

## **Chapter – 1**

### **INTRODUCTION**

#### **1.1. Aim**

The aim of the Book Donation Management Android App is to create a user-friendly platform that facilitates the seamless donation and request of books, promoting knowledge sharing, resource utilization, and educational empowerment. The app aims to connect book donors with individuals in need, streamlining the process and ensuring the efficient distribution of donated books.

#### **1.2. Objectives of the Project**

- To create an application to manage Book Donation.
- To apply various Views, Layout and Components to design the app.
- To display the donated Books and request donated Books.
- To develop an Android Package Kit and install that in a Mobile.

#### **1.3. Book Donation Application**

The Book Donation Management Android App is a user-centric application designed to facilitate the process of book donation and request. This application provides a platform for users to donate their books, including textbooks, written or printed notes, and enables other users to request these donated books. By leveraging the power of Android Studio and the ease of use of an emulator, users can conveniently access and utilize the app's features.

The primary goal of the app is to connect book donors with individuals in need of books, creating a mutually beneficial environment that promotes knowledge sharing and resource utilization. The app's user module plays a crucial role in enabling users to participate in the book donation process seamlessly.

To begin using the Book Donation Android App, users are required to sign up by providing their full name, email address, University Serial Number (USN), and a chosen password. The sign-up process ensures that each user has a unique identity within the app's ecosystem. Once signed up, users can subsequently log in using their USN and password, granting them access to the app's functionalities.

The Book Donation Android App allows logged-in users to donate textbooks, notes, and educational resources by providing details such as title, author, edition, condition, and additional information. Users in need of books can submit requests for specific titles or subjects, and the app matches these requests with available donated books, facilitating the efficient distribution of resources. The user-friendly interface simplifies the process of searching for books, viewing details, and making requests, ensuring that the right resources reach the intended recipients.

---

**Chapter – 2****LITERATURE SURVEY**

The literature survey conducted for the Book Donation Management Android App project involved exploring existing online resources related to various aspects of the development process. The survey aimed to gather insights, best practices, and knowledge from the field of Android app development, book donation management systems, and relevant technologies.

One important aspect of the literature survey focused on understanding the structure and components of an Android Studio project. This involved studying the documentation and resources provided by the official Android Developer website[5], which covered topics such as project organization, file structure, build configurations, and the use of the Project window in Android Studio.

To gain a deeper understanding of Android application development, the survey delved into the Android platform itself. This included studying the Android runtime environment and the Application Programming Interface (API) libraries available for building Android apps. By exploring the official Android documentation and other reputable sources, the survey aimed to identify best practices and recommended approaches for designing and developing the Book Donation Management app.

In order to implement the book donation management system and its connectivity with a database, the survey involved researching different technologies. PHP was identified as a suitable language for database connectivity, and the literature survey focused on understanding its integration with Android apps. Online resources, tutorials, and forums were explored to gather information about PHP integration, database management, and the best practices for handling data in Android apps.

Furthermore, the literature survey included examining various online sources, websites, and YouTube channels to gather code snippets, examples, and insights from the Android development community. These resources provided valuable information on implementing specific features, user interface design, and optimizing performance in Android apps.

Overall, the literature survey served as a foundation for understanding the existing knowledge, practices, and technologies relevant to the development of the Book Donation Management Android App. It helped identify industry-standard approaches, best practices, and valuable resources to ensure a well-informed and effective development process.

## Chapter – 3

### REQUIREMENTS

#### 3.1. System Requirement

System requirements contains the requirements for installing of Android Studio.

For Android Studio	
<b>OS</b>	64-bit Windows 10
<b>Memory</b>	8 GB+ RAM
<b>Storage</b>	8 GB of available disk space minimum (IDE + Android SDK + Android Emulator)
<b>Processor</b>	x86_64 CPU architecture; 2nd generation Intel Core or newer, or AMD CPU with support for a Windows Hypervisor
<b>Other Requirements</b>	x86_64 CPU architecture; 2nd generation Intel Core or newer, or AMD CPU with support for a Windows Hypervisor
<b>Virtualization</b>	Virtualization should be enabled in BIOS settings.

#### 3.2. Software Requirements

These are requirements for developing the app in Android studio.

<b>Programming Language</b>	Java, PHP-8
<b>SDK Version</b>	Android 12 (API Level 31)
<b>Database</b>	MySQL-8
<b>Web Server</b>	XAMPP

#### 3.3. Mobile Requirements

These requirements are for installing and running the app in your mobile device.

<b>Mobile OS</b>	Android 4.4+
<b>Permission Required</b>	Internet

---

## Chapter – 4

### DESIGN AND IMPLEMENTATION

#### 4.1. Manifest File Code

The manifest file in the Book Donation Management Android App contains essential information about the app, such as permissions, activities, services, and receivers, ensuring proper functioning and integration with the Android operating system.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">
    <uses-permission android:name="android.permission.INTERNET" />
    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/Theme.BookApp"
        android:usesCleartextTraffic="true"
        tools:targetApi="31">
        <activity
            android:name=".Login"
            android:exported="true" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category
                    android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity
            android:name=".SignUp"
            android:exported="false" />
        <activity
            android:name=".MainActivity"
            android:exported="false" />
    </application>
</manifest>
```

## 4.2. Donation.java Code

The Donation.java code manages the donation page activities by accepting user inputs from the app and transmitting them to a PHP file for further processing and handling.

```
package com.app.bookapp;
import android.content.Intent;
import android.os.Bundle;
import androidx.fragment.app.Fragment;
import android.os.Handler;
import android.os.Looper;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
import com.google.android.material.textfield.TextInputEditText;
import com.vishnusivadas.advanced_httpurlconnection.PutData;
import org.w3c.dom.Text;
public class Donatation extends Fragment {

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup
container,
                                Bundle savedInstanceState) {
        View view = inflater.inflate(R.layout.fragment_donatation,
container, false);

        TextInputEditText textInputEditTextUsername =
view.findViewById(R.id.usr);
        TextInputEditText textInputEditTextBookname =
view.findViewById(R.id.bookname);
        TextInputEditText textInputEditTextsubject =
view.findViewById(R.id.subject);
        TextInputEditText textInputEditTextsemester =
view.findViewById(R.id.semester);
        TextInputEditText textInputEditTextbooktype =
view.findViewById(R.id.type);
        TextInputEditText textInputEditTextbookMobile =
view.findViewById(R.id.mobile);
        Button buttonSubmit = view.findViewById(R.id.submit);

        //progressB=findViewById(R.id.progress1);

        buttonSubmit.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String bookname, subject, semester,
booktype,mobile,username;

                username=String.valueOf(textInputEditTextUsername.getText());
                bookname =
String.valueOf(textInputEditTextBookname.getText());
                subject =
String.valueOf(textInputEditTextsubject.getText());
                semester =
String.valueOf(textInputEditTextsemester.getText());
                booktype =
String.valueOf(textInputEditTextbooktype.getText());
```

---

```

        mobile =
String.valueOf(textInputEditTextbookMobile.getText());
        if (!username.equals("") && !bookname.equals("") &&
!subject.equals("") && !semester.equals("") && !booktype.equals("") &&
!mobile.equals("")) {
            //progressB.setVisibility(View.VISIBLE);
            Handler handler = new
Handler(Looper.getMainLooper());
            handler.post(new Runnable() {
                @Override
                public void run() {
                    String[] field = new String[6];
                    field[0] = "username";
                    field[1] = "bookname";
                    field[2] = "subject";
                    field[3] = "semester";
                    field[4] = "booktype";
                    field[5] = "mobile";
                    //Creating array for data
                    String[] data = new String[6];
                    data[0] = username;
                    data[1] = bookname;
                    data[2] = subject;
                    data[3] = semester;
                    data[4] = booktype;
                    data[5] = mobile;
                    PutData putData = new
PutData("http://117.247.182.134:8081/ic/mad/Bookapp/donate.php", "POST",
field, data);

                    if (putData.startPut()) {
                        if (putData.onComplete()) {

                            String result = putData.getResult();
                            if (result.equals("Submit
Success")) {
                                Toast.makeText(getActivity().getApplicationContext(), result, Toast.LENGTH
_SHORT).show();

                                    Intent intent=new
Intent(getActivity().getApplicationContext(), Donatation.class);
                                    startActivity(intent);
                                    getActivity().finish();

                                }
                                else {
                                    Toast.makeText(getActivity().getApplicationContext(), result, Toast.LENGTH
_SHORT).show();

                                }

                            }

                        }

                    });
                }
            }
            else {

                Toast.makeText(getActivity().getApplicationContext(), "All fields are
                required", Toast.LENGTH_SHORT).show();

            }

        }

    });
    return view;
}
}

```

### 4.3. Donation XML Code

This code contains the design of the Donation page.

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:fillViewport="true">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        tools:context=".Donation">

        <TextView
            android:id="@+id/textView"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="@string/donate"/>
        <com.google.android.material.textfield.TextInputLayout
            android:id="@+id/textInputLayoutUsername"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="@string/usn"
            >
            <com.google.android.material.textfield.TextInputEditText
                android:id="@+id/usr"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:required="true"
            />
        </com.google.android.material.textfield.TextInputLayout>

        <com.google.android.material.textfield.TextInputLayout
            android:id="@+id/textInputLayoutBookname"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="@string/book_name"
            >
            <com.google.android.material.textfield.TextInputEditText
                android:id="@+id/bookname"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:maxLength="50"
                android:required="true"
            />
        </com.google.android.material.textfield.TextInputLayout>

        <com.google.android.material.textfield.TextInputLayout
            android:id="@+id/textInputLayoutSubject"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="@string/subject"
            >
            <com.google.android.material.textfield.TextInputEditText
                android:id="@+id/subject"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:required="true"
            />
        </com.google.android.material.textfield.TextInputLayout>
    </LinearLayout>
</ScrollView>
```



```

        <com.google.android.material.textfield.TextInputLayout
            android:id="@+id/textInputLayoutSem"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="@string/semester">

            <com.google.android.material.textfield.TextInputEditText
                android:id="@+id/semester"
                android:layout_marginTop="5dp"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:fontFamily="@font/poppinsmedium"
                android:maxLength="5"
                android:required="true"
            />

        </com.google.android.material.textfield.TextInputLayout>

        <com.google.android.material.textfield.TextInputLayout
            android:id="@+id/textInputLayoutType"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="@string/book_type"
            tools:layout_editor_absoluteX="0dp">

            <com.google.android.material.textfield.TextInputEditText
                android:id="@+id/type"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:fontFamily="@font/poppinsmedium"
                android:maxLength="25"
                android:required="true"
            />

        </com.google.android.material.textfield.TextInputLayout>
    <com.google.android.material.textfield.TextInputLayout
        android:id="@+id/textInputLayoutMobile"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Mobile number"
        tools:layout_editor_absoluteX="0dp">

        <com.google.android.material.textfield.TextInputEditText
            android:id="@+id/mobile"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:fontFamily="@font/poppinsmedium"
            android:inputType="number"/>

    </com.google.android.material.textfield.TextInputLayout>
    <ProgressBar
        android:id="@+id/progress1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:visibility="gone"/>

    <Button
        android:id="@+id/submit"
        style="@style/Widget.AppCompat.Button.Colored"
        android:layout_width="200dp"
        android:layout_height="wrap_content"
        android:text="@string/submit" />
</LinearLayout>
</ScrollView>

```

---

## Chapter – 5

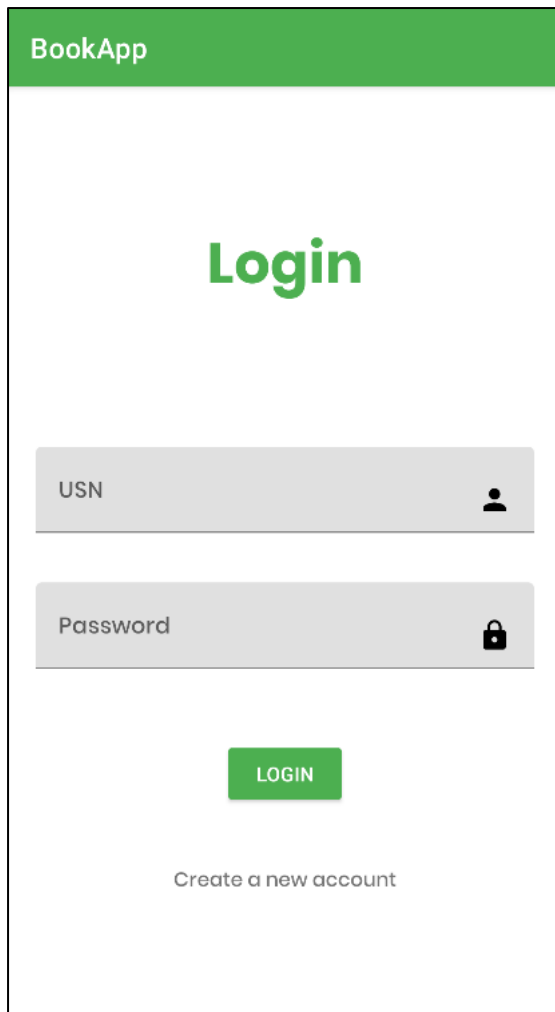
### RESULT

#### 5.1. ACTIVITIES

The design elements and layout of the activities aim to provide a seamless and intuitive experience for users, ensuring a smooth access of the app's ecosystem.

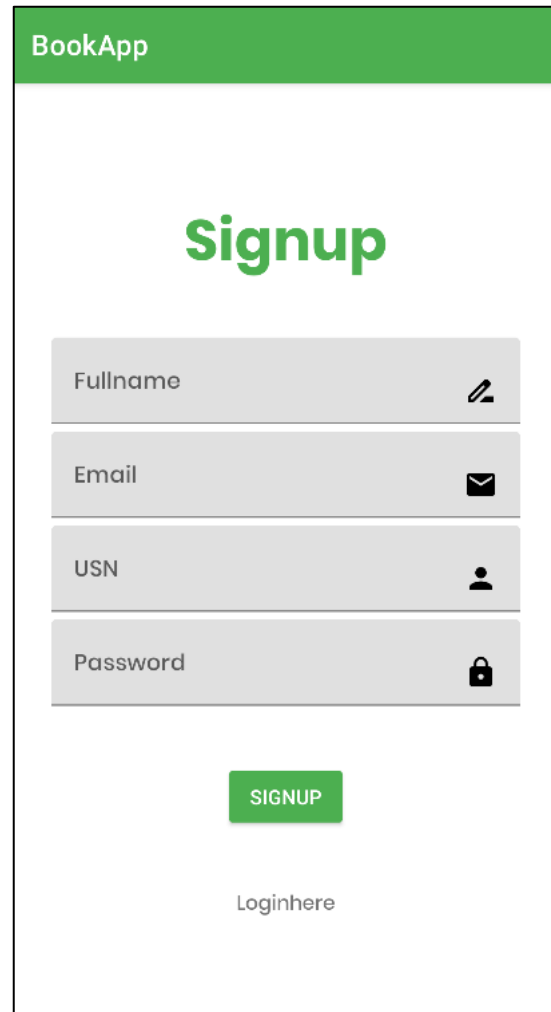
Fig 5.1 shows the login page for the Book Donation Management Android App showcases a user-friendly interface with a login form where users can enter their University Serial Number (USN) and password for secure access to the app's functionalities.

Fig 5.2 shows the sign-up page snapshot of the Book Donation Management Android App displays an intuitive interface where users can input their full name, email address, University Serial Number (USN), and create a password to create an account.



The screenshot shows the 'Login' page of the 'BookApp'. At the top is a green header with the text 'BookApp'. Below the header, the word 'Login' is displayed in a large, bold, green font. There are two input fields: the first is labeled 'USN' with a person icon on the right, and the second is labeled 'Password' with a lock icon on the right. Below these fields is a green button labeled 'LOGIN'. At the bottom, there is a link that says 'Create a new account'.

Fig 5.1: Login Page – Pops up when user opens the App



The screenshot shows the 'Signup' page of the 'BookApp'. At the top is a green header with the text 'BookApp'. Below the header, the word 'Signup' is displayed in a large, bold, green font. There are four input fields: 'Fullname' with a pencil icon, 'Email' with an envelope icon, 'USN' with a person icon, and 'Password' with a lock icon. Below these fields is a green button labeled 'SIGNUP'. At the bottom, there is a link that says 'Loginhere'.

Fig 5.2: Sign Up Page – The user can register to App here

Fig 5.3 shows the home page snapshot of the Book Donation Management Android App features a visually appealing layout, providing easy access to the app's main functionalities, including book donation and book request options. The snapshot showcases a user-centric interface with intuitive navigation, promoting a seamless experience for users.

Fig 5.4 shows the donation page snapshot of the Book Donation Management Android App presents a user-friendly interface where users can input details of the books they wish to donate, such as title, author, edition, and condition, providing a seamless process for contributing to the app's book inventory.

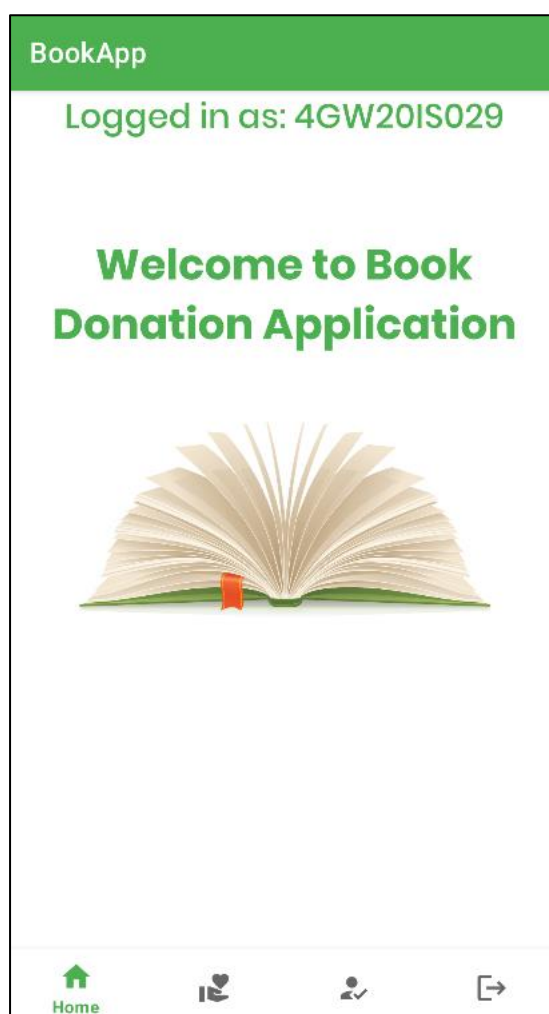


Fig 5.3: Home page - Appears when user logs in.

The screenshot shows the donation page of the BookApp. At the top, a green header bar contains the text "BookApp". Below the header, the word "Donate" is displayed in green. The page contains a form with six input fields: "USN", "Book Name", "Subject", "Semester", "Book Type", and "Mobile number". Below the form is a green "SUBMIT" button. At the bottom, there is a navigation bar with four icons: a house icon, a heart icon labeled "Donate", a person icon, and a right-pointing arrow icon.

Fig 5.4: Donation page – where users can donate their books.

Fig 5.6 shows the request page snapshot of the Book Donation Management Android App showcases a user-centric interface where users can submit requests for specific books they need, allowing for a streamlined process of finding and acquiring the required resources.

Fig 5.7 shows the logout page snapshot of the Book Donation Management Android App displays a logout button to ensure a secure logout process, providing users with a seamless way to log out from their accounts and safeguard their privacy

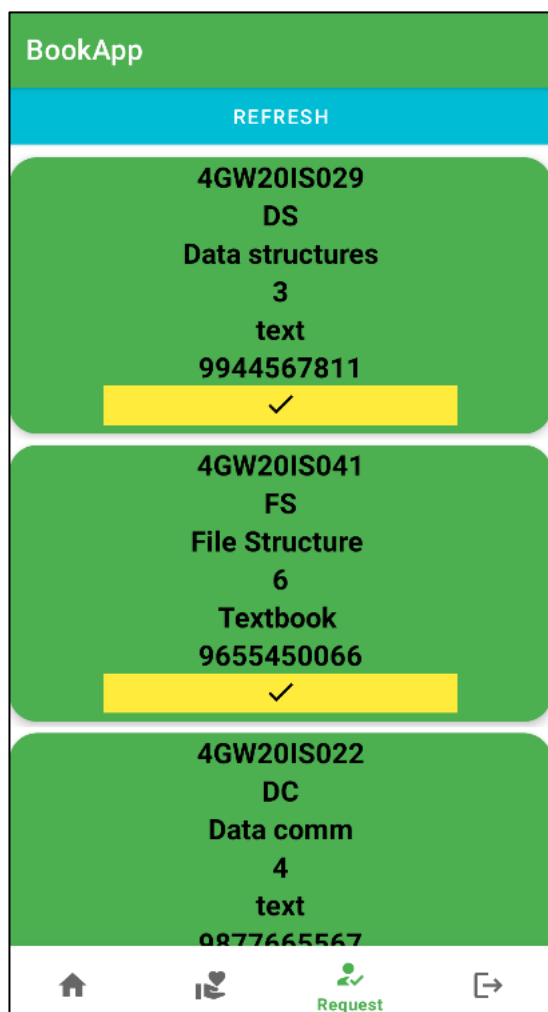


Fig 5.6: Request Page – Users can request books from here

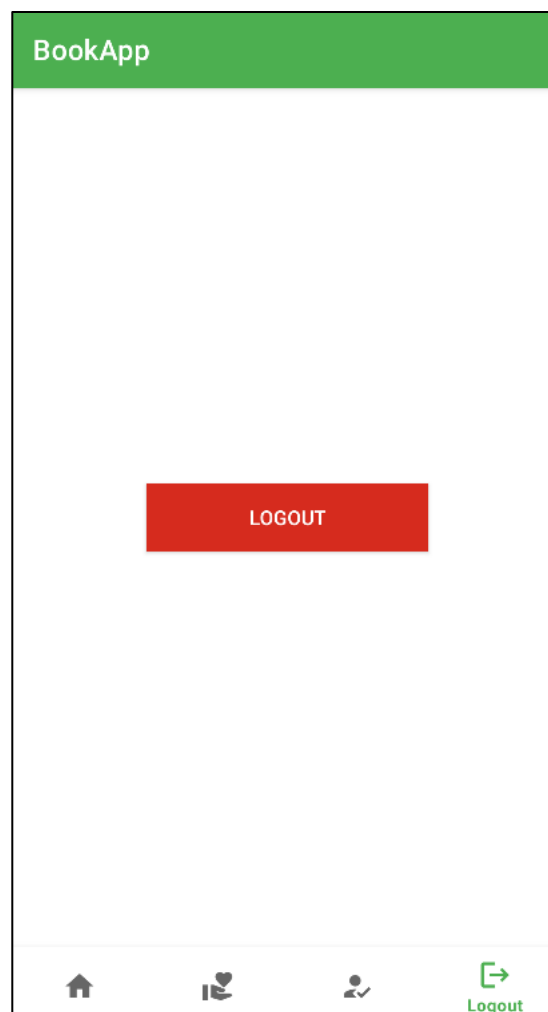


Fig 5.6: Logout Page – Users can logout from here

## **Chapter – 6**

### **CONCLUSION AND LEARNING OUTCOME**

#### **6.1. Conclusion**

The book donation application has proven to be a valuable tool in promoting literacy and providing access to books for individuals in need. Through its user-friendly interface and wide-reaching network, the application has successfully facilitated the donation process, connecting book donors with recipients in an efficient and effective manner.

Furthermore, the book donation application has contributed to sustainability efforts by promoting the reuse and recycling of books. By facilitating the redistribution of pre-owned books, it has reduced waste and promoted a greener approach to education.

#### **6.2. Learning Outcomes**

- Understood the history of Android development and what is required to build Android apps.
- Gained a basic understanding of Android application development.
- Designed and developed user Interfaces for the Android platform.
- Connection of Android application to XAMPP.
- Retrieving data from JSON file.

## Appendix – 1

### APPLICATION INSTRUCTIONS

- Book Donation Application can be downloaded from GitHub.

Link: <https://github.com/meghakprasad?tab=repositories>

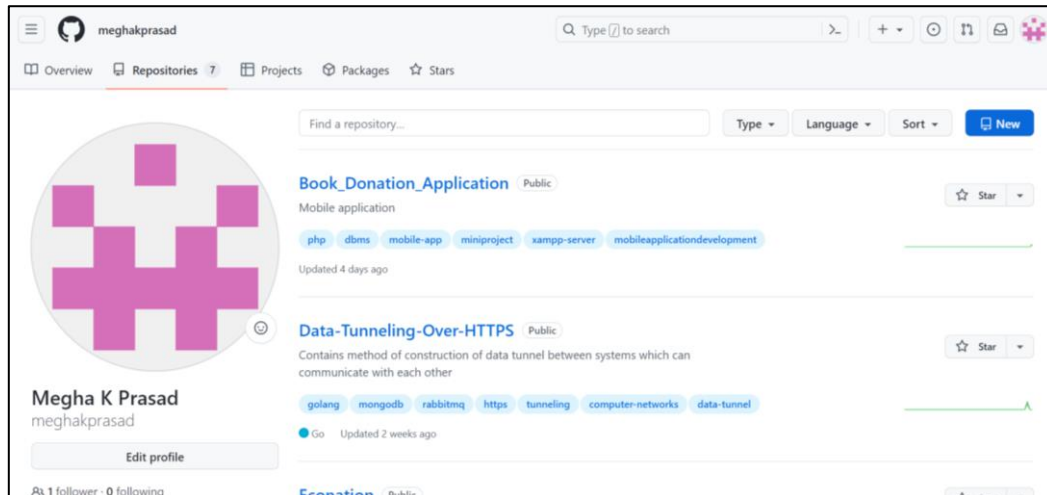


Fig: GitHub Repository to download the application

- Open the Book Donation Application.
- Create an account by Signing up.
- Provide the credentials to login (use the same username and password that was given during Signup). After successful login you will be directed to Home page.
- Navigate to donation page from the bottom navigation. In the donation page give the details of the book you want to donate and submit.
- To request for a donated book, go to request page and select a book from the list provided by clicking ( ✓ ) button. After which you can contact the book owner.

## REFERENCES

- [1] Android Studio download (Online) URL: <https://developer.android.com/studio>  
Accessed on: 12/06/2023
- [2] Login And Sign Up in Android with MySQL. (YouTube Video)  
URL: <https://youtu.be/X8oD4q3XtQQ> . Accessed on: 24/06/2023
- [3] For Recycler View. (YouTube Video) URL: [https://youtu.be/\\_znvxVptpHY](https://youtu.be/_znvxVptpHY) . Accessed  
on: 18/06/2023
- [4] Stack Overflow (Online) URL: <https://stackoverflow.com/> . Accessed on: 20/06/2023
- [5] Android Developers (Online) URL: <https://developer.android.com/docs> . Accessed on:  
12/06/2023
- [6] Intents and intent filters (Online) URL: [https://developer.android.com/guide  
/components/intents-filters](https://developer.android.com/guide/components/intents-filters) Accessed on: 12/06/2023