take My Stuff App

Omar Zaineh 1088546
Mohammad Fares Aljamous 1088672
Hadi Ramzi Albanna 1088677

Supervised By: Eng. Gasm Elbary

Submitted: June 18, 2024

Contents

List of Figures

List of Tables

Abstract

The Take My Stuff project will test our knowledge of the course by implementing a fully functional application that will allow service users to give away items they no longer need. The purpose of this project is to help people who no longer need their stuff, to give it away instead of throwing it in the trash. This project indirectly promotes recycling and reuse. We used the Visual Studio code to be able to bring our idea to life. Visual Studio code consists of 3 programming languages: HTML (.html file), typescript (.ts file), and CSS (.scss file). These languages were used for different reasons. For instance, HTML and CSS were used to allow the user to view the page in an organized and neat way, while typescript was used to add functionality to each button on the page. This is what we call the MVC framework (model, view, and control). The view is represented by HTML and CSS, while the model and control are represented by typescript. With the help of Firebase and Google Maps, this application become more user-friendly. This report expounds on the programming principles behind the application's functionality, its efficiency in real-world scenarios, and potential areas for future enhancement.

1 Introduction

This report describes a project that intends to create an application that is easy to use and completely functioning, facilitating the exchange of products that one user no longer needs but that another user desires. This project's main goal is to spread the ideas of recycling and reuse, which are crucial for creating a clean, sustainable, and healthy world. The goal of this project is to address and lessen the damaging effects of throwing away perfectly good products on the environment in a society where consumerism frequently results in excessive waste. The app is meant to serve as a platform where people can quickly list things they no longer need so that other people may go through and take ownership of them. By doing this, the initiative promotes a mentality shift toward more environmentally friendly lifestyle choices, gently encouraging people to engage in recycling and reuse without necessitating a big adjustment to their everyday routines. The goal of the platform's extensive features and user-friendly design is to maximize user involvement and participation by simplifying the process as much as possible. The application's broad posting system, which enables users to list products they want to give away and define items they are seeking, is one of its most notable features. This dual feature promotes a community-driven approach to resource sharing by guaranteeing that the platform serves both providers and seekers. Moreover, an availability function in the program lets customers schedule a time to pick up things and gives the precise location of the item donor. This degree of specificity guarantees that the exchange procedure is easy and effective for all concerned parties.The project is significant in ways that go beyond its immediate benefits. It addresses a widespread problem: the needless discarding of appliances that are still completely working and other things that may be used to someone else. The program contributes to waste reduction and promotes resource usage that is more conscientious by providing a channel for these objects to find new homes. It basically expresses the notion that anything may truly be treasured by someone else out of rubbish.This program is also important since it tackles an issue that will have serious consequences for future generations if left uncontrolled. Not only are the amount of garbage we produce and the burden it takes on our planet's resources pressing issues, but they also pose long-term risks to our quality of life and those of future generations. Through the promotion of a reuse and recycling culture, this project supports the larger endeavor of environmental sustainability and conservation. In conclusion, the goal of this project is to

promote a cultural change toward more resource-conscious consumption and management rather than only developing an application. It uses technology to address a significant problem and encourages environmental stewardship in a way that is useful and significant. The project hopes to have a beneficial, long-lasting effect on the environment by getting people to reconsider how they dispose of unwanted goods. This project emphasizes how crucial it is to act now to secure a brighter tomorrow and how our current activities will determine the kind of planet that is left for future generations.

1.1 Motivation

This project aims to inspire change and tackle a pressing issue that impacts all of us, rather than merely developing an application. Our plan will be a big step in the right direction toward fusing ethical technology with the pressing need to save the environment. We are leading a movement towards a more sustainable future, rather than just developing a piece of software. Through our work on this initiative, we are utilizing creativity and innovation to bring about significant change. Our work has an influence that goes much beyond its technical components. It stands for a pledge to protect the environment and a commitment to changing the world for the better. Our application seeks to increase consciousness, support sustainable habits, and motivate people and organizations to take steps to reduce their ecological footprint. We are creating a tool that not only satisfies consumers' immediate requirements but also advances the greater objective of environmental conservation by fusing state-of-the-art technology with a deep sense of responsibility. Our idea of a peaceful coexistence of technology and environment, where improvements in one may benefit and improve the other, is embodied in this project.

1.2 Problem Statement

Nowadays, most people are neglecting environmental conditions. They don't care whether it's healthy or not. An issue that is both environmentally and economically detrimental. Our team aims to address this challenge by designing and implementing this innovative application. This solution will effectively help the environment gain its health slowly. All parties will gain from it: the individual searching for that specific item, the one seeking to get rid of something rather than toss it away, and the environment as recycling

and reuse are encouraged. The implementation of this technology promises a practical, user-friendly, and efficient way to mitigate the widespread issue.

1.3 Literature Review

Each system has advantages and disadvantages, and the decision of which to use is frequently influenced by the project's unique requirements as well as financial limitations. After weighing our choices, we concluded that our application is the best compromise between cost and complexity. It is far more trustworthy and comprehensive than simpler systems on the market, even if it is less expensive and sophisticated than many high-tech competitors. We tested and investigated a lot when creating our application, comparing it to other programs that already existed and had comparable features to ours. This practical experience helped us improve our design by giving us important insights into the advantages and disadvantages of various strategies. Our development approach was impacted by the lessons we learned from both the accomplishments and the faults of these comparable programs, which helped us to create a more effective and user-friendly solution. Fortunately enough we were not the first people to care about the environment. Many different systems already exist that serve the same purpose like FreeCycle, a grassroots movement of people who are giving (and getting) stuff for free in their towns and neighborhoods, OLIO, which is an app that connects neighbors and with local businesses so surplus food and other items can be shared, not thrown away, Next door, which is a hyper-local social networking service for neighborhoods, where users can share information, goods, and services with their neighbors. And a lot more apps that share the same concern.

2 Design

Our app contains multiple pages with various features that will improve the user experience while using the app. We will quickly go over these pages and their key features, giving a brief overview of how each part contributes to the overall effectiveness of the application.

2.1 Home Page

The code generates a smooth, responsive home page for an Ionic-Angular application that is easy to use. Its "Take My Stuff" header is centered, and its grid architecture allows it to adjust to different screen widths if you want to use it on different devices. The "Sign Up" and "Log In" buttons are visible within the centered section of the main text. Angular's 'Router' service is being used to direct users to the appropriate sign-up and login pages with these touch-friendly buttons. For both first-time and frequent users, the setup offers an intuitive and practical interface that is simple to use.

HTML File:

```
<ion-header [translucent]="true">
  <ion-toolbar>
    <ion-title class="ion-text-center">
      Take My Stuff
    </ion-title>
  </ion-toolbar>
</ion-header>

<ion-content [fullscreen]="true" class="ion-padding">
  <ion-grid class="center-content">
    <ion-row class="ion-justify-content-center
        ion-align-items-center" style="height: 100%;">
      <ion-col size-md="6" size-lg="4" size-sm="8" size-xs="10">
          <div class="button-container">
            <ion-button expand="block" shape="round"
                (click)="navis()">Sign Up</ion-button>
            <ion-button expand="block" shape="round" color="medium"
                (click)="navil()">Log In</ion-button>
          </div>
      </ion-col>
    </ion-row>
  </ion-grid>
</ion-content>
```

Typescript File:

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

```
@Component({
  selector: 'app-home',
  templateUrl: 'home.page.html',
  styleUrls: ['home.page.scss'],
})
export class HomePage {

  constructor(private router:Router) {}

  navis() {
    this.router.navigate(['/signin'])
  }

  navil() {
    this.router.navigate(['/login'])
  }
}
```

2.1.1 Sign-Up Page

HTML File:

```
<ion-header [translucent]="true">
  <ion-toolbar>
    <ion-title class="text-center">Sign Up</ion-title>
  </ion-toolbar>
</ion-header>

<ion-content [fullscreen]="true" class="ion-padding">
  <ion-grid class="center-content">
    <ion-row class="ion-justify-content-center
        ion-align-items-center" style="height: 100%;">
      <ion-col size-md="6" size-lg="4" size-sm="8" size-xs="10">
        <ion-item lines="none" color="light" class="input-item">
          <ion-input type="text" placeholder="Name" required
              [(ngModel)]="name"></ion-input>
        </ion-item>
        <ion-item lines="none" color="light" class="input-item">
          <ion-input type="text" placeholder="Email" required
              [(ngModel)]="email"></ion-input>
```

```
        </ion-item>
        <ion-item lines="none" color="light" class="input-item">
          <ion-input id="password" [type]="passwordField"
              placeholder="Password" required [(ngModel)]="password">
            <ion-button fill="clear" slot="end"
                (click)="togglePassword()">
              <ion-icon [name]="passwordField === 'text'? 'eye-off'
                  : 'eye'"></ion-icon>
            </ion-button>
          </ion-input>
        </ion-item>
        <ion-item lines="none" color="light" class="input-item">
          <ion-input id="password" [type]="passwordField"
              placeholder="Confirm Password" required
              [(ngModel)]="confirmPassword">
            <ion-button fill="clear" slot="end"
                (click)="togglePassword()">
              <ion-icon [name]="passwordField === 'text'? 'eye-off'
                  : 'eye'"></ion-icon>
            </ion-button>
          </ion-input>
        </ion-item>

        <div class="button-container">
          <ion-button expand="block" (click)="signup()">Sign
              Up</ion-button>
          <ion-button expand="block" color="medium"
              (click)="navib()">Back</ion-button>
        </div>
      </ion-col>
    </ion-row>
  </ion-grid>
</ion-content>
```

Typescript File:

```
import { Component, OnInit } from '@angular/core';
import { Router } from '@angular/router';
import { getAuth, createUserWithEmailAndPassword } from
    'firebase/auth';
```

```
4
5  @Component({
6    selector: 'app-signin',
7    templateUrl: './signin.page.html',
8    styleUrls: ['./signin.page.scss'],
9  })
10 export class SigninPage implements OnInit {
11   name: string = '';
12   email: string = '';
13   password: string = '';
14   confirmPassword: string = '';
15   auth: any;
16   passwordField: string = 'password';
17
18   constructor(private router: Router) {
19     this.auth = getAuth();
20   }
21
22   ngOnInit() {}
23
24   navib() {
25     this.router.navigate(['']);
26   }
27
28   async signup() {
29     if (this.password !== this.confirmPassword) {
30       alert('Passwords do not match!');
31       return;
32     }
33
34     try {
35       const userCredential = await
             createUserWithEmailAndPassword(this.auth, this.email,
             this.password);
36       console.log('User registered:', userCredential.user);
37       this.router.navigate(['\my']);
38     } catch (error: any) {
39       console.error('Error during sign up:', error);
40       alert(error.message);
41     }
```

```
  }
  togglePassword() {
    this.passwordField = this.passwordField=== 'text'? 'password' :
        'text';
  }
}
```

2.1.2 Login Page

HTML File:

```
<ion-header [translucent]="true">
  <ion-toolbar>
    <ion-title class="text-center">Login</ion-title>
  </ion-toolbar>
</ion-header>

<ion-content [fullscreen]="true" class="ion-padding">
  <ion-grid class="center-content">
    <ion-row class="ion-justify-content-center
        ion-align-items-center" style="height: 100%;">
      <ion-col size-md="6" size-lg="4" size-sm="8" size-xs="10">
        <ion-item lines="none" color="light" class="input-item">
          <ion-input type="text" placeholder="Email" required
              [(ngModel)]="email"></ion-input>
        </ion-item>
        <ion-item lines="none" color="light" class="input-item">
          <ion-input id="password" [type]="passwordField"
              placeholder="Password" required [(ngModel)]="password">
            <ion-button fill="clear" slot="end"
                (click)="togglePassword()">
              <ion-icon [name]="passwordField === 'text'? 'eye-off'
                  : 'eye'"></ion-icon>
            </ion-button>
          </ion-input>
        </ion-item>
        <div class="button-container">
          <ion-button expand="block" (click)="login()">Log
              In</ion-button>
```

```
          <ion-button expand="block" color="medium"
              (click)="navib()">Back</ion-button>
        </div>
      </ion-col>
    </ion-row>
  </ion-grid>
</ion-content>
```

Typescript File:

```
import { Component, OnInit } from '@angular/core';
import { Router } from '@angular/router';
import { getAuth, signInWithEmailAndPassword } from
    'firebase/auth';

@Component({
  selector: 'app-login',
  templateUrl: './login.page.html',
  styleUrls: ['./login.page.scss'],
})
export class LoginPage implements OnInit {
  email: string = '';
  password: string = '';
  auth: any;
  user: any;
  passwordField: string = 'password';

  constructor(private router: Router) {
    this.auth = getAuth();

  }
  navib() {
    this.router.navigate(['']);
  }
  ngOnInit() {}

  async login() {
    try {
      const userCredential = await
          signInWithEmailAndPassword(this.auth, this.email,
```

```
        this.password);
      this.user = userCredential.user;
      console.log('User logged in:', this.user);
      this.router.navigate(['/my']);
    } catch (error) {
      console.error('Error signing in:', error);
      alert('Failed to sign in. Please check your credentials.');
    }
  }
  togglePassword() {
    this.passwordField = this.passwordField === 'text'? 'password'
        : 'text';
  }
}
```

2.2 Profile Page

Here, you may monitor the progress of your requests and see all of your posts. Furthermore, you may keep an eye on whether or not someone has booked your item. With this all-inclusive perspective, you can effectively manage your postings, quickly reply to requests, and monitor the interaction and popularity of your listed goods. You can guarantee a smooth and prosperous experience for yourself as well as your possible tenants or purchasers by keeping yourself updated regarding bookings and interactions.

HTML File:

```
<ion-header [translucent]="true">
  <ion-toolbar>
    <ion-title class="ion-text-center">My Posts And
        Requests</ion-title>
  </ion-toolbar>
</ion-header>

<ion-content [fullscreen]="true">
  <ion-fab slot="fixed" vertical="top" horizontal="start"
      [edge]="true">
    <ion-fab-button (click)="sigOut()" size="small">
      <ion-icon name="log-out"></ion-icon>
    </ion-fab-button>
```

```
12  </ion-fab>
13  <ion-fab slot="fixed" vertical="top" horizontal="end"
        [edge]="true">
14    <ion-fab-button>
15      <ion-icon name="add"></ion-icon>
16    </ion-fab-button>
17    <ion-fab-list side="bottom">
18      <ion-fab-button (click)="naviCP()">
19        <ion-icon name="paper-plane"></ion-icon>
20      </ion-fab-button>
21      <ion-fab-button (click)="naviCR()">
22        <ion-icon name="chatbox"></ion-icon>
23      </ion-fab-button>
24    </ion-fab-list>
25  </ion-fab>
26
27  <ion-card>
28    <ion-card-header>
29      <ion-card-title> My Posts </ion-card-title>
30    </ion-card-header>
31    <ion-list *ngFor="let Item of plist; index as i">
32      <ion-card>
33        <img src="{{Item.base64im}}" />
34        <ion-card-header>
35        <ion-item>
36          <ion-card-title> {{Item.title}} </ion-card-title>
37          <ion-button color="danger" slot="end"
                (click)="removeItemFromList(Item)"
                expand="block">x</ion-button>
38        </ion-item>
39        </ion-card-header>
40      <ion-card-content>
41        Type: {{Item.typee}}
42      </ion-card-content>
43      <ion-card-content>
44        Description: {{Item.description}}
45      </ion-card-content>
46      <ion-card-content>
47        Available Times: {{Item.a_times}}
48      </ion-card-content>
```

```
49      <ion-card-content *ngIf="Item.av == true">
50        Booked: No
51      </ion-card-content>
52      <ion-card-content *ngIf="Item.av == false">
53        Booked: Yes
54      </ion-card-content>
55      </ion-card>
56    </ion-list>
57  </ion-card>
58  <ion-card>
59    <ion-card-header>
60      <ion-card-title> My Requests </ion-card-title>
61    </ion-card-header>
62    <ion-list *ngFor="let rItem of rlist; index as i">
63      <ion-card>
64        <ion-card-header>
65        <ion-item>
66          <ion-card-title> {{rItem.title}} </ion-card-title>
67          <ion-button (click)="removeRequest(rItem)" color="danger"
              slot="end" expand="block">x</ion-button>
68        </ion-item>
69        </ion-card-header>
70        <ion-card-content>
71          Type: {{rItem.typee}}
72        </ion-card-content>
73        <ion-card-content>
74          Description: {{rItem.description}}
75        </ion-card-content>
76        <ion-card-content>
77          Available Times: {{rItem.a_times}}
78        </ion-card-content>
79      </ion-card>
80    </ion-list>
81  </ion-card>
82  <ion-card>
83    <ion-card-header>
84      <ion-card-title> My Bookings </ion-card-title>
85    </ion-card-header>
86    <ion-list *ngFor="let Item of bookings; index as i">
87      <ion-card>
```

```
88          <img src="{{Item.base64im}}" />
89          <ion-card-header>
90          <ion-item>
91            <ion-card-title> {{Item.title}} </ion-card-title>
92          </ion-item>
93          </ion-card-header>
94        <ion-card-content>
95          Type: {{Item.typee}}
96        </ion-card-content>
97        <ion-card-content>
98          Description: {{Item.description}}
99        </ion-card-content>
100       <ion-card-content>
101         Available Times: {{Item.a_times}}
102       </ion-card-content>
103       <ion-button fill="clear"
              (click)="unbook(Item)">Unbook</ion-button>
104       <ion-button fill="clear" (click)="onMap(Item)">On
              Map</ion-button>
105       </ion-card>
106     </ion-list>
107   </ion-card>
108 </ion-content>
109   <ion-tab-bar slot="bottom">
110     <ion-tab-button (click)="navimy()">
111       <ion-icon name="person-circle-outline"></ion-icon>
112       <ion-label>Profile</ion-label>
113     </ion-tab-button>
114     <ion-tab-button (click)="navipost()">
115       <ion-icon name="paper-plane"></ion-icon>
116       <ion-label>Posts</ion-label>
117     </ion-tab-button>
118       <ion-tab-button (click)="navireq()">
119         <ion-icon name="chatbox"></ion-icon>
120         <ion-label>Requests</ion-label>
121     </ion-tab-button>
122     <ion-tab-button (click)="navinotify()">
123       <ion-icon name="notifications-outline"></ion-icon>
124       <ion-label>Feed</ion-label>
125   </ion-tab-button>
```

```
</ion-tab-bar>
```

Typescript File:

```
import { Component, OnInit } from '@angular/core';
import { getAuth, onAuthStateChanged, signOut } from
    "firebase/auth";
import { Router } from '@angular/router';
import { getDatabase, ref, set, push, remove, onValue,
    DataSnapshot, Database } from 'firebase/database';
import { post, requests } from 'src/app/posts/posts.page'
import { Geolocation } from '@capacitor/geolocation';
import { Camera, CameraResultType } from '@capacitor/camera';

@Component({
  selector: 'app-my',
  templateUrl: './my.page.html',
  styleUrls: ['./my.page.scss'],
})
export class MyPage implements OnInit {

  plist: any;
  allplist: any;
  uid: any;
  t: any;
  rlist: any;
  allrlist: any;
  booked: any;
  bookings: any;

  constructor(private router:Router) {
    this.t = post.Itypes
    this.rlist = []
    this.plist = []
    this.allrlist = []
    this.allplist = []
    this.booked = []
    this.bookings = []
   }

```

```
35    naviCP() {
36     this.router.navigate(['/cpost'])
37    }
38
39    naviCR() {
40     this.router.navigate(['/creq'])
41    }
42
43    navimy() {
44     this.router.navigate(['/my'])
45   }
46
47    navipost() {
48     this.router.navigate(['/posts'])
49    }
50
51    navireq() {
52     this.router.navigate(['/req'])
53    }
54
55    navinotify() {
56     this.router.navigate(['/noti'])
57    }
58
59   getMyPosts() {
60     onAuthStateChanged(getAuth(), (user) => {
61       if(user) {
62         this.uid = user.uid
63       }})
64     const db = getDatabase();
65     console.log(this.uid)
66     const Ref = ref(db, this.uid)
67     onValue(Ref, (data) => {this.handleData(data);
          console.log(this.plist)} )
68   }
69
70   handleData(data: DataSnapshot) {
71     this.plist = []
72     data.forEach((ITEM) => {this.handleDataI(ITEM)})
73   }
```

```
74
75  handleDataI(data: any){
76    const Key = data.key
77    const item = data
78
79    if(Key == this.uid) {
80      item.forEach((it: any) => {this.handleData2(it)})
81    }
82  }
83
84  handleData2(data: any) {
85    const Key = data.key
86    const item = data.val()
87    this.plist.push(new post(item.title, item.typee,
          item.description, item.a_times, item.lat, item.lng,
          item.base64im, item.av, Key, item.uid))
88  }
89
90  getPosts() {
91    const db = getDatabase();
92    const Ref = ref(db, 'posts')
93    onValue(Ref, (data) => {this.handleData3(data)} )
94  }
95
96  handleData3(data: DataSnapshot) {
97    this.allplist = []
98    data.forEach((ITEM) => {this.handleData4(ITEM)})
99  }
100
101 handleData4(data: any){
102   const Key = data.key
103   const item = data.val()
104
105   this.allplist.push(new post(item.title, item.typee,
          item.description, item.a_times, item.la, item.ln,
          item.base64im, item.av, Key, item.uid))
106 }
107
108 getBookedPosts() {
109   const db = getDatabase();
```

```
    const Ref = ref(db, 'book')
    onValue(Ref, (data) => {this.handleDataa(data)})
  }

  async handleDataa(data: DataSnapshot) {
    this.booked = []
    await data.forEach((ITEM) => {this.handleDataa2(ITEM)})
      console.log(this.booked)
  }

  async handleDataa2(data: any){
    const Key = data.key
    await data.forEach((data: any) => {this.handleDataa3(data)})
    this.booked.forEach((item:any) => {
      for(let i =0; i < this.plist.length; i++) {
        if((this.plist[i].title == item.title) &&
            (this.plist[i].typee == item.typee) &&
            (this.plist[i].description == item.description)) {
          this.plist[i].av = false;
        }
      }
    })
  }

  handleDataa3(data: any) {
    const Key = data.key
    const item = data.val()
    if(item.uid == this.uid) {
      this.booked.push(new post(item.title, item.typee,
          item.description, item.a_times, item.la, item.ln,
          item.base64im, item.av, Key, item.uid))
      }
  }

  getMyBookings() {
    onAuthStateChanged(getAuth(), (user) => {
      if(user) {
        this.uid = user.uid
        const db = getDatabase();
        const Ref = ref(db, 'book/'+this.uid)
```

```
      this.bookings = []
      onValue(Ref, (data) => {this.handleDataaa4(data)})
    }})
}

handleDataaa4(data:DataSnapshot) {
  const Key = data.key
  data.forEach((item) => {this.handleDataaa5(item)})
}

handleDataaa5(data: any) {
  const Key = data.key
  const item = data.val()
  console.log(item)
  this.bookings.push(new post(item.title, item.typee,
      item.description, item.a_times, item.lat, item.lng,
      item.base64im, item.av, item.key, item.uid))
}

removeItemFromList(item:post) {
  let index: number = this.plist.indexOf(item);
  this.plist.splice(index,1)
  const db = getDatabase();

  remove(ref(db, this.uid+'/'+item.key))
 for(let j = 0; j < this.allplist.length; j++) {
    if((item.title == this.allplist[j].title) && (item.typee ==
        this.allplist[j].typee) && (item.a_times ==
        this.allplist[j].a_times) && (item.description ==
        this.allplist[j].description) && (item.uid ==
        this.allplist[j].uid))
    {
      item = this.allplist[j]
      break;
    }
  }

  let index2: number = this.allplist.indexOf(item);
  this.allplist.splice(index2,1)

```

```
    remove(ref(db, 'posts/'+item.key))
  }

  unbook(item: post) {
    const db = getDatabase();
    let object = new post(item.title, item.typee, item.description,
        item.a_times, item.lat, item.lng, item.base64im, true,
        item.key, item.uid)
    set(ref(db, 'posts/'+item.key), object)

    let index: number = this.bookings.indexOf(item);
    this.bookings.splice(index,1)
    remove(ref(db, 'book/'+this.uid))
  }

  onMap(Item: post) {
    const db = getDatabase();
    set(ref(db, this.uid+"lat"), Item.lat);
    set(ref(db, this.uid+"lng"), Item.lng);
    this.router.navigate(['/mapp']);
  }

getMyRequests() {
  onAuthStateChanged(getAuth(), (user) => {
    if(user) {
      this.uid = user.uid
      const db = getDatabase();
      console.log("r" + this.uid)
      const Ref = ref(db, "r" + this.uid)
      onValue(Ref, (data) => {this.handlerData(data);
          console.log(this.rlist)} )
    }})
}

handlerData(data: DataSnapshot) {
  this.rlist = []
  data.forEach((ITEM) => {this.handlerData2(ITEM)})
}

handlerData2(data: any){
```

```
  const Key = data.key
  const item = data.val()

  this.rlist.push(new requests(item.title, item.typee,
      item.description, item.a_times, item.av, Key, item.uid))
}

getRequests() {
  const db = getDatabase();
  const Ref = ref(db, 'requests')
  onValue(Ref, (data) => {this.handlerData3(data)} )
}

handlerData3(data: DataSnapshot) {
  this.allrlist = []
  data.forEach((ITEM) => {this.handlerData4(ITEM)})
}

handlerData4(data: any){
  const Key = data.key
  const item = data.val()

  this.allrlist.push(new requests(item.title, item.typee,
      item.description, item.a_times, item.av, Key, item.uid))
}

removeRequest(item:any) {
  let index: number = this.rlist.indexOf(item);
  this.rlist.splice(index,1)
  const db = getDatabase();
  remove(ref(db, "r"+this.uid+'/'+item.key))

  for(let j = 0; j < this.allrlist.length; j++) {
    if((item.title == this.allrlist[j].title) && (item.typee ==
        this.allrlist[j].typee) && (item.description ==
        this.allrlist[j].description) && (item.uid ==
        this.allrlist[j].uid))
      {
        item = this.allrlist[j]
      }
```

```
252   }
253
254   let index2: number = this.allrlist.indexOf(item);
255   this.allrlist.splice(index2,1)
256
257   remove(ref(db, 'requests/'+item.key))
258 }
259
260 sigOut() {
261   signOut(getAuth())
262   this.router.navigate(['/home'])
263 }
264
265
266   ngOnInit() {
267     onAuthStateChanged(getAuth(), (user) => {
268       if(user) {
269         this.uid = user.uid
270       }
271       else{
272         this.router.navigate(['/home'])
273       }})
274     this.getMyPosts();
275     this.getPosts();
276     this.getRequests();
277     this.getMyRequests();
278     this.getBookedPosts();
279     this.getMyBookings()
280     }
281 }
```

2.2.1 Creating Posts Page

You may make a post for the item you wish to give away, using the camera capacitor plugin in this part of the app. Writing a precise and descriptive title that describes the item is the first step. To let people know when the item may be delivered or picked up, please include the availability time at the end. After you've finished these steps, other people will be able to see your post and read the information. In addition to your location, so that people can

decide whether to book this item or not, depending on location. This feature was provided through Google Maps services. This simplified procedure helps people locate what they're seeking while also making it simple for you to share things you no longer need.

HTML File:

```
<ion-header [translucent]="true">
  <ion-toolbar>
    <ion-title class="ion-text-center">Create Post</ion-title>
  </ion-toolbar>
</ion-header>
<ion-content [fullscreen]="true">
  <ion-item expand="block">
    <ion-input [(ngModel)]="Title" placeholder="Enter the
        Title"></ion-input>
  </ion-item>
  <ion-item>
    <ion-select label="Default label" placeholder="Item Type"
        [(ngModel)]="Type">
      <ion-select-option *ngFor="let type of t; index as i"
          value="{{t[i]}}">{{t[i]}}</ion-select-option>
    </ion-select>
  </ion-item>
  <ion-item expand="block">
    <ion-input [(ngModel)]="desc" placeholder="Enter the
        Description"></ion-input>
  </ion-item>
  <ion-item expand="block">
    <ion-input [(ngModel)]="time" placeholder="Enter the
        Time"></ion-input>
  </ion-item>
  <ion-button (click)="takepic()" expand="block">Take A
      Picture</ion-button>
  <img src="{{base64im}}" />
  <ion-button (click)="createPost()" expand="block">Create The
      Post</ion-button>
</ion-content>
  <ion-tab-bar slot="bottom">
    <ion-tab-button (click)="navimy()">
      <ion-icon name="person-circle-outline"></ion-icon>
```

```
      <ion-label>Profile</ion-label>
    </ion-tab-button>
    <ion-tab-button (click)="navipost()">
      <ion-icon name="paper-plane"></ion-icon>
      <ion-label>Posts</ion-label>
    </ion-tab-button>
      <ion-tab-button (click)="navireq()">
        <ion-icon name="chatbox"></ion-icon>
        <ion-label>Requests</ion-label>
    </ion-tab-button>
    <ion-tab-button (click)="navinotify()">
      <ion-icon name="notifications-outline"></ion-icon>
      <ion-label>Feed</ion-label>
  </ion-tab-button>
</ion-tab-bar>
```

Typescript File:

```
import { Component, OnInit } from '@angular/core';
import { getAuth, onAuthStateChanged } from "firebase/auth";
import { Router } from '@angular/router';
import { getDatabase, ref, set, push, remove, onValue,
    DataSnapshot, Database } from 'firebase/database';
import { post, requests } from 'src/app/posts/posts.page'
import { Geolocation } from '@capacitor/geolocation';
import { Camera, CameraResultType } from '@capacitor/camera';

@Component({
  selector: 'app-cpost',
  templateUrl: './cpost.page.html',
  styleUrls: ['./cpost.page.scss'],
})
export class CpostPage implements OnInit {

  Title: any;
  Type: any;
  desc: any;
  time: any;
  lat: any;
  lng: any;
```

```
22  base64im: any;
23  plist: any;
24  uid: any;
25  t: any;
26
27  constructor(private router:Router) {
28    this.t = post.Itypes
29    this.plist = []
30   }
31
32   navimy() {
33    this.router.navigate(['/my'])
34  }
35
36  navipost() {
37    this.router.navigate(['/posts'])
38   }
39
40   navireq() {
41   this.router.navigate(['/req'])
42   }
43
44   navinotify() {
45    this.router.navigate(['/noti'])
46   }
47
48  createPost() {
49    if(this.plist.length < 3) {
50      let object = new post(this.Title, this.Type, this.desc,
          this.time, this.lat, this.lng, this.base64im, true, '',
          this.uid);
51      const db = getDatabase();
52      push(ref(db, 'posts'), object);
53      push(ref(db, this.uid), object);
54      this.navimy()
55    }
56    else {
57      console.log("Max Number Of Posts Reached")
58    }
59  }
```

```
60
61  async getPosition() {
62    const cords = await Geolocation.getCurrentPosition()
63    this.lat = cords.coords.latitude
64    this.lng = cords.coords.longitude
65   }
66
67  async takepic() {
68      let im = await Camera.getPhoto({
69        quality:90,
70        allowEditing:false,
71        resultType:CameraResultType.Base64
72        });
73      this.base64im = ''+ im.base64String;
74   }
75
76   getMyPosts() {
77    onAuthStateChanged(getAuth(), (user) => {
78      if(user) {
79        this.uid = user.uid
80      }})
81    const db = getDatabase();
82    console.log(this.uid)
83    const Ref = ref(db, this.uid)
84    onValue(Ref, (data) => {this.handleData(data);
          console.log(this.plist)} )
85  }
86
87  handleData(data: DataSnapshot) {
88    this.plist = []
89    data.forEach((ITEM) => {this.handleDataI(ITEM)})
90  }
91
92  handleDataI(data: any){
93    const Key = data.key
94    const item = data
95
96    if(Key == this.uid) {
97      item.forEach((it: any) => {this.handleData2(it)})
98    }
```

```
  }

  handleData2(data: any) {
    const Key = data.key
    const item = data.val()
    this.plist.push(new post(item.title, item.typee,
        item.description, item.a_times, item.la, item.ln,
        item.base64im, item.av, Key, item.uid))
  }

  ngOnInit() {
    onAuthStateChanged(getAuth(), (user) => {
      if(user) {
        this.uid = user.uid
      }
      else{
        this.router.navigate(['/home'])
      }})
    this.getMyPosts()
    this.getPosition()
  }
}
```

2.2.2 Creating Request Page

This app feature gives you the ability to post specific requests. Once you give the app specifics about what you're searching for, it can suggest articles that are similar to your description. By making sure that the content you view are extremely relevant to your requirements and interests, our personalized recommendation system aims to improve your overall user experience. This function facilitates finding the exact item you need, whether you're looking for anything specific to buy. To make sure you always have access to the most up-to-date and appropriate alternatives, the app constantly refreshes and improves these suggestions based on the most recent listings and availability. Utilizing this feature will allow you to save time and effort. Increasing the efficiency of your search.

HTML File:

```
<ion-header [translucent]="true">
```

```
  <ion-toolbar>
    <ion-title class="ion-text-center">Create Request</ion-title>
  </ion-toolbar>
</ion-header>
<ion-content [fullscreen]="true">
  <ion-item expand="block">
    <ion-input [(ngModel)]="rTitle" placeholder="Enter the
        Title"></ion-input>
  </ion-item>
  <ion-item>
    <ion-select label="Default label" placeholder="Item Type"
        [(ngModel)]="rType">
      <ion-select-option *ngFor="let type of t; index as i"
          value="{{t[i]}}">{{t[i]}}</ion-select-option>
    </ion-select>
  </ion-item>
  <ion-item expand="block">
    <ion-input [(ngModel)]="rdesc" placeholder="Enter the
        Description"></ion-input>
  </ion-item>
  <ion-item expand="block">
    <ion-input [(ngModel)]="rtime" placeholder="Enter the
        Time"></ion-input>
  </ion-item>
    <ion-button (click)="createRequest()" expand="block">Create The
        Request</ion-button>
  </ion-content>
    <ion-tab-bar slot="bottom">
      <ion-tab-button (click)="navimy()">
        <ion-icon name="person-circle-outline"></ion-icon>
        <ion-label>Profile</ion-label>
      </ion-tab-button>
      <ion-tab-button (click)="navipost()">
        <ion-icon name="paper-plane"></ion-icon>
        <ion-label>Posts</ion-label>
      </ion-tab-button>
        <ion-tab-button (click)="navireq()">
          <ion-icon name="chatbox"></ion-icon>
          <ion-label>Requests</ion-label>
      </ion-tab-button>
```

```
      <ion-tab-button (click)="navinotify()">
        <ion-icon name="notifications-outline"></ion-icon>
        <ion-label>Feed</ion-label>
    </ion-tab-button>
</ion-tab-bar>
```

Typescript File:

```
import { Component, OnInit } from '@angular/core';
import { getAuth, onAuthStateChanged } from "firebase/auth";
import { Router } from '@angular/router';
import { getDatabase, ref, set, push, remove, onValue,
    DataSnapshot, Database } from 'firebase/database';
import { post, requests } from 'src/app/posts/posts.page'
import { Geolocation } from '@capacitor/geolocation';
import { Camera, CameraResultType } from '@capacitor/camera';

@Component({
  selector: 'app-creq',
  templateUrl: './creq.page.html',
  styleUrls: ['./creq.page.scss'],
})
export class CreqPage implements OnInit {

  uid: any;
  t: any;
  rTitle: any;
  rType: any;
  rdesc: any;
  rtime: any;
  rlist: any;

  constructor(private router:Router) {
    this.t = post.Itypes
    this.rlist = []
   }

   navimy() {
    this.router.navigate(['/my'])
  }
```

```
32
33    navipost() {
34      this.router.navigate(['/posts'])
35     }
36
37     navireq() {
38     this.router.navigate(['/req'])
39     }
40
41     navinotify() {
42      this.router.navigate(['/noti'])
43     }
44
45     createRequest() {
46      if(this.rlist.length < 3) {
47        let object = new requests(this.rTitle, this.rType,
            this.rdesc, this.rtime, true, '', this.uid);
48        const db = getDatabase();
49        push(ref(db, 'requests'), object);
50        push(ref(db, "r" + this.uid), object);
51        this.navimy()
52      }
53      else {
54        console.log("Max Number Of Requests Reached")
55      }
56  }
57
58  getMyRequests() {
59    onAuthStateChanged(getAuth(), (user) => {
60      if(user) {
61        this.uid = user.uid
62        const db = getDatabase();
63        console.log("r" + this.uid)
64        const Ref = ref(db, "r" + this.uid)
65        onValue(Ref, (data) => {this.handlerData(data);
            console.log(this.rlist)} )
66      }
67      else{
68        this.router.navigate(['/home'])
69      }})
```

```
}

handlerData(data: DataSnapshot) {
  this.rlist = []
  data.forEach((ITEM) => {this.handlerData2(ITEM)})
}

handlerData2(data: any){
  const Key = data.key
  const item = data.val()

  this.rlist.push(new requests(item.title, item.typee,
      item.description, item.a_times, item.av, Key, item.uid))
}

  ngOnInit() {
    this.getMyRequests();
  }
}
```

2.2.3 Posts Page

On the post page, where the user can browse other people's products. While viewing this page you can view items with their descriptions and the availability of the user who posted the item. In addition, a booking button that is used to book the item to inform the person who has created this post that someone is coming to pick it up. Moreover, to make things easier for the user we added the on-map function, which tells you the exact location of the product using the Google Maps services.

HTML File:

```
<ion-header [translucent]="true">
  <ion-toolbar>
    <ion-title class="ion-text-center">posts</ion-title>
  </ion-toolbar>
</ion-header>

<ion-content [fullscreen]="true">
  <ion-fab slot="fixed" vertical="top" horizontal="start"
```

```
      [edge]="true">
    <ion-fab-button (click)="sigOut()" size="small">
      <ion-icon name="log-out"></ion-icon>
    </ion-fab-button>
  </ion-fab>

  <ion-card>
    <ion-card-header>
      <ion-card-title> All Posts </ion-card-title>
    </ion-card-header>

    <ion-list *ngFor="let Item of plist; index as i">
      <ion-card *ngIf="Item.av == true">
        <img src="{{Item.base64im}}" />
        <ion-card-header>
          <ion-card-title> {{Item.title}} </ion-card-title>
        </ion-card-header>

      <ion-card-content>
        {{Item.typee}}
      </ion-card-content>
      <ion-card-content>
        {{Item.description}}
      </ion-card-content>
      <ion-card-content>
        {{Item.a_times}}
      </ion-card-content>
      <ion-button fill="clear"
          (click)="book(Item)">Book</ion-button>
      <ion-button fill="clear" (click)="onMap(Item)">On
          Map</ion-button>
      </ion-card>
    </ion-list>
  </ion-card>
</ion-content>

  <ion-tab-bar slot="bottom">
    <ion-tab-button (click)="navimy()">
      <ion-icon name="person-circle-outline"></ion-icon>
      <ion-label>Profile</ion-label>
```

```
    </ion-tab-button>
    <ion-tab-button (click)="navipost()">
      <ion-icon name="paper-plane"></ion-icon>
      <ion-label>Posts</ion-label>
    </ion-tab-button>
    <ion-tab-button (click)="navireq()">
        <ion-icon name="chatbox"></ion-icon>
        <ion-label>Requests</ion-label>
    </ion-tab-button>
    <ion-tab-button (click)="navinotify()">
      <ion-icon name="notifications-outline"></ion-icon>
      <ion-label>Feed</ion-label>
    </ion-tab-button>
  </ion-tab-bar>
```

Typescript File:

```
import { Component, OnInit } from '@angular/core';
import { getAuth, onAuthStateChanged, signOut } from
    "firebase/auth";
import { Router } from '@angular/router';
import {getDatabase, ref, set, push, remove, onValue,
    DataSnapshot, Database} from 'firebase/database';

export class post {
  static Itypes: string[] = ['Desk' , 'Table' , 'Kitechen
      Applaiance' , 'Bed' , 'Shelf' , 'Drawer' , 'Closet' , 'Sofa',
      'Chair']
  title: any;
  typee: string;
  description: string;
  a_times: string[];
  av: any;
  lat: number;
  lng: number;
  base64im: any;
  key: any;
  uid: any;

  constructor(ti: any, t: string, d: string, a: string[], la: any,
```

```
      ln: any, im: any, av: boolean, key:any, uid: any) {
    this.title = ti;
    this.typee = t;
    this.description = d;
    this.a_times = a;
    this.lat = la
    this.lng = ln
    this.base64im = im;
    this.av = av;
    this.key = key;
    this.uid = uid;
   }
}

export class requests {
  static Itypes: string[] = ['Desk' , 'Table' , 'Kitechen
      Applaiance' , 'Bed' , 'Shelf' , 'Drawer' , 'Closet' , 'Sofa',
      'Chair']
  title: any;
  typee: string;
  description: string;
  key: any;
  a_times: any;
  av: any;
  uid: any;

  constructor(ti: any, t: string, d: string, a:any, av: any,
      key:any, uid: any) {
    this.title = ti;
    this.typee = t;
    this.description = d;
    this.key = key;
    this.a_times = a;
    this.av = av;
    this.uid = uid
   }
}

@Component({
  selector: 'app-posts',
```

```
  templateUrl: './posts.page.html',
  styleUrls: ['./posts.page.scss'],
})
export class PostsPage implements OnInit {

  uid: any;
  plist: any;

  constructor(private router:Router) { }

  uploadpost() {}

  navimy() {
    this.router.navigate(['/my'])
  }

  navipost() {
    this.router.navigate(['/posts'])
   }

   navireq() {
   this.router.navigate(['/req'])
   }

   navinotify() {
    this.router.navigate(['/noti'])
   }

  getPosts() {
    const db = getDatabase();
    const Ref = ref(db, 'posts')
    onValue(Ref, (data) => {this.handleData(data)} )
  }

  handleData(data: DataSnapshot) {
    this.plist = []
    data.forEach((ITEM) => {this.handleDataI(ITEM)})
  }

  handleDataI(data: any){
```

```
96      const Key = data.key
97      const item = data.val()
98      console.log(Key)
99
100     this.plist.push(new post(item.title, item.typee,
            item.description, item.a_times, item.lat, item.lng,
            item.base64im, item.av, Key, item.uid))
101     console.log(this.plist)
102   }
103
104   onMap(Item: post) {
105     const db = getDatabase();
106     set(ref(db, this.uid+"lat"), Item.lat);
107     set(ref(db, this.uid+"lng"), Item.lng);
108     this.router.navigate(['/mapp']);
109   }
110
111   sigOut() {
112     signOut(getAuth())
113     this.router.navigate([''])
114   }
115
116   book(item: post) {
117     if(item.uid != this.uid) {
118     const db = getDatabase();
119     let object = new post(item.title, item.typee, item.description,
            item.a_times, item.lat, item.lng, item.base64im, false,
            item.key, item.uid)
120     set(ref(db, 'posts/'+item.key), object)
121     push(ref(db, 'book/'+this.uid), object)
122     this.getPosts()
123     }
124     else {
125       console.log("You Cannot Book Your Items")
126     }
127   }
128
129   ngOnInit() {
130     onAuthStateChanged(getAuth(), (user) => {
131       if(user) {
```

```
        this.uid = user.uid
        console.log(this.uid)
      }
      else{
        this.router.navigate(['/home'])
      }})
    this.getPosts()
  }
}
```

2.3 Requests Page

You can request a specific item on this page, along with a thorough description of what you're searching for. Your request will be updated regularly on the feed page as soon as it is submitted. As relevant things become available, the feed will automatically match your request with other users' postings to make sure you see them. By quickly and effectively matching you with possible matches, this tool helps you locate exactly what you're looking for depending on the parameters you've set.

HTML File:

```
<ion-header [translucent]="true">
  <ion-toolbar>
    <ion-title>Request Page</ion-title>
  </ion-toolbar>
</ion-header>

<ion-content [fullscreen]="true">
  <ion-fab slot="fixed" vertical="top" horizontal="start"
      [edge]="true">
    <ion-fab-button (click)="sigOut()" size="small">
      <ion-icon name="log-out"></ion-icon>
    </ion-fab-button>
  </ion-fab>

  <ion-card>
    <ion-card-header>
      <ion-card-title> All Requests </ion-card-title>
    </ion-card-header>
```

```
18
19      <ion-list *ngFor="let Item of allrlist; index as i">
20        <ion-card>
21          <ion-card-header>
22            <ion-card-title> {{Item.title}} </ion-card-title>
23          </ion-card-header>
24
25        <ion-card-content>
26          {{Item.typee}}
27        </ion-card-content>
28        <ion-card-content>
29          {{Item.description}}
30        </ion-card-content>
31        <ion-card-content>
32          {{Item.a_times}}
33        </ion-card-content>
34        </ion-card>
35      </ion-list>
36    </ion-card>
37  </ion-content>
38
39    <ion-tab-bar slot="bottom">
40      <ion-tab-button (click)="navimy()">
41        <ion-icon name="person-circle-outline"></ion-icon>
42        <ion-label>Profile</ion-label>
43      </ion-tab-button>
44      <ion-tab-button (click)="navipost()">
45        <ion-icon name="paper-plane"></ion-icon>
46        <ion-label>Posts</ion-label>
47      </ion-tab-button>
48        <ion-tab-button (click)="navireq()">
49          <ion-icon name="chatbox"></ion-icon>
50          <ion-label>Requests</ion-label>
51      </ion-tab-button>
52      <ion-tab-button (click)="navinotify()">
53        <ion-icon name="notifications-outline"></ion-icon>
54        <ion-label>Feed</ion-label>
55    </ion-tab-button>
56  </ion-tab-bar>
```

Typescript File:

```
import { Component, OnInit } from '@angular/core';
import { getAuth, onAuthStateChanged, signOut } from
    "firebase/auth";
import { Router } from '@angular/router';
import { getDatabase, ref, set, push, remove, onValue,
    DataSnapshot, Database } from 'firebase/database';
import { post, requests } from 'src/app/posts/posts.page'
import { Geolocation } from '@capacitor/geolocation';
import { Camera, CameraResultType } from '@capacitor/camera';

@Component({
  selector: 'app-req',
  templateUrl: './req.page.html',
  styleUrls: ['./req.page.scss'],
})
export class ReqPage implements OnInit {

  uid: any;
  allrlist: any;

  constructor(private router:Router) {
    this.allrlist = []
   }

   getRequests() {
    const db = getDatabase();
    const Ref = ref(db, 'requests')
    onValue(Ref, (data) => {this.handlerData3(data)} )
  }

  handlerData3(data: DataSnapshot) {
    this.allrlist = []
    data.forEach((ITEM) => {this.handlerData4(ITEM)})
  }

  handlerData4(data: any){
    const Key = data.key
    const item = data.val()
```

```
37
38      this.allrlist.push(new requests(item.title, item.typee,
            item.description, item.a_times, item.av, Key, item.uid))
39    }
40
41    ngOnInit() {
42      onAuthStateChanged(getAuth(), (user) => {
43        if(user) {
44          this.uid = user.uid
45          console.log(this.uid)
46        }
47        else{
48          this.router.navigate(['/home'])
49        }})
50      this.getRequests()
51    }
52
53    navimy() {
54      this.router.navigate(['/my'])
55    }
56
57    navipost() {
58      this.router.navigate(['/posts'])
59     }
60
61     navireq() {
62     this.router.navigate(['/req'])
63     }
64
65     navinotify() {
66      this.router.navigate(['/noti'])
67     }
68
69     sigOut() {
70      signOut(getAuth())
71      this.router.navigate([''])
72    }
73  }
```

2.4 Feed Page

You will get customized post recommendations on the feed page based on the requests you have placed. You can always rely on these suggestions to be dynamically updated to correspond with item availability, giving you access to the most current and important information. This tool seeks to improve your browsing experience by instantly matching you with products that fit your interests and needs. Furthermore, the feed will automatically update to reflect changes in item availability, giving you a new and relevant alternative to peruse. In this manner, you may remain up to date and choose wisely when it comes to possible purchases or rentals. [1]

HTML File:

```
<ion-header [translucent]="true">
  <ion-toolbar>
    <ion-title>Feed</ion-title>
  </ion-toolbar>
</ion-header>

<ion-content [fullscreen]="true">
  <ion-fab slot="fixed" vertical="top" horizontal="start"
      [edge]="true">
    <ion-fab-button (click)="sigOut()" size="small">
      <ion-icon name="log-out"></ion-icon>
    </ion-fab-button>
  </ion-fab>

  <ion-card>

    <ion-card-header>
      <ion-card-title> Recommended Posts </ion-card-title>
    </ion-card-header>

    <ion-list *ngFor="let Item of reclist; index as i">
      <ion-card *ngIf="Item.av == true">
        <img src="{{Item.base64im}}" />
        <ion-card-header>
        <ion-item>
          <ion-card-title> {{Item.title}} </ion-card-title>
        </ion-item>
```

```
      </ion-card-header>

    <ion-card-content>
      Type: {{Item.typee}}
    </ion-card-content>
    <ion-card-content>
      Description: {{Item.description}}
    </ion-card-content>
    <ion-card-content>
      Available Times: {{Item.a_times}}
    </ion-card-content>
    </ion-card>
  </ion-list>
 </ion-card>
</ion-content>

 <ion-tab-bar slot="bottom">
  <ion-tab-button (click)="navimy()">
    <ion-icon name="person-circle-outline"></ion-icon>
    <ion-label>Profile</ion-label>
  </ion-tab-button>
  <ion-tab-button (click)="navipost()">
    <ion-icon name="paper-plane"></ion-icon>
    <ion-label>Posts</ion-label>
  </ion-tab-button>
    <ion-tab-button (click)="navireq()">
      <ion-icon name="chatbox"></ion-icon>
      <ion-label>Requests</ion-label>
  </ion-tab-button>
  <ion-tab-button (click)="navinotify()">
    <ion-icon name="notifications-outline"></ion-icon>
    <ion-label>Feed</ion-label>
 </ion-tab-button>
</ion-tab-bar>
```

Typescript File:

```
import { Component, OnInit } from '@angular/core';
import { getAuth, onAuthStateChanged, signOut } from
   "firebase/auth";
```

```
import { Router } from '@angular/router';
import { getDatabase, ref, set, push, remove, onValue,
    DataSnapshot, Database } from 'firebase/database';
import { post, requests } from 'src/app/posts/posts.page'
import { Geolocation } from '@capacitor/geolocation';
import { Camera, CameraResultType } from '@capacitor/camera';

@Component({
  selector: 'app-noti',
  templateUrl: './noti.page.html',
  styleUrls: ['./noti.page.scss'],
})
export class NotiPage implements OnInit {

  uid:any;
  allplist: any;
  rlist: any;
  reclist: any;

  constructor(private router:Router) {
    this.allplist = []
    this.reclist = []
    this.rlist = []
   }

  async ngOnInit() {
    onAuthStateChanged(getAuth(), (user) => {
      if(user) {
        this.uid = user.uid
        this.getPosts()
        this.getMyRequests()
      }
      else{
        this.router.navigate(['/home'])
      }})
  }

  navimy() {
    this.router.navigate(['/my'])
  }
```

```
42
43   navipost() {
44     this.router.navigate(['/posts'])
45    }
46
47    navireq() {
48    this.router.navigate(['/req'])
49    }
50
51    navinotify() {
52     this.router.navigate(['/noti'])
53    }
54
55    sigOut() {
56     signOut(getAuth())
57     this.router.navigate([''])
58   }
59
60  getPosts() {
61     const db = getDatabase();
62     const Ref = ref(db, 'posts')
63     onValue(Ref, (data) => {this.handleData3(data)} )
64   }
65
66   handleData3(data: DataSnapshot) {
67     this.allplist = []
68     data.forEach((ITEM) => {this.handleData4(ITEM)})
69   }
70
71   handleData4(data: any){
72     const Key = data.key
73     const item = data.val()
74
75     this.allplist.push(new post(item.title, item.typee,
         item.description, item.a_times, item.la, item.ln,
         item.base64im, item.av, Key, item.uid))
76   }
77
78     getMyRequests() {
79       onAuthStateChanged(getAuth(), (user) => {
```

```
80      if(user) {
81        this.uid = user.uid
82        const db = getDatabase();
83        console.log("r" + this.uid)
84        const Ref = ref(db, "r" + this.uid)
85        onValue(Ref, (data) => {this.handlerData(data);
              console.log(this.rlist)
86          this.rlist.forEach((req: requests) => {
87          console.log(req)
88          for(let i = 0; i < this.allplist.length; i++) {
89            let inn = false;
90            for(let j = 0; i < this.reclist.length; i++) {
91              if(this.reclist[j] == this.allplist[i]) {
92                inn = true
93              }
94            }
95            if (req.typee == this.allplist[i].typee && !inn &&
                  req.uid != this.allplist[i].uid){
96              this.reclist.push(this.allplist[i]);
97            }
98          }
99        })} )
100
101     }})
102   }
103
104   handlerData(data: DataSnapshot) {
105     this.rlist = []
106     data.forEach((ITEM) => {this.handlerData2(ITEM)})
107   }
108
109   handlerData2(data: any){
110     const Key = data.key
111     const item = data.val()
112
113     this.rlist.push(new requests(item.title, item.typee,
          item.description, item.a_times, item.av, Key, item.uid))
114   }
115 }
```

2.5 System Overview

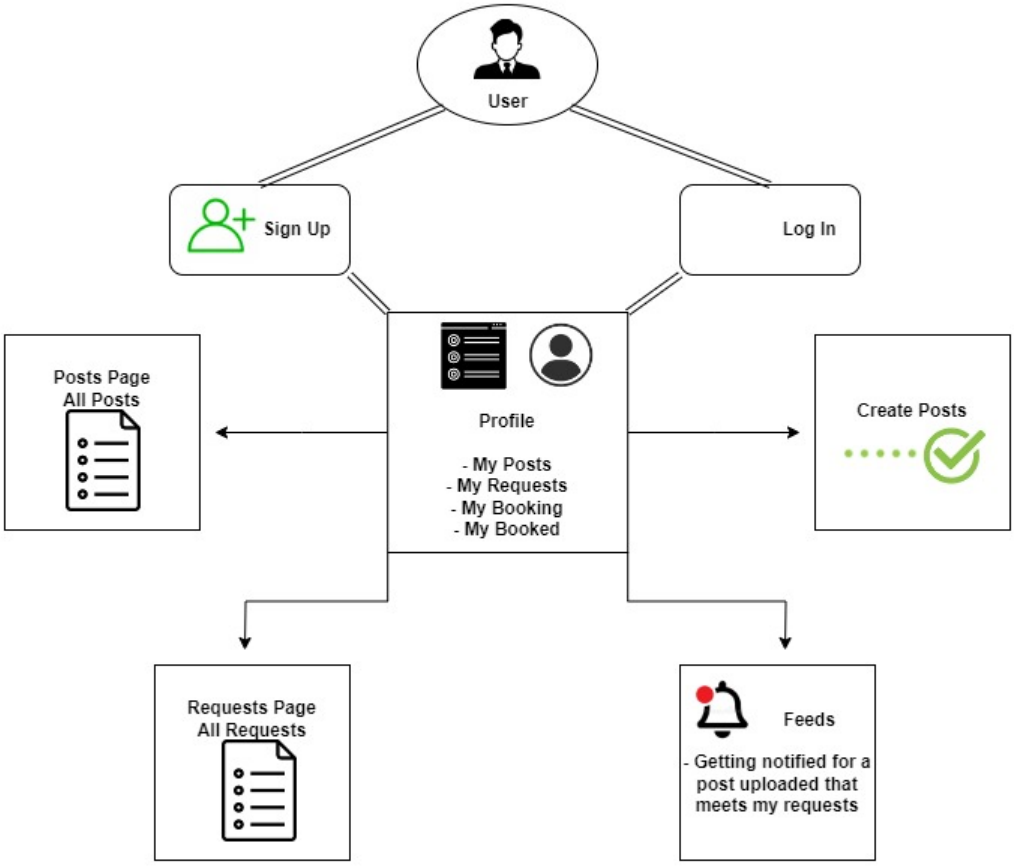

Figure 1: Take My Stuff Overview.

3 Experimental Results

3.1 Results

We ended up with a fully functional application that contains a sign-up and login page. In addition to a page where you can create a post with your availability time for other users to book it. Moreover, an on-map feature that routes you to the location of the product, and a feed that is updated according to your requirements. More features will be represented in a video. [2]

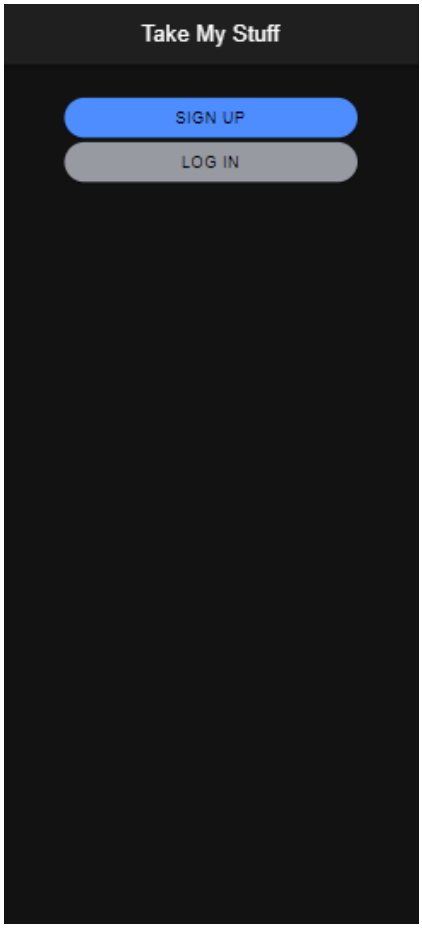

Figure 2: Take My Stuff Pages.

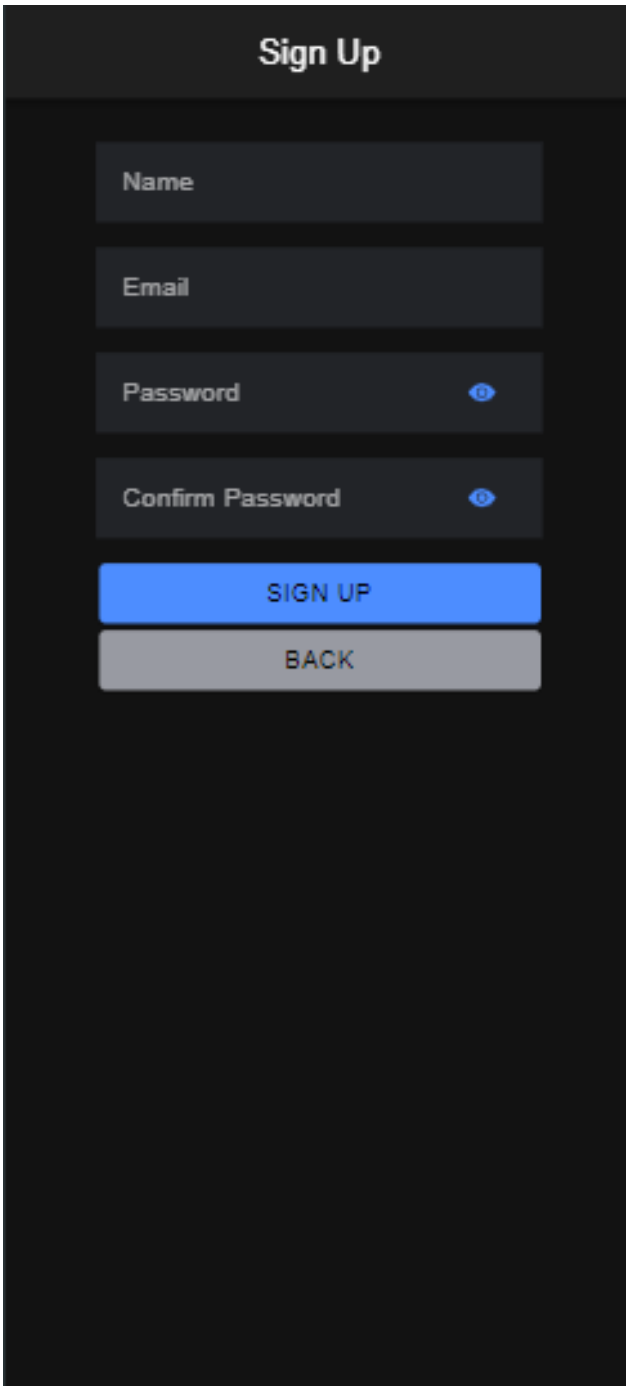

Figure 3: Take My Stuff Pages.

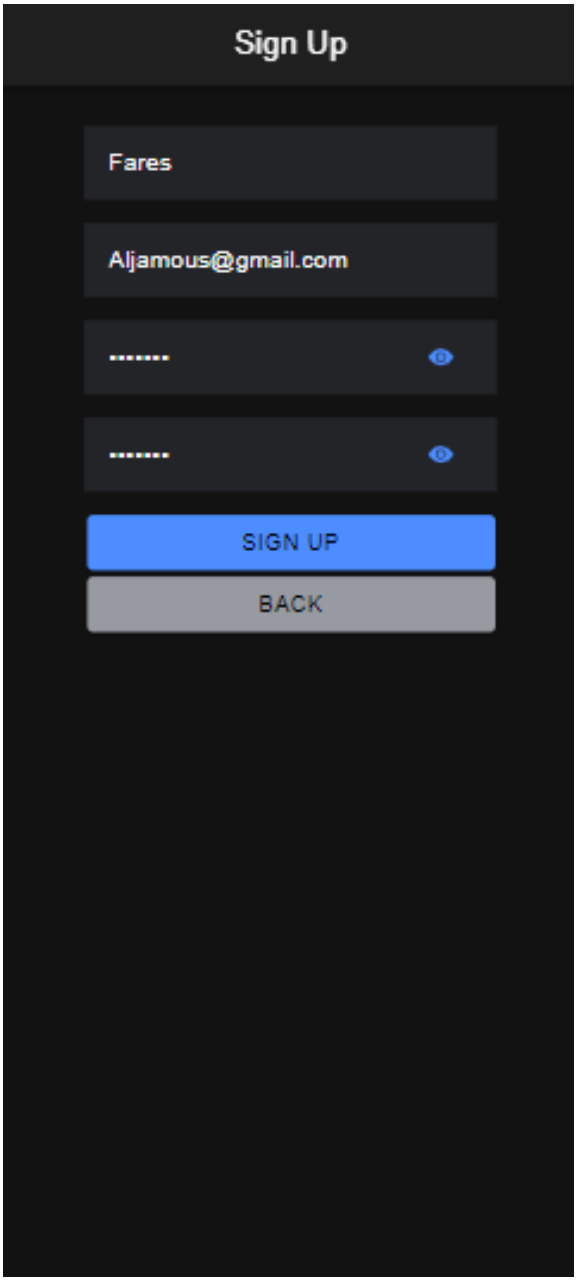

Figure 4: Take My Stuff Pages.

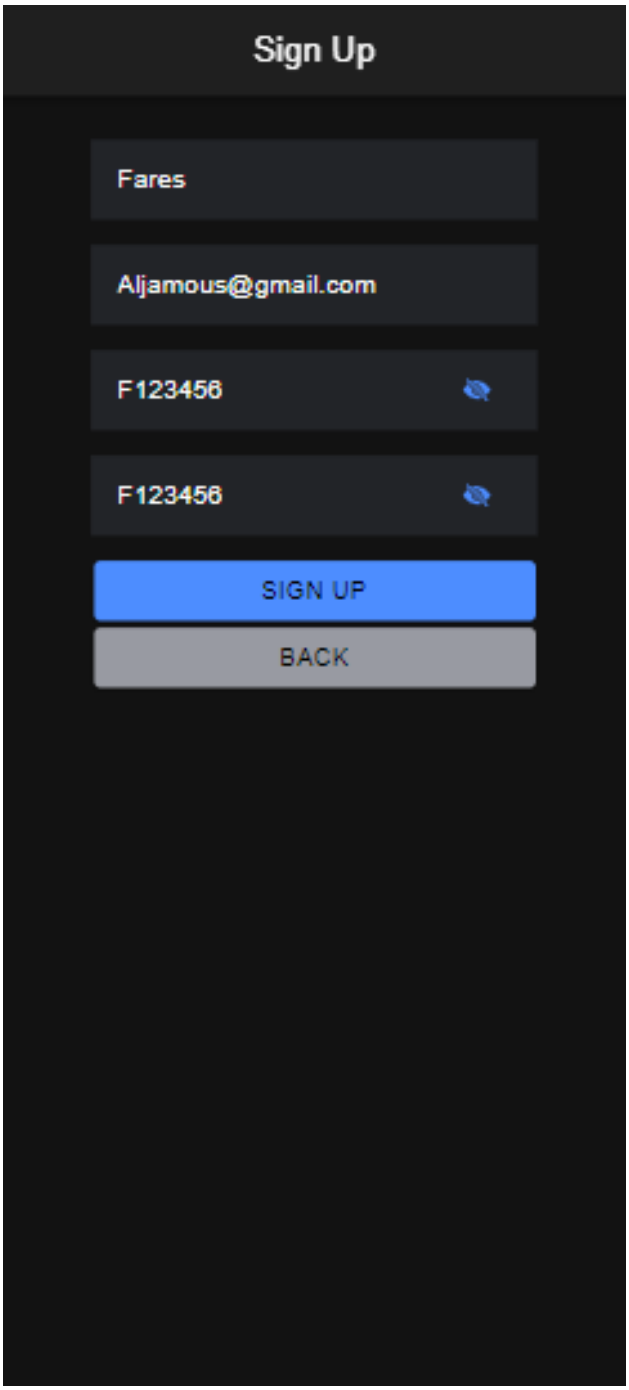

Figure 5: Take My Stuff Pages.

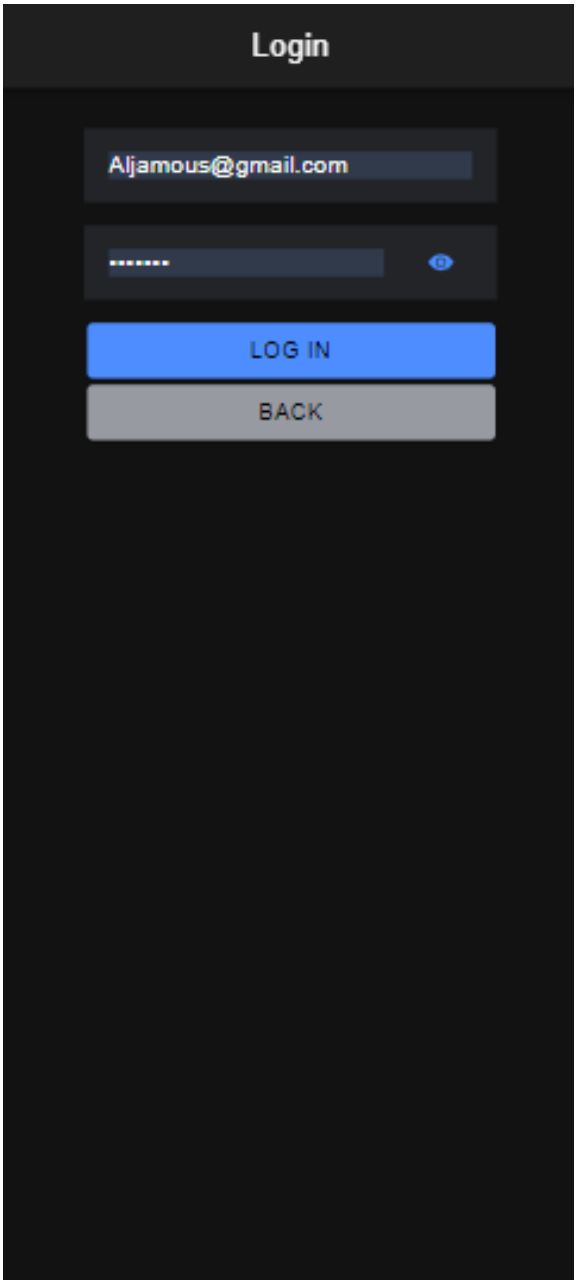

Figure 6: Take My Stuff Pages.

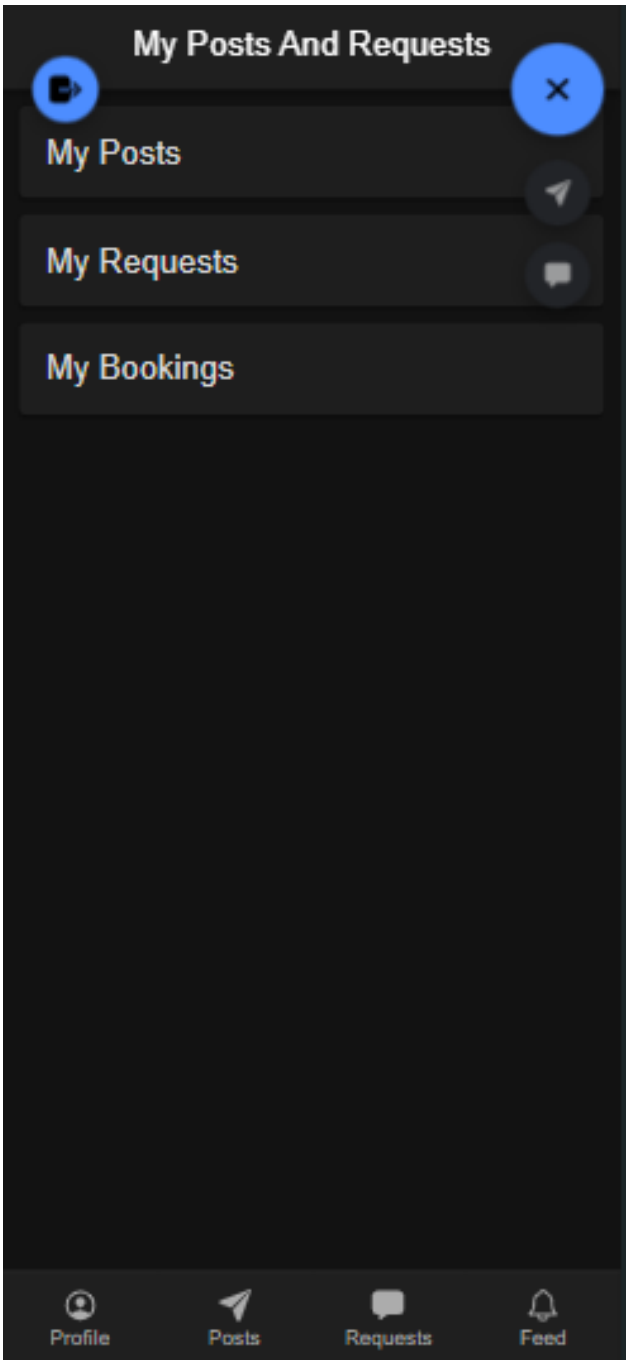

Figure 7: Take My Stuff Pages.

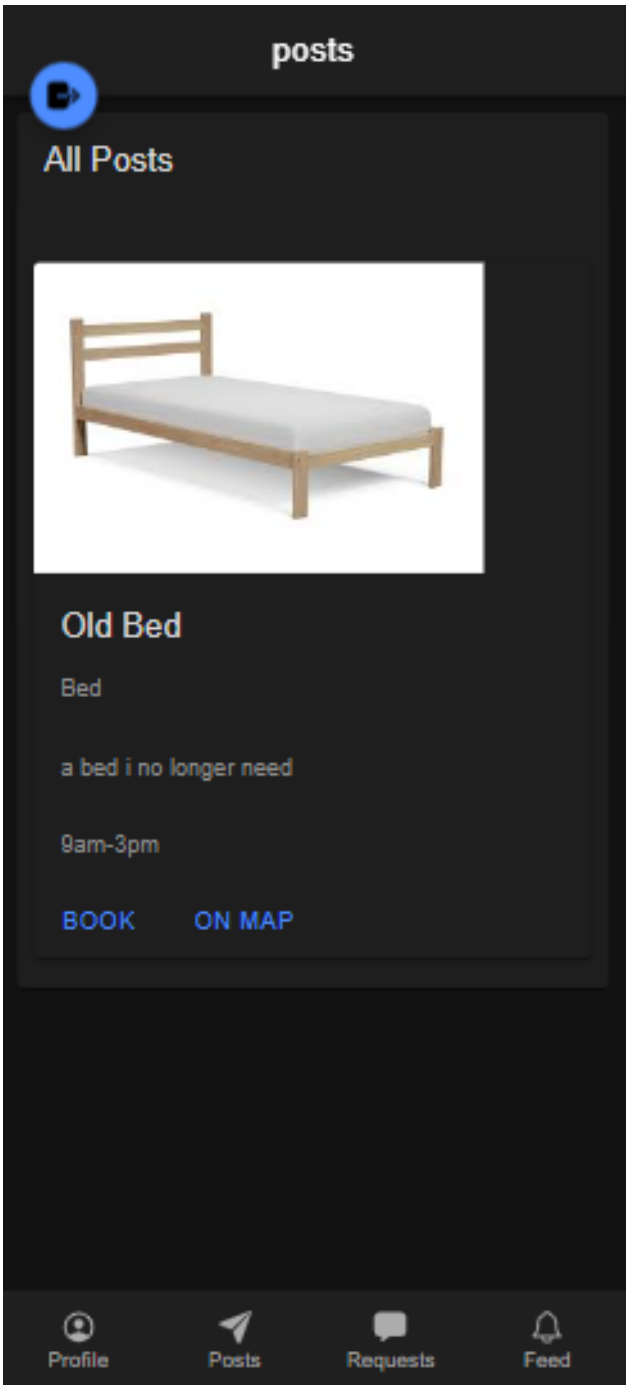

Figure 8: Take My Stuff Pages.

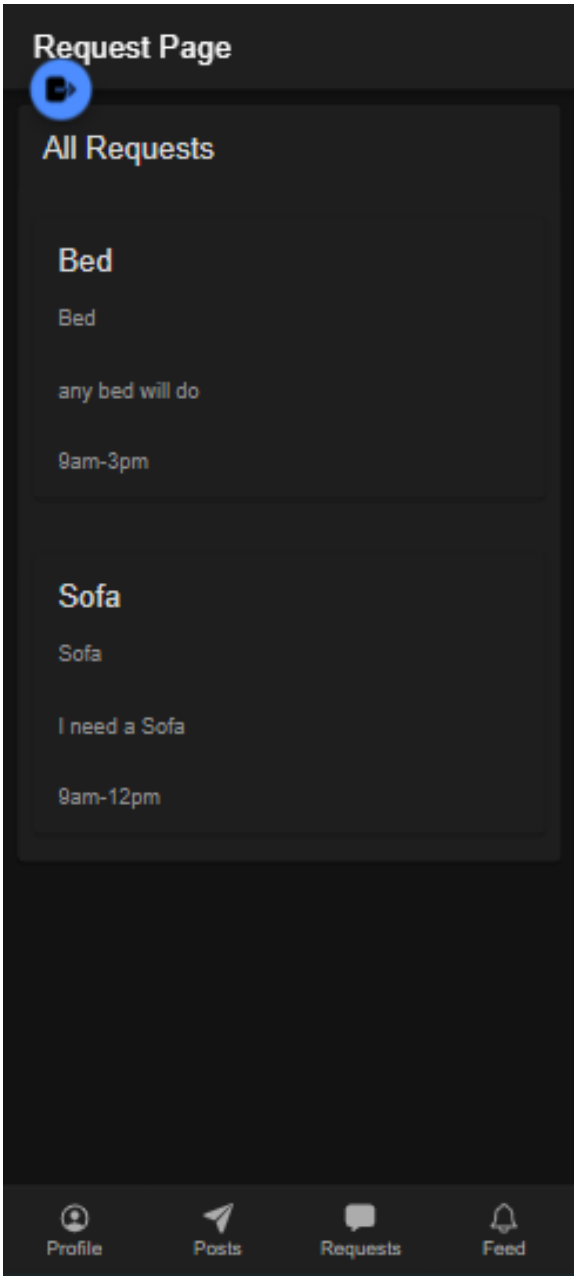

Figure 9: Take My Stuff Pages.

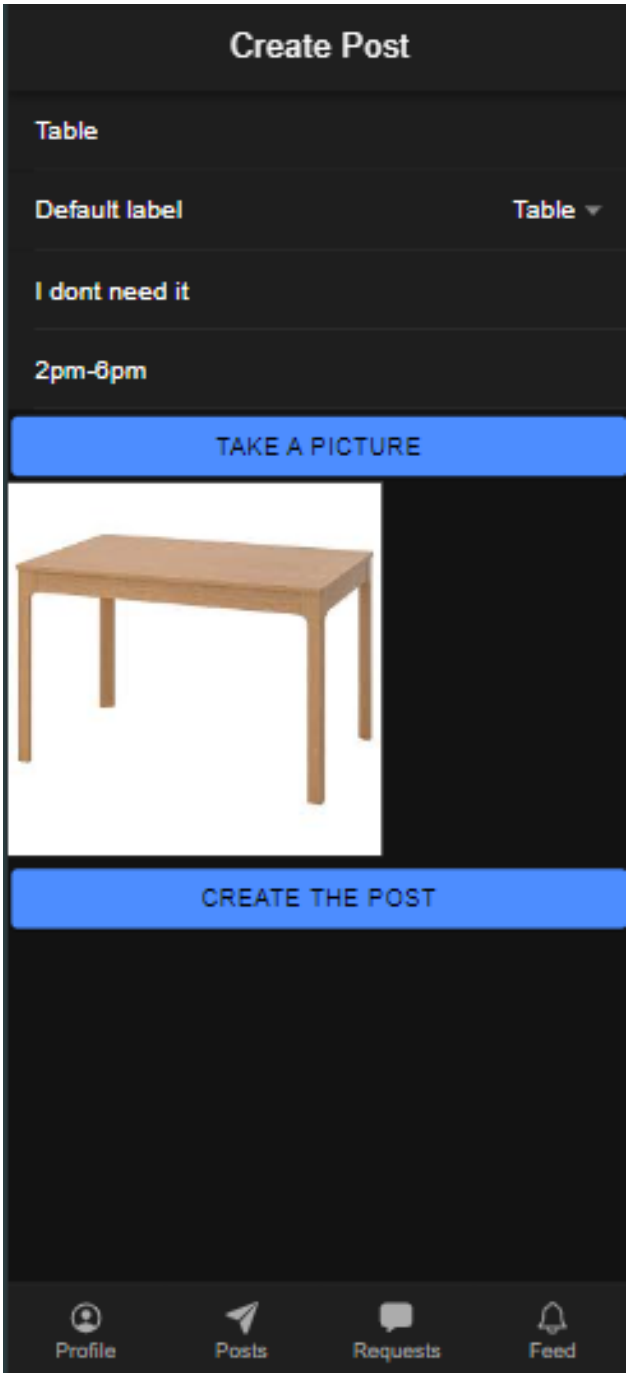

Figure 10: Take My Stuff Pages.

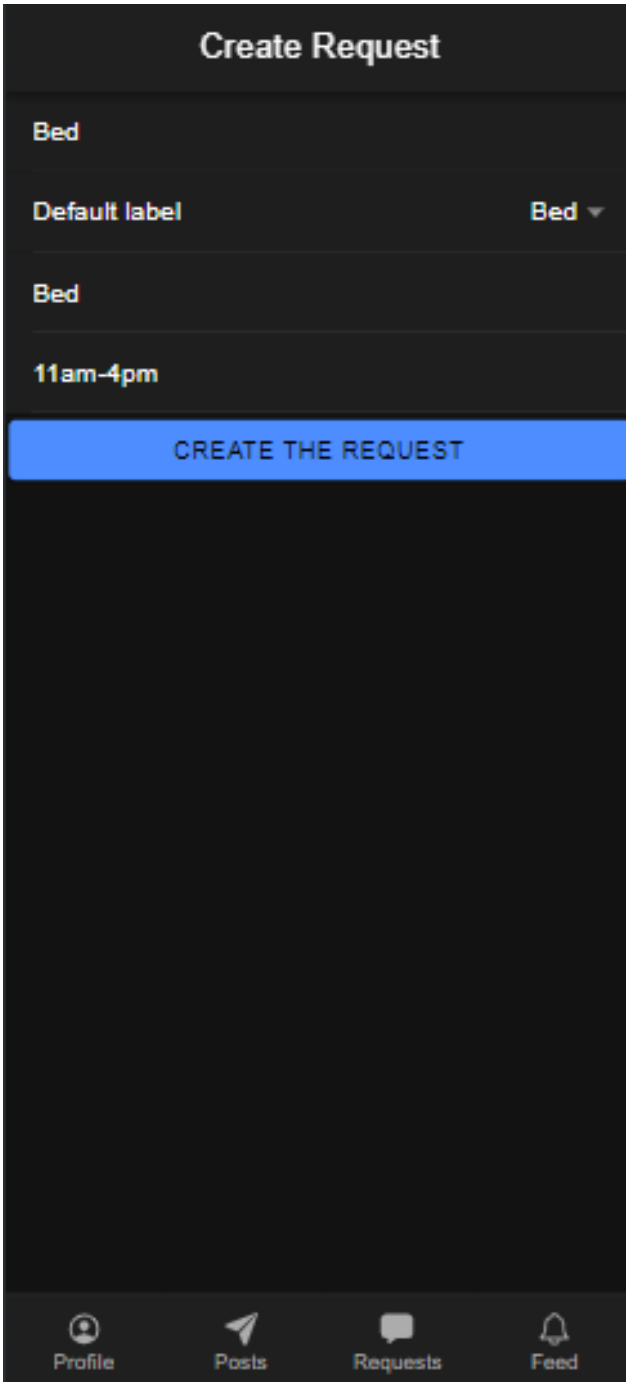

Figure 11: Take My Stuff Pages.

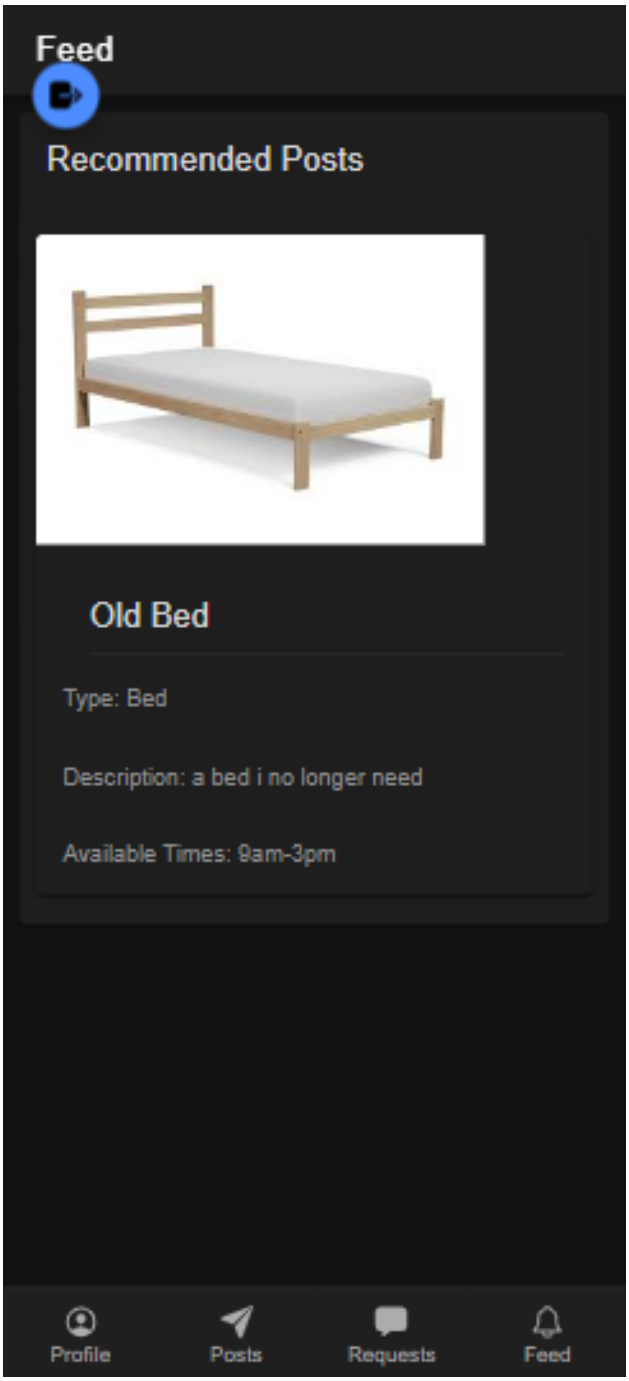

Figure 12: Take My Stuff Pages.

4 Conclusion

4.1 Summary

To sum up everything that was stated so far, our projects focused on helping people to recycle or reuse stuff that they no longer needed through a sophisticated application. Through many hours spend on writing the code, and making sure that any user could understand it easily making it user friendly was confirmed. This experiment's significance lies in its real-world application, highlighting the importance of utilizing technology to solve common environmental problems. The results demonstrate the successful implementation of a simple yet impactful solution, emphasizing the potential of such approaches to address broader environmental concerns and promote environment conservation. In conclusion, the project underscores the practical implications of the results and their relevance in a broader context of sustainable resource management.

4.2 Future Improvements and Takeaways

We could scale equipment to make it more efficient and to be usable in most countries.

4.3 Lessons Learned

We were able to gain more knowledge about developing a fully functional and user-friendly application, which might help us in our future careers to build applications or websites that will serve the purpose of the company that we are working for. We also learned how recycling and reusing could help the human race to survive for many years to come, during the research part that we did during the planning process.

4.4 Team Dynamics

We made sure that there was open communication and active participation from every team member to foster a collaborative and inclusive atmosphere. Our learning objectives were clearly defined, and we worked together to develop the assignments. Even though we didn't utilize a Gantt chart, we consistently monitored and updated chores in line with that. We convened

in the hall and the library to conduct brainstorming sessions and created a WhatsApp group for instant communication. We tried to include everyone in conversations and decision-making, even if not every member could attend every meeting. All things considered, our strategy encouraged cooperation, and we accomplished our goals.

4.5 Impact Statement

Environmental Impact Analysis								
Impact of your project	**Nature**	**Extent**	**Timing**	**Severity**	**Duration**	**Reversibility**	**Uncertainty**	**Significance**
The climate Example: *Does the project affect the emission of greenhouse gases into the atmosphere?*	Direct Positive	Local	Immediate	High	Temporary	Reversible	Low Likelihood	Unimportant
	Justification/Explanation: yes, because our project is to develop a mobile application that allows users to give away items they no longer need, promoting a community of recycling and reuse, thus decreasing the pollution.							
Use of Energy Example: *Does your project affect the energy consumption of the economy? How?*	Direct Positive	Local	Immediate	High	Temporary	Reversible	Low Likelihood	Unimportant
	Justification/Explanation: yes because less stores need to open ,thus eliminating the energy these stores might use.							
Air quality Example: *Does the project have an effect on emissions of harmful air pollutants that might affect human health, damage crops or buildings or lead to deterioration in the environment (soil or rivers)?*	Direct Positive	Local	Immediate	High	Temporary	Reversible	Low Likelihood	Unimportant
	Justification/Explanation: es, because our project is to develop a mobile application that allows users to give away items they no longer need, promoting a community of recycling and reuse, thus decreasing the pollution.							
Biodiversity, flora, fauna and landscapes Example: *Does it affect endangered species, their habitats or ecologically-sensitive areas?*	Direct Positive	Local	Immediate	High	Temporary	Reversible	Low Likelihood	Unimportant
	Justification/Explanation: yes because we will not have to cut down more trees to build more stores and make more items to use .							

Figure 13: .

Water quality and resources Example: *Does the project decrease or increase the quality or quantity of freshwater and groundwater?*	Direct Positive	Local	Immediate	High	Temporary	Reversible	Low Likelihood	Unimportant
	Justification/Explanation: yes ,by decreasing the pollution more clean water sources might be available to our use.							
Renewable or non-renewable resources Example: *Does the project reduce or increase use of non-renewable resources?*	Direct Positive	Local	Immediate	High	Temporary	Reversible	Low Likelihood	Unimportant
	Justification/Explanation: no , because our project is to develop a mobile application that allows users to give away items they no longer need, promoting a community of recycling and reuse.							
Sustainability Example: Does the option lead to more sustainable production and consumption? How?	Direct Positive	Local	Immediate	High	Temporary	Reversible	Low Likelihood	Unimportant
	Justification/Explanation: yes ,because our project promotes recycling and reusing .							
Waste production/generation/recycling Example: *Does the project affect waste production (solid, urban, agricultural, industrial, mining, radioactive or toxic waste) or how waste is treated, disposed of or recycled?*	Direct Positive	Local	Immediate	High	Temporary	Reversible	Low Likelihood	Unimportant
	Justification/Explanation: yes, because our project is to develop a mobile application that allows users to give away items they no longer need, promoting a community of recycling and reuse.and lowering waste by throwing less stuff to the trash.							

Figure 14: .

Economic Impact Analysis								
Impact of your project	**Nature**	**Extent**	**Timing**	**Severity**	**Duration**	**Reversibility**	**Uncertainty**	**Significance**
Economic Prosperity Example: *Does the project affect the GDP/capita, employment rate, household savings?*	Direct Positive	Local	Immediate	High	Temporary	Reversible	Low Likelihood	Unimportant
	Justification/Explanation: yes, because our project is to develop a mobile application that allows users to give away items they no longer need,many buisness owners will adapt this way of trading and more money will flow through the country.							
Investment Flows Example: *Does your project affect the flow of investment from outside the country? Does it encourage local investment in it?*	Direct Positive	Local	Immediate	High	Temporary	Reversible	Low Likelihood	Unimportant
	Justification/Explanation: yes, because our project is to develop a mobile application that allows users to give away items they no longer need,buisness owners might use this app instead of opning more stores which will cost a lot of money							
Public Budgets or Services Example: *Does the project affect the budgets of hospitals, community services, older people services, transport services, service quality, schools, policing, municipality services..etc?*	Direct Positive	Local	Immediate	High	Temporary	Reversible	Low Likelihood	Unimportant
	Justification/Explanation: yes, because our project is to develop a mobile application that allows users to give away items they no longer need, a lot of budget cuts will happen because they can get rid of many things that will be compensated by this app.ie the stores and salesmen.							
Market Mechanisms Example: *Does it affect the private sector business opportunities? Help companies reach more costumers? Change how business is done?*	Direct Positive	Local	Immediate	High	Temporary	Reversible	Low Likelihood	Unimportant
	Justification/Explanation: yes, because our project is to develop a mobile application that allows users to give away items they no longer need,everyone nowadays owns a phone and are lazy to go to a busy store to buy the things they want ,so they will start to use this method.							

Figure 15: .

Innovation, Research and Development Example: *Does the project have commercialization potential, lead to a potential patent? Does it allow others to innovate/research through it?*	Direct Positive	Local	Immediate	High	Temporary	Reversible	Low Likelihood	Unimportant
	Justification/Explanation: yes , it might help programmers to open buisnesses from the comfort of their homes.							
Sustainable Consumption and Production Example: Does the project produce a sustainably consumed product or service? Can it be produced sustainably?	Direct Positive	Local	Immediate	High	Temporary	Reversible	Low Likelihood	Unimportant
	Justification/Explanation: yes, because our project is to develop a mobile application that allows users to give away items they no longer need, promoting a community of recycling and reuse, thus decreasing the pollution.							

Figure 16: .

Social Impact Analysis								
Impact of your project	**Nature**	**Extent**	**Timing**	**Severity**	**Duration**	**Reversibility**	**Uncertainty**	**Significance**
Health and Longevity Example: *Does the project impact health and longevity? Does it affect physical activity, nutrition, chronic diseases, accidental injuries, independent living, mental wellbeing?*	Direct Positive	Local	Immediate	High	Temporary	Reversible	Low Likelihood	Unimportant
	Justification/Explanation: yes, because our project is to develop a mobile application that allows users to give away items they no longer need, promoting a community of recycling and reuse, thus decreasing the pollution, and a healthy living place will make you healthy							
Safety Example: *Does your project affect safety of social environment, protection of older people against abuse, protection against risks, response to emergency cases, feelings of safety, physical safety?*	Direct Positive	Local	Immediate	High	Temporary	Reversible	Low Likelihood	Unimportant
	Justification/Explanation: no, because our project is to develop a mobile application that allows users to give away items they no longer need, promoting a community of recycling and reuse, thus decreasing the pollution.							
Productive and Valued Activities Example: *Does the project increase leisure time, reduce stress, lead to positive behavior, increase productivity?*	Direct Positive	Local	Immediate	High	Temporary	Reversible	Low Likelihood	Unimportant
	Justification/Explanation: yes, because our project is to develop a mobile application that allows users to give away items they no longer need, so more people will start opening buisnesses and make it							

Figure 17: .

Standard of Living Example: *Does it affect the quality of life? Make lives easier? Reduce poverty and deprivation? Increase life choices and opportunities?*	Direct Positive | Local | Immediate | High | Temporary | Reversible | Low Likelihood | Unimportant **Justification/Explanation:** yes, because our project is to develop a mobile application that allows users to give away items they no longer need, promoting a community of recycling and reuse, thus decreasing the pollution.
Education/Life-long Learning Example: *Does the project affect literacy, use of ICT, chances of higher education, quality of education, life-long learning? Improve attainment of learning outcomes?*	Direct Positive | Local | Immediate | High | Temporary | Reversible | Low Likelihood | Unimportant **Justification/Explanation:** no, because our project is to develop a mobile application that allows users to give away items they no longer need, promoting a community of recycling and reuse, thus decreasing the pollution.
Quality of Social Interaction Example: *Does the project affect social connectedness, social participation, volunteering?*	Direct Positive | Local | Immediate | High | Temporary | Reversible | Low Likelihood | Unimportant **Justification/Explanation:** no, because our project is to develop a mobile application that allows users to give away items they no longer need, promoting a community of recycling and reuse, thus decreasing the pollution.

Figure 18: .

Privacy and Personal Data Example: *Does the project reveal the user identities? Create potential private data leaks or identity theft?*	Direct Positive | Local | Immediate | High | Temporary | Reversible | Low Likelihood | Unimportant **Justification/Explanation:** no, because our project is to develop a mobile application that allows users to give away items they no longer need, promoting a community of recycling and reuse, thus decreasing the pollution.
Social Reasonability Example: *Does the project affect access to products and services for people of determination? Does it affect their integration into society? Does it affect their participation in the economy? Does it address their needs?*	Direct Positive | Local | Immediate | High | Temporary | Reversible | Low Likelihood | Unimportant **Justification/Explanation:** no, because our project is to develop a mobile application that allows users to give away items they no longer need, promoting a community of recycling and reuse, thus decreasing the pollution.

Figure 19: .

References

[1] C. Khawas and P. Shah, "Application of firebase in android app development-a study," *International Journal of Computer Applications*, vol. 179, no. 46, pp. 49–53, 2018.

[2] S. Amann, S. Proksch, S. Nadi, and M. Mezini, "A study of visual studio usage in practice," in *2016 IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER)*, vol. 1. IEEE, 2016, pp. 124–134.