

BEAUTY_BUDD MOBILE APP

By

Ayesha Ijaz

2021-GCUF-073655

ASSOCIATE DEGREE

IN

COMPUTER SCIENCE



DEPARTMENT OF COMPUTER SCIENCE

Government College University Faisalabad

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

"In the name of Allah, most Gracious, most Compassionate"

"شروع اللہ کے پاک نام سے جو بڑا مہربان نہایت رحم والا ہے"

DECLARATION

The work reported in this project is carried by me under the supervision of **MR. UMAR** Principal, **TIPS COLLEGE**, Faisalabad, Pakistan.

I hereby declare that my project, “**BEAUTYBUDD MOBILE APP**” and all its contents that include Software are the product of my research. I further declare that this work has not been submitted for the award of any other degree diploma. The University has all rights to take into action if the information provided is found inaccurate at any stage.

Signature of the Student:

Registration No. 2021-GCUF-073655

CERTIFICATE BY THE PROJECT SUPERVISOR

It is to certify that **[name]** under Roll no. **100384** and Registration No. **2020-GCUF-** has completed his Final Year Project named as: “**BEAUTY BUDD APP**” for Government College University, Faisalabad, at the Department of Information Technology to fulfill the partial requirement of the degree of Bachelor of Science in computer science.

Member of Supervisory Committee

Member of Supervisory Committee

Chairperson

Name:_____ Signature:_____ Designation
with Stamp: _____

Name:_____ Signature:_____ Designation
with Stamp: _____

ACKNOWLEDGEMENT

First and foremost, all prayers go to **ALLAH**, I find no words to define my heartiest sense of gratitude to the **Almighty ALLAH**, the Most Generous Compassionate, and Beneficent, and to **Prophet MUHAMMAD (SAW)**, Who is the best light of guidance for whole humanity from dawn to dusk. I am so much thankful to **ALLAH** by whom Grace and Blessings I completed my project on time.

I am also grateful to my loving parents for their encouragement and all sort of moral, social, and financial support not only in my academic career but also in all other aspects of life. Their prayers were incessant and enabled me to reach this stage. I am also thankful to all my family and my grandparents for their kindness, love and ever supporting prayers.

Words are very important to convey thoughts and thanks; the words are impossible to find to thanks my father and whole family for their prayers and

encouragement for us and for our work. Finally, I apologize If I have caused anger of offence to anybody and the errors that remain in the project.

AYESHA IJAZ

2021-GCUF-073655

Table Of Contents

ABSTRACT	9
CHAPTER -1	10
INTRODUCTION	10
Background:	10
Scope:	11
Reasons to Develop this Product:	11
User of this product:	13
CHAPTER -2	14
2.1 Functional Requirements:	14
2.1.1 List of Functional Requirements:	14
2.1.2 Description of Each Requirement in Requirement Shell:	14
Requirement Name: User Registration/Sign Up :	14
Requirement Name: User Login	15
Requirement Name: Main Screen	16
Requirement Name: Salon List	16
Requirement name: Reservation	17
2.2 Non-Functional Requirements:	18
2.2.1 Accessibility:	18
2.2.2 Security:	18
2.3 System Requirements:	18
Performance Requirement:	19
Software Requirement:	19
1. Development Environment:	19
2. React Native Framework:	19
3. Expo:	20
4. React Navigation:	20
CHAPTER-3	21
DESIGN - APP SCREENSHOTS	21
USER SIGN UP	21
USER SIGNIN	23
MAIN SCREEN	24
LOCATION PERMISSION	26
SALOON LIST	26
SEE LOCATION OF SALON	27
RESERVATION FORM	29
CHAPTER - 4 DIAGRAMS	30
FLOW CHART DIAGRAM	30
WORKING OF APP	31
CHAPTER - 5 CODING	32
SIGN UP	32
SIGN IN	39
MAIN PAGE	45
SALON	46

SALON LIST	50
LOCATION	51
RESERVATION	54
PARENT WRAPPER	58
DEPENDENCIES INSTALLATION DETAILS	60
Test Case	63
Test Cases	63
<i>Table : Login</i>	65
Programming Languages	66
Operating System & Tools	66
Appendix A: Glossary	67

ABSTRACT

Abstract: In current time, finding contact numbers and address of beauty salon is difficult for customers to fix an appointment. Now it's a digital era, customer can save their time by using this application as they use it from home or other places. As this provide a single environment through which a customer can find many various salons nearby their locality and book any of them. User can have their own account. User can enter their location and nearby salon to their locality appears. Then they click one of them that will show the salon's beauty services with their prices. User can fill a booking form in order to book that salon. And when users get their beauty treatment done. They can give their feedback according to their experience regarding the salon.

CHAPTER -1

INTRODUCTION

This chapter provides background information about salon hunting and outlines its scope. It also postulates the main aim and research objectives, the limitations, thesis outline and key definitions of salon hunting.

Background:

In old times, finding contact numbers and address of beauty salons were difficult for customers. Also, people can't find good salons near their locality. Precious time of people were wasted in searching of salon in their locality. This searching affected the cost of the customers. And it was very time taking as well as money consuming to fix an appointment. In order to avail the beauty services of the salon.

- I designed a Salon Hunting Application as a single platform where user can find various salons in their locality and can book it.
- In order to fix an appointment to user's nearby locality.
- So, it will save time as it is convenient to use and save money as it provides affordable prices.
- User can have their own account. They turn on their mobile location and nearby salons to their locality will appear.

- Then they can click one of them that will show the salon's beauty services with their prices. User can fill a booking form in order to book that salon through sending SMS or by calling them.

Scope:

- It is beneficial for both the customers and the salon owners.

Stakeholders:

- Customers (Salon)
- Salons

Reasons to Develop this Product:

In old times, finding contact numbers and address of beauty salons were difficult for customers. Also, people can't find good salons near their locality. Precious time of people were wasted in searching of salon in their locality. This searching affected the cost of the customers. And it was very time taking as well as money consuming to fix an appointment. In order to avail the beauty services of the salon.

Certainly, here are the reasons to develop this product in a more detailed bullet-point format:

- **Addressing Market Demand:**
 - Responding to the growing demand for online salon booking and appointment management services.
 - Meeting the evolving preferences of modern consumers who seek convenience in booking beauty services.

- **Improving Salon Operations:**

- Streamlining salon operations by providing a centralized platform for appointment management.
- Optimizing resource allocation, reducing scheduling conflicts, and enhancing customer experience.

- **Enhancing User Experience:**

- Offering users an intuitive and user-friendly app for discovering, comparing, and booking salon services.
- Providing transparency through detailed service information and user reviews.

- **Promoting Local Businesses:**

- Supporting local salons by offering them an online presence and access to a broader customer base.
- Empowering small and independent salons to compete with larger chains.

- **Generating Revenue:**

- Creating revenue streams through booking fees, premium listings, and promotions.
- Offering value-added services to salons for better visibility and customer engagement.

- **Competitive Advantage:**

- Gaining a competitive edge in the beauty and salon industry by offering a feature-rich, tech-driven platform.
- Meeting the changing expectations of customers who prefer digital solutions.

- **Meeting Modern Expectations:**

- Aligning with contemporary consumer preferences for mobile app-based services.
- Providing a platform that caters to users' desire for on-demand and hassle-free booking.

- **Data Insights:**

- Gathering valuable data on user preferences, booking trends, and salon performance.
- Leveraging data analytics for informed decision-making and continuous product improvement.

Each of these reasons contributes to the rationale behind developing this salon app, catering to both user needs and the business potential in the beauty and salon industry.

User of this product:

- Customers (Salon)

- Salon Owner

CHAPTER -2

2.1 Functional Requirements:

2.1.1 List of Functional Requirements:

- User Registration/Sign Up.
- User Login.
- Main Screen.
- Salon List.
- Reservation.

2.1.2 Description of Each Requirement in Requirement Shell:

Requirement Name: User Registration/Sign Up :

User registration is a fundamental process in the Salon App, allowing individuals to create their accounts and access the platform's features. This essential step ensures that users can take full advantage of the app's capabilities, including booking salon appointments, receiving notifications, and customizing their experience. The registration process is designed to be straightforward, user-friendly, and secure, requiring users to provide certain information to establish their identities and preferences within the app.

Requirement #:	Req01	Requirement Type:	Functional
Description:	User Registration		
Rationale:	This Requirement is important because the user has to register his or her self for the application.		
Originator:	User		
Fit Criterion:	User has to register his or her self for the application.		
Customer Satisfaction:	5	Customer Dissatisfaction:	10
Priority:	3	Conflicts:	Nil
Supporting Materials:	N/A		
History:	N/A		

Table 3.1 User Registration/Sign Up

Requirement Name: User Login

User login is a crucial component of the Salon App, enabling registered users to access their accounts and utilize the app's full functionality. It provides a secure and convenient way for users to authenticate themselves and access their profiles, booked appointments, and other personalized features.

Requirement #:	Req02	Requirement Type:	Functional
Description:	User Login		
Rationale:	This Requirement is important because the user has to login the application.		
Originator:	User		
Fit Criterion:	User must have to login on application after registration process.		
Customer Satisfaction:	5	Customer Dissatisfaction:	10
Priority:	3	Conflicts:	Nil
Supporting Materials:	N/A		
History:	N/A		

Table 3.2 User Login

Requirement Name: Main Screen

The main screen of the Salon App serves as the user's entry point and provides quick access to essential features, primarily focusing on obtaining the user's location and finding nearby salons. This screen is designed for user convenience and efficiency, enabling them to discover and engage with salon services effortlessly.

Requirement #:	Req03	Requirement Type:	Functional
Description:	Main Screen		
Rationale:	This Requirement is also important because after login main screen will appear with different categories.		
Originator:	User		
Fit Criterion:	User will see different categories on main screen of app.		
Customer Satisfaction:	5	Customer Dissatisfaction:	10
Priority:	3	Conflicts:	Nil
Supporting Materials:	N/A		
History:	N/A		

Table 3.3 Main Screen

Requirement Name: Salon List

The Salon List is a vital component of the Salon App, offering users a comprehensive view of nearby salons and their available services. This feature enables users to browse, compare, and make informed decisions when selecting a salon for their beauty needs.

Requirement #:	Req04	Requirement Type:	Functional
Description:	This requirement is very important because user can find different salons.		
Rationale:	This requirement is important because user can find different salons.		
Originator:	User		
Fit Criterion:			
Customer Satisfaction:	5	Customer Dissatisfaction:	10
Priority:	3	Conflicts:	Nil
Supporting Materials:	N/A		
History:	N/A		

Table 3.4 Salon List

Requirement name: Reservation

Requirement #:	Req05	Requirement Type:	Functional
Description:	Reservation		
Rationale:	User can send Reservation salons according to his/her need easily.		
Originator:	User		
Customer Satisfaction:	5	Customer Dissatisfaction:	0
Priority:	3	Conflicts:	Nil
Supporting Materials:	N/A		
History:	N/A		

Table 3.6 Booking

2.2 Non-Functional Requirements:

Non-functional requirements are often called "Quality Attributes" of a system. Evolution qualities, such as testability, maintainability, extensibility and scalability, which are embodied in the static structure of the software system.

2.2.1 Accessibility:

It should be easily accessible from everywhere where internet is available. User will be able to access our application even if they do not have an internet connection or if they were previously logged in.

2.2.2 Security:

The application is secured by using email and password so no one can harm the data of the user. Also, only user can change his or her password by using forgot password option.

2.3 System Requirements:

This unit lists the minimum software and hardware requirements needed to run our application. Location based to do list for Android.

- VS CODE
- Internet
- GPS / Location
- Android actual device e.g., android mobile, tablet etc.

Performance Requirement:

This document will provide general description of the project product perspective, and overview of requirement, general constraint and user view of the product while using. In addition will also provide the specific requirement and functional needs for this project such as interface, functional and performance requirements. The purpose of this software requirement specification is to properly document the requirement of the user necessary in order to build this application.

Software Requirement:

1. Development Environment:

Node.js: Required for running JavaScript and npm commands.

npm (Node Package Manager): Used for installing and managing project dependencies.

Git: Version control system for tracking changes in the codebase.

Code Editor: A code editor or integrated development environment (IDE) like Visual Studio Code or JetBrains WebStorm for writing and debugging React Native code.

2. React Native Framework:

- React Native: A powerful open-source framework for mobile app development.
- Cross-Platform Compatibility: Write code once, run it on iOS and Android.
- Native-Like Performance: Apps built with React Native perform like native apps.
- Reusable Components: Rich library of customizable UI components.
- Hot Reloading: Real-time code changes for faster development.
- Community and Ecosystem: Active developer community and extensive library support.
- Open Source: Continuously improved with regular updates.
- Support for Native Modules: Access device features when needed.
- Simplified Development: Single codebase for iOS and Android apps.
- Efficiency and High Quality: Deliver great user experiences on multiple devices.

3. Expo:

Expo CLI (if using Expo): Provides a set of tools and libraries for simplifying the development process and accessing device features.

4. React Navigation:

React Navigation library for implementing app navigation and routing.

CHAPTER-3

DESIGN - APP SCREENSHOTS

USER SIGN UP



Beautitioner
Beautyboud

SIGNUP


Already have an account?

GO TO LOGIN



USER SIGNIN





Beautitioner
Beautyboud

[SIGN IN](#)

Dont have account? Create One

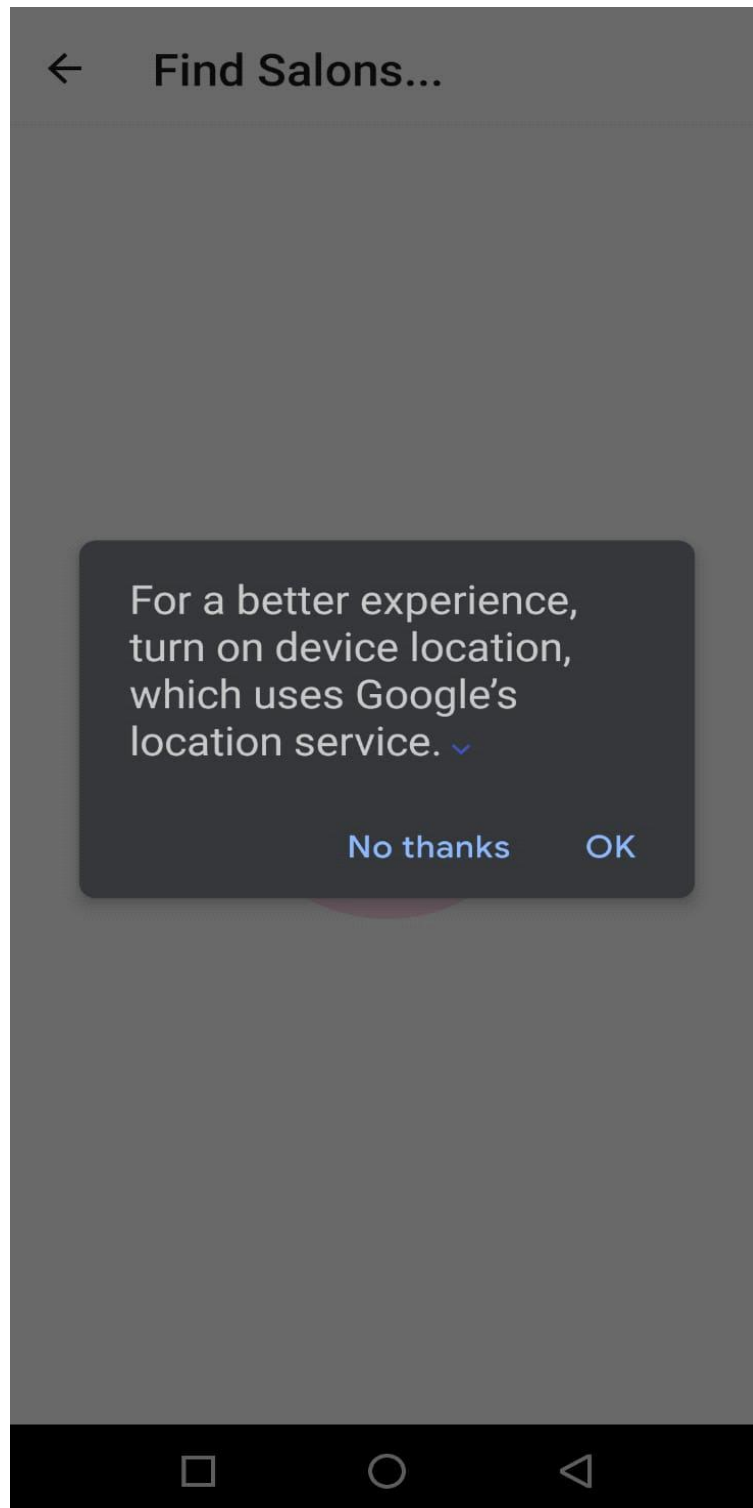
[CREATE AN ACCOUNT](#)

MAIN SCREEN

← Find Salons...



LOCATION PERMISSION



SALOON LIST

← Select Salon...

Alle'nora Annie's Signature Salon
Faisalabad



Rating : 4.1

Status : **Open**

36-D Imam Hussain Road, Faisalabad

SEE ON MAP

Teejee`s Signature Salon



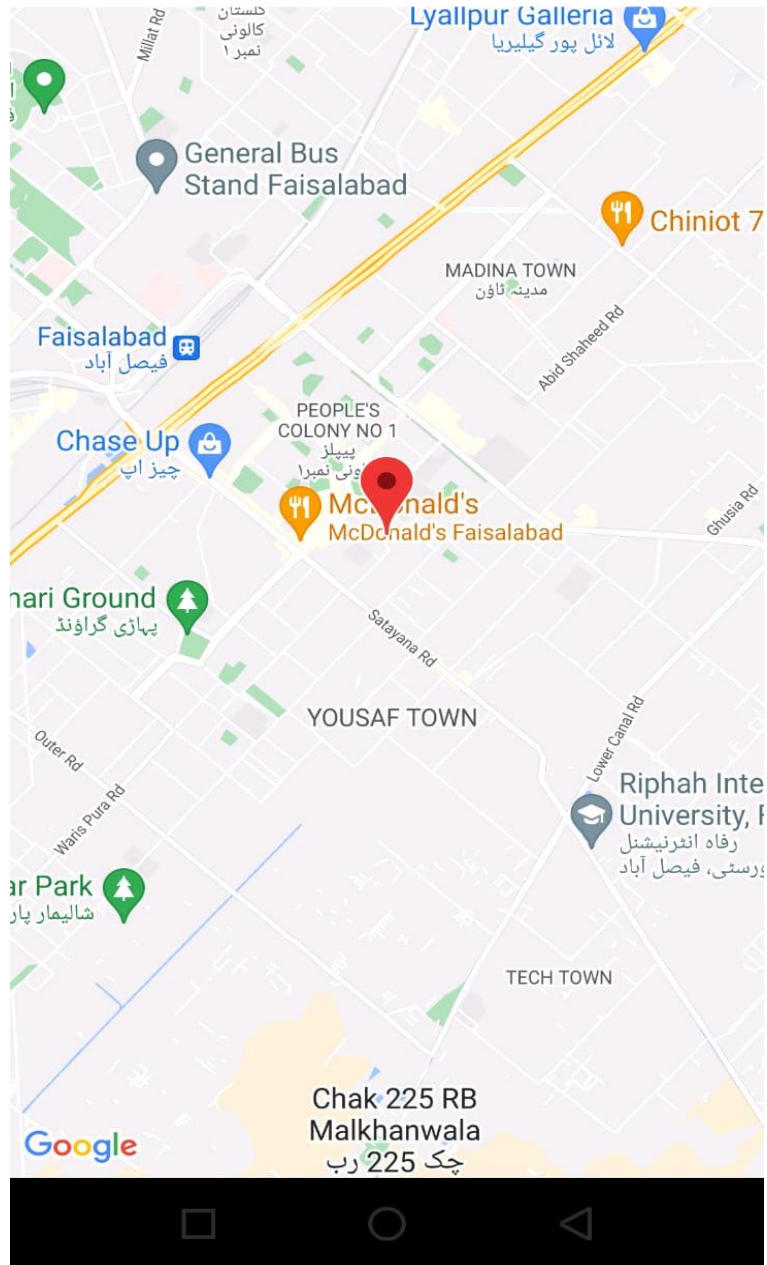
Rating : 4.6

Status : **Open**

SEE LOCATION OF SALON

← Location

RESERVE



RESERVATION FORM

← Reservation

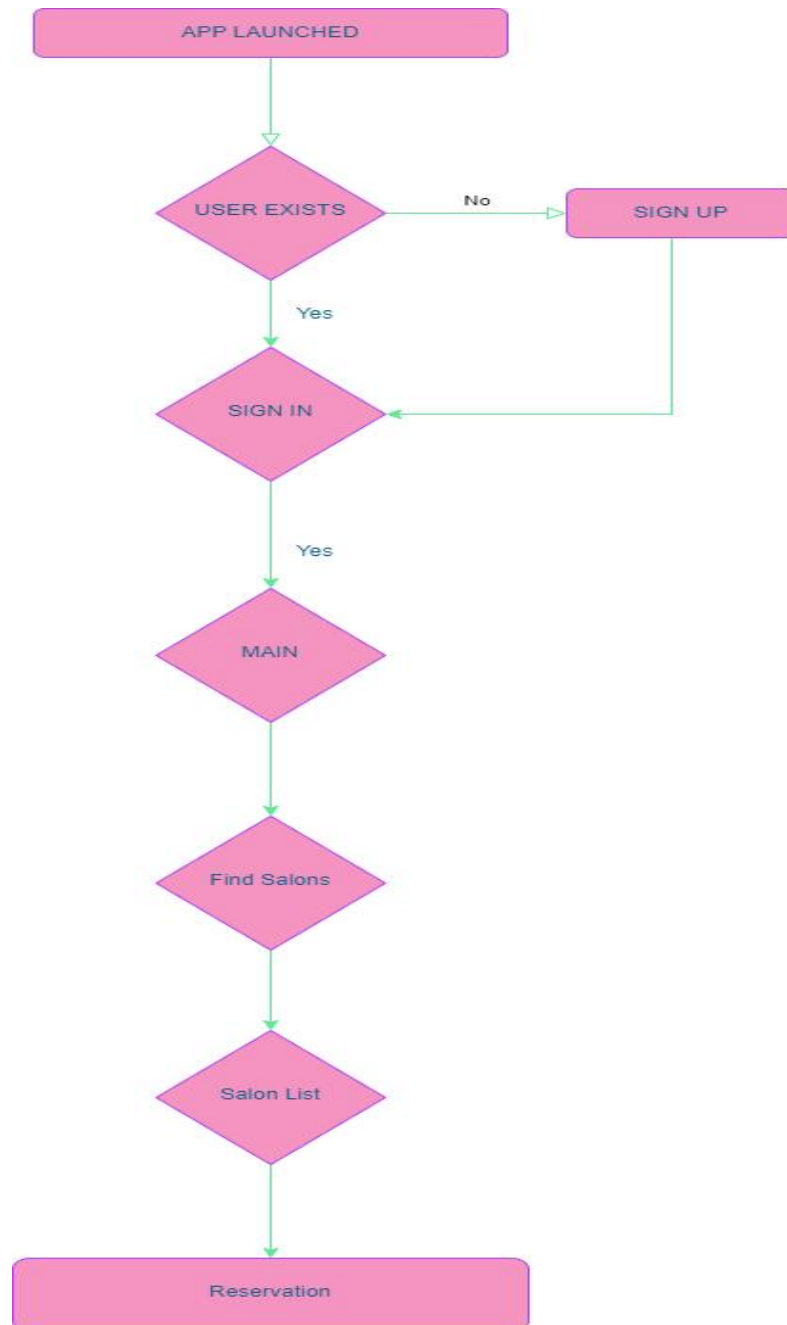
Enter your name...

'08:30'/'AM - PM'/'30/12/2023'

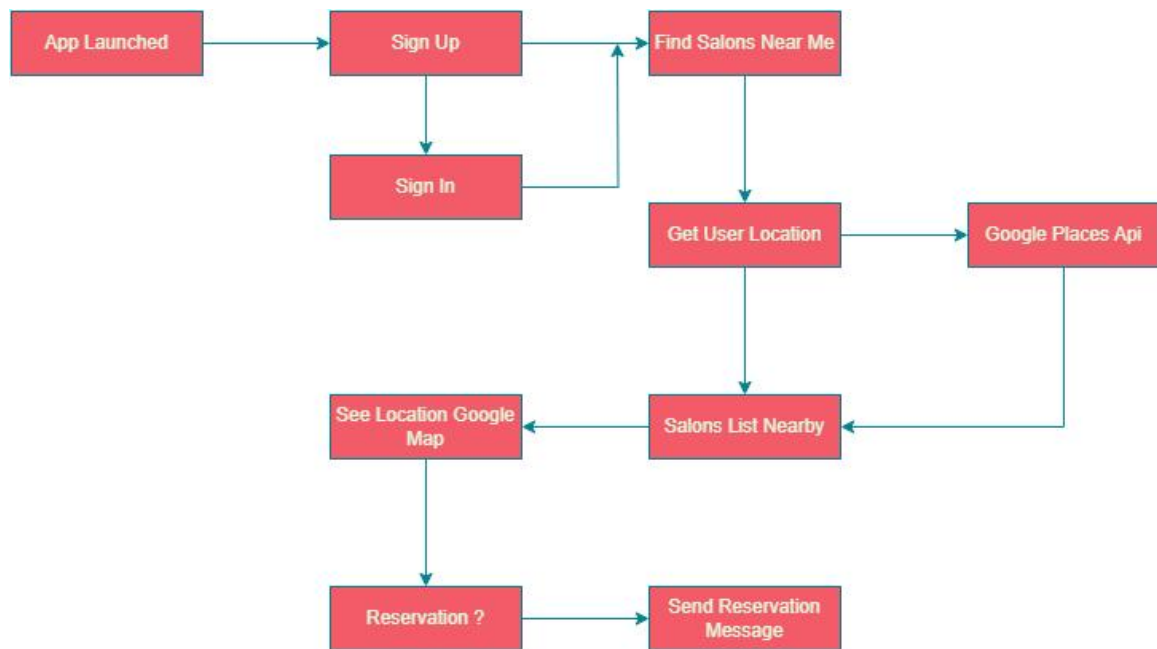
Reserve

CHAPTER - 4 DIAGRAMS

FLOW CHART DIAGRAM



WORKING OF APP



CHAPTER - 5 CODING

SIGN UP

```
import React from "react";

import { TextInput, Button,

  Alert,

  StyleSheet,

  SafeAreaView,

  Text,

  Image,

  KeyboardAvoidingView,

  View,

  ToastAndroid,

} from "react-native";

import axios from "axios";

import logo from "../assets/beautybudd-logo.png";

import { useNavigation } from "@react-navigation/native";

import USERS from "../data/Users";

import { AsyncStorage } from "react-native";

// signup component function
```



```

function Signup() {

    const navigation = useNavigation();

    const [userExists, setUserExists] = React.useState(false);

    const [email, setEmail] = React.useState("");

    const [password, setPassword] = React.useState("");

    const [passwordConfirm, setPasswordConfirm] = React.useState("");


    // handling new user

    async function signUpHandler() {

        if (!email.includes("@")) {

            return Alert.alert("Email Validity", "Please use valid email.");

        }

        if (password.length < 6) {

            return Alert.alert(

                "Password Validity",

                "Password should be > 6 characters."

            );

        }

        if (password !== passwordConfirm) {

            return Alert.alert(

                "Password Validity",

```

```

        "Your both passwords should match."

    );

}

for (var i = 0; i < USERS.length; i++) {

    if (USERS[i].email === email) {

        setUserExists(true);

    }

}

if (userExists) {

    return Alert.alert("User already exists", "Try Different email");

}

const user = {

    email,

    password,

};

setEmail("");

setPassword("");

setPasswordConfirm("");

USERS.push(user);

ToastAndroid.show("Account created successfully!", ToastAndroid.SHORT);

return navigation.navigate("signin");

```

```

}

const HAVE_ACCOUNT = "Already have an account?";

// signup ui components

return (

  <KeyboardAvoidingView>

    <Image style={styles.logo} source={logo} />

    <SafeAreaView>

      <TextInput

        style={styles.input}

        placeholder="Email"

        placeholderTextColor="#FF3FA4"

        value={email}

        onChangeText={setEmail}

      />

      <TextInput

        style={styles.input}

        placeholder="Password"

        secureTextEntry={true}

        placeholderTextColor="#FF3FA4"

        value={password}

        onChangeText={setPassword}

```

```
/>
```

```
<TextInput
```

```
    style={styles.input}
```

```
    placeholder="Repeat Password"
```

```
    secureTextEntry={true}
```

```
    placeholderTextColor="#FF3FA4"
```

```
    value={passwordConfirm}
```

```
    onChangeText={setPasswordConfirm}
```

```
/>
```

```
<View
```

```
    style={{
```

```
        width: "100%",
```

```
        justifyContent: "center",
```

```
        alignItems: "center",
```

```
    }}  
>
```

```
<Button
```

```
    style={styles.button}
```

```
    title="Signup"
```

```
    color="#FF3FA4"
```

```
    accessibilityLabel="Learn more about this purple button"
```

```

        onPress={signUpHandler}

      />

</View>

<Text style={{ textAlign: "center", padding:
10 }}>{HAVE_ACCOUNT}</Text>

<View

  style={{

    width: "100%",

    justifyContent: "center",

    alignItems: "center",

  }}

>

  <Button

    style={styles.button}

    onPress={() => {

      navigation.navigate("signin");

    }}

    title="Go To Login"

    color="#FF3FA4"

    accessibilityLabel="Learn more about this purple button"

```

```

        />

    </View>

</SafeAreaView>

</KeyboardAvoidingView>

);

}

```

```

const styles = StyleSheet.create({

  input: {

    height: 40,

    margin: 12,

    borderWidth: 1,

    padding: 10,

    borderColor: "#FF3FA4",

    color: "#FF3FA4",

  },

  button: {

    padding: 5,

    margin: 12,

  },

  logo: {

```

```
        height: 150,  
  
        width: 335,  
  
        alignSelf: "center",  
  
        marginTop: 10,  
  
    },  
  
});
```

export default Signup;

SIGN IN

```
import React, { useState, useEffect } from "react";  
  
import {  
  
    TextInput,  
  
    Button,  
  
    Alert,  
  
    StyleSheet,  
  
    SafeAreaView,  
  
    Text,  
  
    Keyboard,  
  
    View,  
  
    Image,
```

```

} from "react-native";

import axios from "axios";

import logo from "../assets/beautybudd-logo.png";

import { useNavigation } from "@react-navigation/native";

import USERS from "../data/Users";

function Signin() {

  const navigation = useNavigation();

  const [email, setEmail] = React.useState("");

  const [password, setPassword] = React.useState("");

  const [keyboardStatus, setKeyboardStatus] = useState("");

  async function signInHandler() {

    if (!email.includes("@")) {

      return Alert.alert("Email Validity", "Please use valid email.");

    }

    if (password.length < 6) {

      return Alert.alert(

        "Password Validity",

        "Password should be > 6 characters."

      );

```



```

    }

    const index = USERS.findIndex((obj) => obj.email === email);

    if (index === -1) {

        Alert.alert("No User Exists", "Create account first!");

    } else {

        return navigation.navigate("main");

    }

}

const DONT_HAVE_ACCOUNT_TEXT = "Dont have account? Create One";

return (

    <SafeAreaView>

        <Image style={styles.logo} source={logo} />

        <TextInput

            style={styles.input}

            onChangeText={setEmail}

            placeholder="email"

            value={email}

            onSubmitEditing={Keyboard.dismiss}

            selectTextOnFocus

            placeholderTextColor="#FF3FA4"

```

```
/>
```

```
<TextInput
```

```
    style={styles.input}
```

```
    onChangeText={setPassword}
```

```
    placeholder="Password"
```

```
    value={password}
```

```
    onSubmitEditing={Keyboard.dismiss}
```

```
    selectTextOnFocus
```

```
    secureTextEntry={true}
```

```
    placeholderTextColor="#FF3FA4"
```

```
/>
```

```
<View
```

```
    style={{
```

```
        width: "100%",
```

```
        justifyContent: "center",
```

```
        alignItems: "center",
```

```
    }}  
>
```

```
<Button
```

```
    style={styles.button}
```

```
    onPress={signInHandler}
```

```

        title="Sign In"

        color="#FF3FA4"

        accessibilityLabel="Learn more about this purple button"

    />

</View>

<Text style={{ textAlign: "center", padding: 10 }}>

    {DONT_HAVE_ACCOUNT_TEXT}

</Text>

<View

    style={{

        width: "100%",

        justifyContent: "center",

        alignItems: "center",

    }}

>

    <Button

        style={styles.button}

        onPress={() => {

            navigation.navigate("signup");

        }}

```

```

        title="Create an Account"

        color="#FF3FA4"

        accessibilityLabel="Learn more about this purple button"

    />

</View>

<Text style={{ textAlign: "center", padding: 10 }}>{keyboardStatus}</Text>

</SafeAreaView>

);

}

```

```

const styles = StyleSheet.create({

  input: {

    height: 40,

    margin: 12,

    borderWidth: 1,

    padding: 10,

    borderColor: "#FF3FA4",

    color: "#FF3FA4",

  },

  button: {

```

```

        padding: 5,

        margin: 12,

    },

    logo: {

        height: 150,

        width: 335,

        alignSelf: "center",

        marginTop: 10,

    },

});

```

export default Signin;

MAIN PAGE

```

import react from "react";

import { View, StyleSheet } from "react-native";

import Nearme from "../Nearme";

const Main = () => {

    return (

        <View style={styles.wrapper}>

```

```

        <Nearme />

    </View>

    );

};

const styles = StyleSheet.create({

    wrapper: {

        backgroundColor: "#F1EFEF",

        color: "black",

    },

    text: {

        backgroundColor: "#8b0000",

        borderRadius: 10,

    },

});

```

export default Main;

SALON

```

import restaurant from "../assets/wp1874196.jpg";

import { View, Text, Image, TouchableOpacity, StyleSheet } from "react-native";

import MapButton from "../MapButton";

```

```

import { useNavigation } from "@react-navigation/native";

function Salon({ item }) {

  const logo = item.icon;

  return (

    <View

      style={{

        backgroundColor: "white",

        width: "90%",

        alignSelf: "center",

        marginTop: 20,

        borderRadius: 10,

        shadowColor: "rgba(46, 229, 157, 0.4)",

        shadowOpacity: 1.5,

        elevation: 8,

        shadowRadius: 20,

        shadowOffset: { width: 1, height: 13 },

      }}

    >

    <Text

```

```

        style={{
            fontWeight: "600",

            padding: 10,

            borderBottomWidth: 2,

            borderBottomColor: "#F1EFEF",

        }}
    >

        {item.name}

</Text>

<Image

    style={{ width: "auto", height: 180 }}

    source={{ uri: logo }}

></Image>

<View

    style={{

        flexDirection: "row",

        justifyContent: "space-between",

        borderBottomWidth: 2,

        borderBottomColor: "#F1EFEF",

    }}

    >

```



```

<Text style={{ padding: 10, fontWeight: "600" }}>

    Rating : {item.rating}

</Text>

<Text style={{ padding: 10, fontWeight: "600" }}>

    Status : <Text style={{ color: "green" }}>Open</Text>

</Text>

</View>

<View style={{ paddingTop: 10, paddingLeft: 10, paddingRight: 10 }}>

    <Text>{item.vicinity}</Text>

</View>

<View

    style={{

        flexDirection: "row",

        justifyContent: "center",

        padding: 10,

    }}

>

    <MapButton

        location={item.geometry.location}

        name={item.name}

    ></MapButton>

```

```

        </View>

    </View>

    );
}

```

export default Salon;

SALON LIST

```

import React from "react";

import { ScrollView, FlatList, View } from "react-native";

import Salon from "../Salon";

import DATA_SET from "../data/SalonsDataset";

function Salons() {

    return (

        <View>

            <FlatList data={DATA_SET} renderItem={Salon} />

        </View>

    );

}

```

export default Salons;

LOCATION

```
import React from "react";
```

```
import { View, Text, StyleSheet, TouchableOpacity } from "react-native";
```

```
import MapView, { Marker } from "react-native-maps";
```

```
import { useRoute, useNavigation } from "@react-navigation/native";
```

```
function Map() {
```

```
    const route = useRoute();
```

```
    const nav = useNavigation();
```

```
    const mapData = route.params;
```

```
    const loc = {
```

```
        lat: mapData.coords.lat,
```

```
        lng: mapData.coords.lng,
```

```
    };
```

```
    return (
```

```
        <View style={styles.container}>
```

```
            <MapView
```

```
                style={styles.map}
```

```
                initialRegion={{
```

```

        latitude: loc.lat,

        longitude: loc.lng,

        latitudeDelta: 0.0922,

        longitudeDelta: 0.0421,

    }}
>

<Marker

    coordinate={{

        latitude: loc.lat,

        longitude: loc.lng,

    }}

    title="Salon's Location"

    description="Faisalabad Pakistan"

/>

</MapView>

<View

    style={{

        position: "absolute",

        left: 0,

        top: 0,

        backgroundColor: "white",

```

```

        width: "100%",

        padding: 10,

    }}

>

<TouchableOpacity

    onPress={() => {

        nav.navigate("form");

    }}

    style={{

        backgroundColor: "green",

        padding: 8,

        width: "40%",

        borderRadius: 10,

        alignSelf: "center",

    }}

>

<Text

    style={{ textAlign: "center", color: "white", fontWeight:

"600" }}

>

RESERVE

```

```

        </Text>

        </TouchableOpacity>

    </View>

</View>

);

}

```

```

const styles = StyleSheet.create({

    container: {

        flex: 1,

    },

    map: {

        width: "100%",

        height: "100%",

    },

});

```

export default Map;

RESERVATION

```

import React from "react";

```

```

import {

    View,

    Text,

    TextInput,

    TouchableOpacity,

    StyleSheet,

} from "react-native";

import Communications, { phonecall } from "react-native-communications";

function ReservationForm() {

    const [name, setName] = React.useState("");

    const [message, setMessage] = React.useState("");

    const reservationHandler = () => {

        Communications.text("+3456000789", "Here is the reservation message");

    };

    return (

        <View style={styles.container}>

            <View style={styles.inner}>

                <TextInput

                    onChangeText={setName}

```

```

        value={name}

        style={styles.input}

        placeholder="Enter your name..."

    />

    <TextInput

        onChangeText={setMessage}

        value={message}

        style={styles.input}

        placeholder=""08:30/'AM - PM/'30/12/2023""

    />

    <TouchableOpacity style={styles.button}
onPress={reservationHandler}>

        <Text style={styles.text}>Reserve</Text>

    </TouchableOpacity>

</View>

</View>

);

}

const styles = StyleSheet.create({

```



```
    container: { flex: 1 },

    inner: { padding: 10 },

    input: {

        borderWidth: 1,

        marginTop: 10,

        padding: 5,

        marginBottom: 10,

    },

    button: {

        backgroundColor: "#FF3FA4",

        padding: 8,

        marginTop: 10,

        marginBottom: 10,

    },

    text: {

        color: "white",

        textAlign: "center",

    },

});
```

export default ReservationForm;

PARENT WRAPPER

```
import { StatusBar } from "expo-status-bar";

import * as React from "react";

import { NavigationContainer } from "@react-navigation/native";

import { createStackNavigator } from "@react-navigation/stack";

import Signup from "../components/Signup";

import Signin from "../components/SignIn";

import Main from "../components/Main";

import Salons from "../components/Salons";

import Map from "../components/Map";

import ReservationForm from "../components/ReservationForm";

const Stack = createStackNavigator();

import { StyleSheet, Text, View } from "react-native";

export default function App() {

  return (

    <NavigationContainer>

      <Stack.Navigator>

        <Stack.Screen
```

```

        name="signup"

        component={Signup}

        options={{ headerTitle: "" }}

    />

    <Stack.Screen

        name="signin"

        component={Signin}

        options={{ headerTitle: "" }}

    />

    <Stack.Screen

        name="main"

        component={Main}

        options={{ headerTitle: "Find Salons..." }}

    />

    <Stack.Screen

        name="salonList"

        component={Salons}

        options={{ headerTitle: "Select Salon..." }}

    />

    <Stack.Screen

        name="map"

```

```

        component={Map}

        options={{ headerTitle: "Location" }}

    />

    <Stack.Screen

        name="form"

        component={ReservationForm}

        options={{ headerTitle: "Reservation" }}

    />

</Stack.Navigator>

</NavigationContainer>

);

}

```

DEPENDENCIES INSTALLATION DETAILS

```

{
  "name": "beautybudd",
  "version": "1.0.0",
  "main": "node_modules/expo/AppEntry.js",
  "scripts": {
    "start": "expo start",
    "android": "expo start --android",
    "ios": "expo start --ios",
    "web": "expo start --web"
  },
  "dependencies": {
    "@react-native-async-storage/async-storage": "^1.19.3",
    "@react-native-firebase/app": "^18.5.0",
    "@react-native-firebase/auth": "^18.5.0",
    "@react-navigation/native": "^6.1.8",
    "@react-navigation/stack": "^6.3.18",

```

```

    "axios": "^1.5.1",
    "expo": "~49.0.13",
    "expo-location": "~16.1.0",
    "expo-status-bar": "~1.6.0",
    "react": "^18.2.0",
    "react-native": "^0.72.5",
    "react-native-communications": "^2.2.1",
    "react-native-maps": "1.7.1",
    "react-native-safe-area-context": "4.6.3",
    "react-native-screens": "~3.22.0"
  },
  "devDependencies": {
    "@babel/core": "^7.20.0"
  },
  "private": true
}

```

```

{
  "name": "beautybudd",
  "version": "1.0.0",
  "lockfileVersion": 3,
  "requires": true,
  "packages": {
    "": {
      "name": "beautybudd",
      "version": "1.0.0",
      "dependencies": {
        "@react-native-async-storage/async-storage": "^1.19.3",
        "@react-native-firebase/app": "^18.5.0",
        "@react-native-firebase/auth": "^18.5.0",
        "@react-navigation/native": "^6.1.8",
        "@react-navigation/stack": "^6.3.18",
        "axios": "^1.5.1",
        "expo": "~49.0.13",
        "expo-location": "~16.1.0",
        "expo-status-bar": "~1.6.0",
        "react": "^18.2.0",
        "react-native": "^0.72.5",
        "react-native-communications": "^2.2.1",
        "react-native-maps": "1.7.1",
        "react-native-safe-area-context": "4.6.3",
        "react-native-screens": "~3.22.0"
      }
    }
  }
}

```

```

    },
    "devDependencies": {
      "@babel/core": "^7.20.0"
    }
  },
  "node_modules/@ampproject/remapping": {
    "version": "2.2.1",
    "resolved": "https://registry.npmjs.org/@ampproject/remapping/-/remapping-2.2.1.tgz",
    "integrity": "sha512-lFMjJTrFL3j7L9yBxwYfCq2k6qqwHyzuU1/XBnif78PWTJJYl/dfowQHWE3sp6U6ZzqWiiI
ZnpTM096zhkjw==",
    "dependencies": {
      "@jridgewell/gen-mapping": "^0.3.0",
      "@jridgewell/trace-mapping": "^0.3.9"
    },
    "engines": {
      "node": ">=6.0.0"
    }
  },
  "node_modules/@babel/code-frame": {
    "version": "7.22.13",
    "resolved": "https://registry.npmjs.org/@babel/code-frame/-/code-frame-7.22.13.tgz",
    "integrity": "sha512-XktuhWlJ5g+3TJXc5upd9Ks1HutSArif6jf2eAjYfYIO4ej3RN+184cZbzDvbPnuTJIUhp
KKJE3cIsYTiAT3w==",
    "dependencies": {
      "@babel/highlight": "^7.22.13",
      "chalk": "^2.4.2"
    },
    "engines": {
      "node": ">=6.9.0"
    }
  },
  "node_modules/@babel/compat-data": {
    "version": "7.22.20",
    "resolved": "https://registry.npmjs.org/@babel/compat-data/-/compat-data-7.22.20.tgz",
    "integrity": "sha512-BQYjKbpXjoXwFW5jGqiizJQQT/aC7pFm90k10WssonuguICi2641bgMzRp2ZMmRS1fkX6Ds
WDDcsrctK8Rwfiw==",
    "engines": {
      "node": ">=6.9.0"
    }
  }

```

```

    }
  },
  "node_modules/@babel/core": {
    "version": "7.23.0",
    "resolved": "https://registry.npmjs.org/@babel/core/-/core-7.23.0.tgz",
    "integrity": "sha512-97z/ju/Jy1rZmDxybphrBuI+jtJjFVoz7Mr9yUQVVVi+DNZE333uFQeMOqcCIy1x3WYBIbWftUSLmbNXNT7qFQ==",
    "dependencies": {
      "@ampproject/remapping": "^2.2.0",
      "@babel/code-frame": "^7.22.13",
      "@babel/generator": "^7.23.0",
      "@babel/helper-compilation-targets": "^7.22.15",
      "@babel/helper-module-transforms": "^7.23.0",
      "@babel/helpers": "^7.23.0",
      "@babel/parser": "^7.23.0",
      "@babel/template": "^7.22.15",
      "@babel/traverse": "^7.23.0",
      "@babel/types": "^7.23.0",
      "convert-source-map": "^2.0.0",
      "debug": "^4.1.0",
      "gensync": "^1.0.0-beta.2",
      "json5": "^2.2.3",
      "semver": "^6.3.1"
    }
  },

```

Test Case

Test Cases

Test Case:App Login

Test Case #:	1	Test Case Name:	Login User
---------------------	---	------------------------	------------

System:	Login	Sub System:	Login
Designed By:		Design Date:	
Executed By:		Execution Date:	
Short Description:	Enter email and password in the login form and press login button.		

Pre-Condition:	Email and Password
-----------------------	--------------------

Operating System:			Android / IOS		Environment:		React Native	
Software Tools & Technologies with version			Visual Studio Code, React Native.					
Step s	Strateg y (Test-to Pass / (Test-to-Fail)	Action	Input	Actual Result / System Response	Expected Result/ Expected System Response	Status (Pass/Fai l)	Remark s	
1	T-T-P	Email and passwor d	Email and passwor d	Successful ly login	Successfull y login	Pass	Test is pass	
2	T-T-F	Email and passwor	Email and passwor	Not login	Not login	Pass	Test is pass	

		d	d				
--	--	---	---	--	--	--	--

Table : Login

Comments -----

Test Case: Find Salons

Test Case #:	2	Test Case Name:	Salons Fetching
System:	User	Sub System:	User
Designed By:		Design Date:	
Executed By:		Execution Date:	
Short Description:	Fetching Salons		

Pre-Condition:	User is login
-----------------------	---------------

Operating System:		Android/IOS		Environment:		React Native	
Software Tools & Technologies with version		MS VSCODE, React Native					
Step s	Strateg y (Test-to-Pass) / (Test-	Actio n	Input	Actual Result / System	Expected Result/ Expected System	Statu s (Pass	Remark s

	to-Fail)			Response	Response	/	Fail)
1	T-T-P	Fetch Salon s	Location Coordinate s	Successfull y fetched.	Successfully fetched.	Pass	Test is pass
2	T-T-F	Fetch Salon s	Location Coordinate s	Show error	Error shown	Pass	Test is pass

Comments -----

Tools & Technologies

Programming Languages

- React Native (JAVASCRIPT)

Operating System & Tools

- VS CODE
- ANDROID
- IOS
- INTERNET

Appendix A: Glossary

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

Appendix C: Check List

<i>Check List</i>	Yes	No
<i>I. Starting/Ending Dates</i>		
<i>II. Project Scope</i>		
<i>III. Product modules (covering all aspects of scope)</i>		
<i>IV. System Features (covering scope)</i>		
<i>V. Interface Requirements</i>		
<i>VI. Non-Functional Requirements</i>		
<i>VII. WBS</i>		
<i>VIII. Tools and Technologies Detail (for implementation)</i>		
<i>IX. Plagiarism Report</i>		

Appendix D: Supervisory Committee

<i>For Approval of any two Consultant Teachers</i>	
<i>Teacher Consulted</i>	<i>Teacher Consulted</i>
<i>Name:</i> _____.	<i>Name:</i> _____.
<i>Designation:</i> _____.	<i>Designation:</i> _____.
<i>Comments:</i> _____	<i>Comments:</i> _____
_____	_____
_____	_____
_____	_____
_____	_____
<i>Signature:</i> _____	<i>Signature:</i> _____

(For office use only)

OUTRO

As a student at **"TIPS COLLEGE"**, embarking on the journey of creating **"BEAUTYBUDD APP"** has been both challenging and incredibly rewarding. Our team poured their hearts and minds into crafting an app that simplifies your salon and beauty experience.

It's our hope that you'll find **"BEAUTYBUDD APP"** not just as a mobile application but as a trusted companion for discovering nearby salons and enjoying beauty services. We're excited to share this creation with you, and your support means the world to us.

Your feedback and suggestions will drive our ongoing improvements. Thank you for being a part of our journey. We can't wait to see how **"BEAUTYBUDD APP"** enhances your salon experience.

Warm regards,

AYESHA IJAZ
