RAMAIAH INSTITUTE OF TECHNOLOGY, BANGALORE – 560054
(Autonomous Institute, Affiliated to VTU)

Department of Computer Science & Engineering

Internship Report

on

Mobile Application Development

Institutional Internship

STUDENT NAME : TANISHKA CHHAJER AND TEJASWINI V

USN :1MS24IM048 AND 1MS24IM049

Ramaiah Institute of Technology
(Autonomous Institute, Affiliated to VTU)

MSR Nagar, MSRIT Post, Bangalore-560054

August - 2025

RAMAIAH INSTITUTE OF TECHNOLOGY, BANGALORE – 560054

(Autonomous Institute, Affiliated to VTU)

Department of Computer Science & Engineering

CERTIFICATE

This is to certify that Ms.TANISHKA CHHAJER(1MS24IM048) and Ms TEJASWINI V(1MS24IM049) a student of Bachelor of Engineering, has successfully completed, 24 Hours: from 05.08.2025 to 12.08.2025 Intra Institutional Internship in Mobile Application Development from the Department of Computer Science & Engineering, M S Ramaiah Institute of Technology, Bangalore.

SL No.	Component	Maximum Marks	Marks Obtained
1	Continuous Evaluation	50	
2	Presentation	20	
3	Report	30	
Total Marks		100	

.

Signature of the Student with Date **Signature of the Faculty Co-Ordinator**

Signature of Head of the Department

OVERVIEW OF INTERNSHIP ACTIVITIES

OVERVIEW OF THE INTETERNSHIP

DATE	DAY	NAME OF THE TOPIC COMPLETED
5.8.25	Monday	Introduction to dart and flutter
6.8.25	Tuesday	Briefing on dart and flutter program
7.8.25	Wednesday	Scaffold and it’s applications
9.8.25	Thursday	Introduction to Animation
11.8.25	Friday	Firebase studio and internal Assisment
12.8.25	Saturday	Report submission

TABLE OF CONTENTS

Contents **Page No.**

Overview of project in terms of Implementation

INTRODUCTION

SkillSwap is a cross-platform mobile application designed for Android and iOS, built using Flutter and Firebase. It serves as a community-driven skill exchange platform, where users can offer their expertise in one area while learning from others in a different skill.

The app connects people with diverse talents such as programming, photography, cooking, design, music, and more. Users can list skills, browse categories, initiate swap requests, participate in community forums, join events, access learning guides, and even connect with mentors.

With a modern Material 3 design, teal-white theme, and real-time Firebase backend, SkillSwap ensures a smooth, interactive, and engaging experience.

INTERNSHIP ACTIVITIES / WORK UNDERTAKEN

1.Connect Learners & Experts – Create a network where knowledge can be exchanged without monetary transactions.

2.Promote Lifelong Learning – Encourage people to learn new skills and improve existing ones.

3.Build Communities – Provide a platform for like-minded individuals to connect, collaborate, and grow.

4.Encourage Mentorship – Support an environment where experienced individuals guide beginners.

Learning Outcomes from Building/Using the App

1.For Developers (Building the App)

2.Flutter Mastery – Understanding Material 3 widgets, responsive layouts, and cross-platform UI.

3.State Management – Practical experience using Riverpod/Provider for scalable app state handling.

4.Firebase Integration – Authentication, Firestore real-time database, and Firebase Storage for images.

5.Cloud Data Modeling – Designing collections and documents in Firestore to handle complex relationships.

6.Modular Architecture – Structuring a Flutter app with screens, widgets, services, and models for maintainability.

7.Geolocation & External APIs – Implementing local event filtering and external resource linking.

For Users (Using the App)

1.Skill Discovery – Learn about new skills through guides, forums, and community posts.

2.Practical Exchange – Swap skills directly with other users in a structured and fair system.

3.Community Engagement – Participate in challenges, events, and forums.

4.Personal Growth – Gain mentorship, share success stories, and receive feedback.

Challenges Faced During Development

1. Complex Feature Set – Integrating multiple modules like skills, swaps, guides, events, and forums while keeping UI simple.

2. Real-Time Synchronization – Maintaining consistent state between client-side and Firestore without performance issues.

3. Scalable Data Modeling – Designing Firestore collections to efficiently handle queries for filters, searches, and relationships.

4. Image Handling – Managing upload, retrieval, and caching of profile and skill images from Firebase Storage.
5. Geolocation Filtering – Implementing location-based event suggestions with accurate filtering.

6. User Authentication Flow – Designing secure, user-friendly sign-up/sign-in with Firebase Auth.

7. Cross-Platform Responsiveness – Ensuring the UI adapts to various screen sizes and operating systems.

FOLDER STRUCTURE

```
skillswap/
├── android/
├── ios/
├── lib/
│   ├── main.dart
│   ├── app.dart
│   ├── routes.dart
│   ├── constants/
│   │   └── app_colors.dart
│   ├── theme/
│   │   └── app_theme.dart
│   ├── models/
│   │   ├── user_model.dart
│   │   ├── skill_model.dart
│   │   ├── swap_request_model.dart
│   │   ├── guide_model.dart
│   │   ├── forum_models.dart
│   │   ├── event_model.dart
│   │   └── testimonial_model.dart
│   ├── services/
│   │   ├── auth_service.dart
│   │   ├── firestore_service.dart
│   │   ├── storage_service.dart
│   │   └── geolocation_service.dart
│   ├── providers/
│   │   ├── auth_provider.dart
│   │   ├── skills_provider.dart
│   │   ├── swap_provider.dart
│   │   ├── guides_provider.dart
│   │   ├── forum_provider.dart
│   │   └── events_provider.dart
│   ├── screens/
│   │   ├── splash_screen.dart
│   │   ├── login_screen.dart
│   │   ├── register_screen.dart
│   │   ├── home_screen.dart
│   │   ├── skills/
│   │   │   ├── skills_list_screen.dart
│   │   │   └── skill_detail_screen.dart
│   │   ├── forum_screen.dart
│   │   ├── events_screen.dart
│   │   └── profile_screen.dart
│   ├── widgets/
│   │   ├── skill_card.dart
│   │   ├── loading_indicator.dart
│   │   ├── bottom_nav.dart
│   │   └── app_drawer.dart
│   └── seed/
│       └── seed_data.dart
└── pubspec.yaml
```

main.dart

```
import 'package:flutter/material.dart';
import 'screens/splash_screen.dart';

void main() {
  runApp(SkillSwapApp());
}

class SkillSwapApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'SkillSwap',
      debugShowCheckedModeBanner: false,
      theme: ThemeData(primarySwatch: Colors.green),
      home: SplashScreen(),
    );
  }
}
```

Add_screen .dart

```
import 'package:flutter/material.dart';
import 'package:uuid/uuid.dart';

class AddSkillScreen extends StatefulWidget {
  @override
  _AddSkillScreenState createState() => _AddSkillScreenState();
}

class _AddSkillScreenState extends State<AddSkillScreen> {
  final _form = GlobalKey<FormState>();
  final _titleCtrl = TextEditingController();
  final _ownerCtrl = TextEditingController();
  String _category = 'General';

  void _save() {
    if (!_form.currentState!.validate()) return;
    final id = Uuid().v4();
    Navigator.pop(context, {'id': id, 'title': _titleCtrl.text.trim(), 'owner': _ownerCtrl.text.trim(), 'category':
_category});
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: const Text('Add Skill')),
      body: Padding(
        padding: const EdgeInsets.all(12),
        child: Form(
          key: _form,
          child: ListView(children: [
            TextFormField(controller: _titleCtrl, decoration: InputDecoration(labelText: 'Skill Title'), validator:
(v)=> v==null||v.trim().isEmpty? 'Enter title':null),
            SizedBox(height: 8),
            TextFormField(controller: _ownerCtrl, decoration: InputDecoration(labelText: 'Your Name'), validator:
(v)=> v==null||v.trim().isEmpty? 'Enter name':null),
            SizedBox(height: 8),
            DropdownButtonFormField<String>(
              value: _category,
              items:
['Music','Design','Cooking','Coding','General'].map((c)=>DropdownMenuItem(value:c,child:Text(c))).toList(),
              onChanged: (v)=> setState(()=> _category = v ?? 'General'),
              decoration: InputDecoration(labelText: 'Category'),
            ),
            SizedBox(height: 12),
            ElevatedButton(onPressed: _save, child: Text('Save')),
          ]),
        ),
      ),
    );
  }
}
```

Edit_screen.dart

```
import 'package:flutter/material.dart';

class EditSkillScreen extends StatefulWidget {
  final Map<String,String> skill;
  EditSkillScreen({required this.skill});
  @override
  _EditSkillScreenState createState() => _EditSkillScreenState();
}

class _EditSkillScreenState extends State<EditSkillScreen> {
  late TextEditingController _titleCtrl;
  late TextEditingController _ownerCtrl;
  String _category = 'General';

  @override
  void initState() {
    super.initState();
    _titleCtrl = TextEditingController(text: widget.skill['title']);
    _ownerCtrl = TextEditingController(text: widget.skill['owner']);
    _category = widget.skill['category'] ?? 'General';
  }

  void _save() {
    Navigator.pop(context, {'id': widget.skill['id']!, 'title': _titleCtrl.text.trim(), 'owner':
_ownerCtrl.text.trim(), 'category': _category});
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text('Edit Skill')),
      body: Padding(
        padding: EdgeInsets.all(12),
        child: Column(children: [
          TextField(controller: _titleCtrl, decoration: InputDecoration(labelText: 'Title')),
          TextField(controller: _ownerCtrl, decoration: InputDecoration(labelText: 'Owner')),
          DropdownButton<String>(
            value: _category,
            items:
['Music','Design','Cooking','Coding','General'].map((c)=>DropdownMenuItem(value:c,child:Text(c))).
toList(),
            onChanged: (v)=> setState(()=> _category = v ?? 'General'),
          ),
          ElevatedButton(onPressed: _save, child: Text('Save')),
        ]),
      ),
    );
  }
}
```

Home_screen.dart

```
class EditSkillScreen extends StatefulWidget {
  final Map<String,String> skill;
  EditSkillScreen({required this.skill});
  @override
  _EditSkillScreenState createState() => _EditSkillScreenState();
}

class _EditSkillScreenState extends State<EditSkillScreen> {
  late TextEditingController _titleCtrl;
  late TextEditingController _ownerCtrl;
  String _category = 'General';

  @override
  void initState() {
    super.initState();
    _titleCtrl = TextEditingController(text: widget.skill['title']);
    _ownerCtrl = TextEditingController(text: widget.skill['owner']);
    _category = widget.skill['category'] ?? 'General';
  }

  void _save() {
    Navigator.pop(context, {'id': widget.skill['id']!, 'title': _titleCtrl.text.trim(), 'owner': _ownerCtrl.text.trim(), 'category': _category});
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text('Edit Skill')),
      body: Padding(
        padding: EdgeInsets.all(12),
        child: Column(children: [
          TextField(controller: _titleCtrl, decoration: InputDecoration(labelText: 'Title')),
          TextField(controller: _ownerCtrl, decoration: InputDecoration(labelText: 'Owner')),
          DropdownButton<String>(
            value: _category,
            items:
['Music','Design','Cooking','Coding','General'].map((c)=>DropdownMenuItem(value:c,child:Text(c))).
toList(),
            onChanged: (v)=> setState(()=> _category = v ?? 'General'),
          ),
          ElevatedButton(onPressed: _save, child: Text('Save')),
        ]),
      ),
    );
  }
}
```

Login_screen.dart

```
import 'package:flutter/material.dart';
import 'signup_screen.dart';
import 'home_screen.dart';

class LoginScreen extends StatelessWidget {
  final TextEditingController emailCtrl = TextEditingController();
  final TextEditingController passCtrl = TextEditingController();

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: const Text('Login')),
      body: Padding(
        padding: const EdgeInsets.all(16),
        child: Column(
          children: [
            TextField(controller: emailCtrl, decoration: InputDecoration(labelText: "Email")),
            SizedBox(height: 8),
            TextField(controller: passCtrl, decoration: InputDecoration(labelText: "Password"),
obscureText: true),
            SizedBox(height: 20),
            ElevatedButton(
              onPressed: () {
                Navigator.pushReplacement(context, MaterialPageRoute(builder: (_) => HomeScreen()));
              },
              child: Text("Login"),
            ),
            TextButton(
              onPressed: () => Navigator.push(context, MaterialPageRoute(builder: (_) =>
SignUpScreen())),
              child: Text("Create Account"),
            )
          ],
        ),
      ),
    );
  }
}
```

My_skill.dart

```
import 'package:flutter/material.dart';
import 'edit_skill_screen.dart';

class MySkillsScreen extends StatefulWidget {
  @override
  _MySkillsScreenState createState() => _MySkillsScreenState();
}

class _MySkillsScreenState extends State<MySkillsScreen> {
  List<Map<String,String>> mySkills = [
    {'id':'m1','title':'Piano Level 1','owner':'You','category':'Music'},
    {'id':'m2','title':'C basics','owner':'You','category':'Coding'},
  ];

  void _edit(int idx) async {
    final res = await Navigator.push(context, MaterialPageRoute(builder: (_) => EditSkillScreen(skill:
mySkills[idx])));
    if (res != null && res is Map<String,String>) {
      setState(()=> mySkills[idx] = res);
    }
  }

  void _delete(int idx) {
    setState(()=> mySkills.removeAt(idx));
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: const Text('My Skills')),
      body: ListView.builder(
        padding: EdgeInsets.all(12),
        itemCount: mySkills.length,
        itemBuilder: (context, i) {
          final s = mySkills[i];
          return Card(
            child: ListTile(
              title: Text(s['title']!),
              subtitle: Text(s['category']!),
              trailing: Row(mainAxisSize: MainAxisSize.min, children: [
                IconButton(icon: Icon(Icons.edit), onPressed: ()=> _edit(i)),
                IconButton(icon: Icon(Icons.delete), onPressed: ()=> _delete(i)),
              ]),
            ),
          );
        },
      ),
    );
  }
}
```

Profile.dart

```
import 'package:flutter/material.dart';

class ProfileScreen extends StatelessWidget {
 @override
 Widget build(BuildContext context) {
  // Simple static profile for college submission
  return Scaffold(
   appBar: AppBar(title: const Text('Profile')),
   body: Padding(
    padding: EdgeInsets.all(12),
    child: Column(children: [
     CircleAvatar(radius: 40, child: Text('U')),
     SizedBox(height: 12),
     Text('Name: Guest User'),
     SizedBox(height: 6),
     Text('Email: guest@example.com'),
    ]),
   ),
  );
 }
}
```

Setting_screen.dart

```
import 'package:flutter/material.dart';

class SettingsScreen extends StatefulWidget {
 @override
 _SettingsScreenState createState() => _SettingsScreenState();
}

class _SettingsScreenState extends State<SettingsScreen> {
 bool dark = false;
 @override
 Widget build(BuildContext context) {
  return Scaffold(
   appBar: AppBar(title: const Text('Settings')),
   body: ListView(padding: EdgeInsets.all(12), children: [
    SwitchListTile(title: Text('Dark Mode'), value: dark, onChanged: (v)=> setState(()=> dark = v)),
    ListTile(title: Text('Version'), trailing: Text('1.0.0')),
   ]),
  );
 }
}
```

Sign_screen.dart

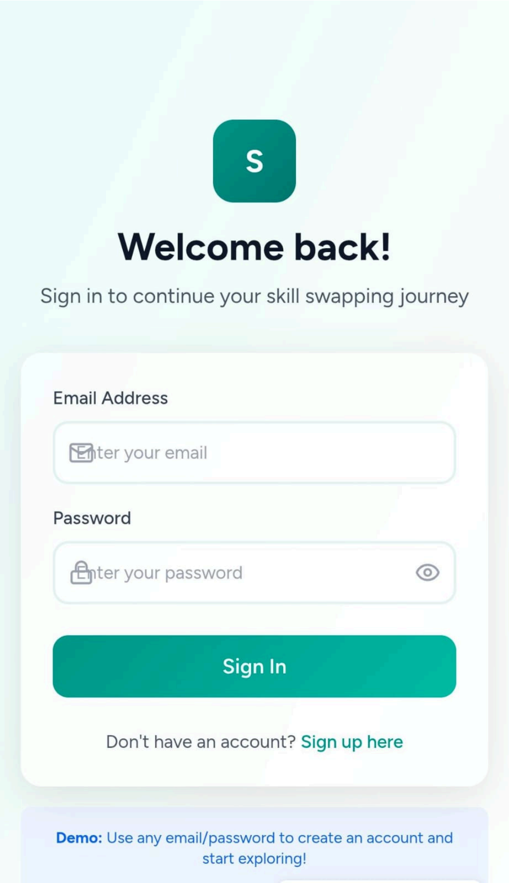
```
import 'package:flutter/material.dart';
import 'home_screen.dart';

class SignUpScreen extends StatelessWidget {
 final TextEditingController nameCtrl = TextEditingController();
 final TextEditingController emailCtrl = TextEditingController();
 final TextEditingController passCtrl = TextEditingController();

 @override
 Widget build(BuildContext context) {
  return Scaffold(
   appBar: AppBar(title: const Text('Sign Up')),
   body: Padding(
    padding: const EdgeInsets.all(16),
    child: Column(
     children: [
      TextField(controller: nameCtrl, decoration: InputDecoration(labelText: "Name")),
      SizedBox(height: 8),
      TextField(controller: emailCtrl, decoration: InputDecoration(labelText: "Email")),
      SizedBox(height: 8),
      TextField(controller: passCtrl, decoration: InputDecoration(labelText: "Password"),
obscureText: true),
      SizedBox(height: 20),
      ElevatedButton(
       onPressed: () => Navigator.pushReplacement(context, MaterialPageRoute(builder: (_) =>
HomeScreen())),
       child: Text("Sign Up"),
      ),
     ],
    ),
   ),
  );
 }
}
```

Skill_screen.dart

```
import 'package:flutter/material.dart';
import 'add_skill_screen.dart';

class SkillDetailsScreen extends StatelessWidget {
  final String skillId;
  final String title;
  final String owner;

  SkillDetailsScreen({required this.skillId, required this.title, required this.owner});

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text(title)),
      body: Padding(
        padding: EdgeInsets.all(12),
        child: Column(crossAxisAlignment: CrossAxisAlignment.start, children: [
          Text(title, style: Theme.of(context).textTheme.headlineSmall),
          SizedBox(height: 8),
          Text('By $owner'),
          SizedBox(height: 12),
          Text('Detailed description about the skill would appear here.'),
          Spacer(),
          Row(children: [
            ElevatedButton.icon(onPressed: () { Navigator.push(context, MaterialPageRoute(builder: (_)
=> AddSkillScreen())); }, icon: Icon(Icons.send), label: Text('Request')),
            SizedBox(width: 8),
            ElevatedButton.icon(onPressed: () { Navigator.pop(context); }, icon: Icon(Icons.arrow_back),
label: Text('Back')),
          ])
        ]),
      ),
    );
  }
}
```

Splash_screen.dart

```
class SplashScreen extends StatefulWidget {
  @override
  _SplashScreenState createState() => _SplashScreenState();
}

class _SplashScreenState extends State<SplashScreen> {
  @override
  void initState() {
   super.initState();
   Future.delayed(Duration(seconds: 2), () {
    Navigator.pushReplacement(context, MaterialPageRoute(builder: (_) => LoginScreen()));
   });
  }

  @override
  Widget build(BuildContext context) {
   return Scaffold(
    body: Center(
     child: Text("SkillSwap", style: TextStyle(fontSize: 30, fontWeight: FontWeight.bold)),
    ),
   );
  }
}
```

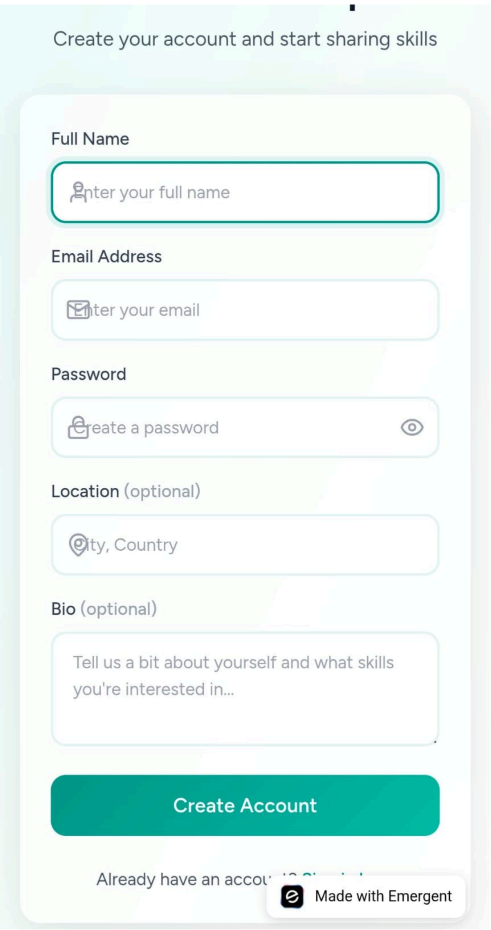


Welcome back!

Sign in to continue your skill swapping journey

Don't have an account? Sign up here

Demo: Use any email/password to create an account and start exploring!



Create your account and start sharing skills

Full Name

Enter your full name

Email Address

Enter your email

Password

Create a password

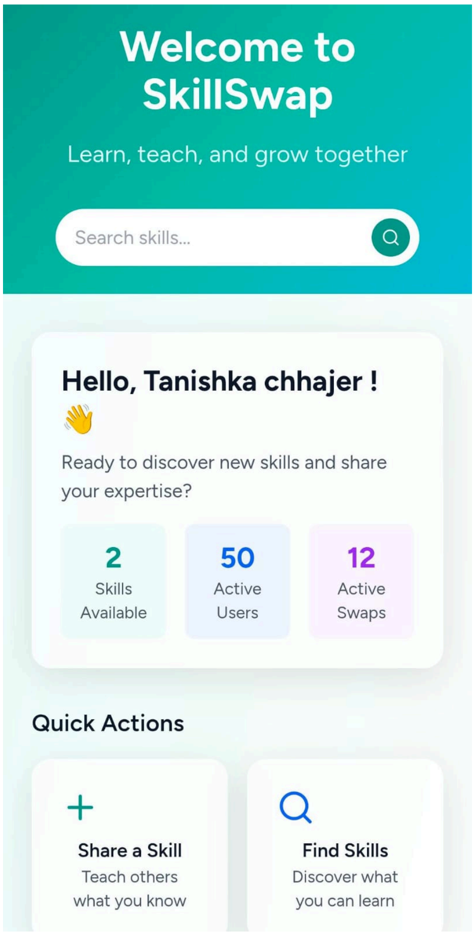
Location (optional)

City, Country

Bio (optional)

Tell us a bit about yourself and what skills you're interested in...

Already have an accou



Welcome to SkillSwap

Learn, teach, and grow together

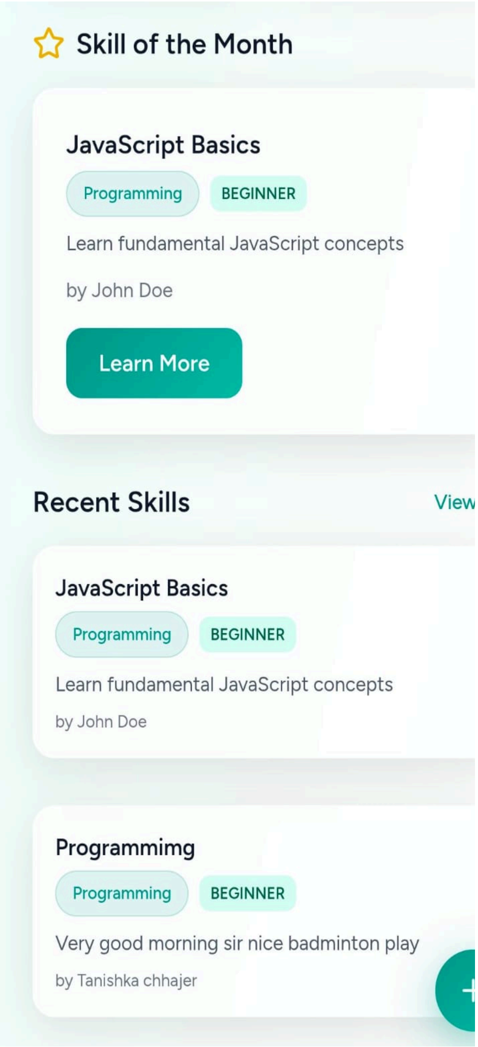
Hello, Tanishka chhajer ! 👋

Ready to discover new skills and share your expertise?

2 Skills Available	50 Active Users	12 Active Swaps

Quick Actions

+ Share a Skill Teach others what you know	Find Skills Discover what you can learn



JavaScript Basics

Learn fundamental JavaScript concepts

by John Doe

Recent Skills

JavaScript Basics

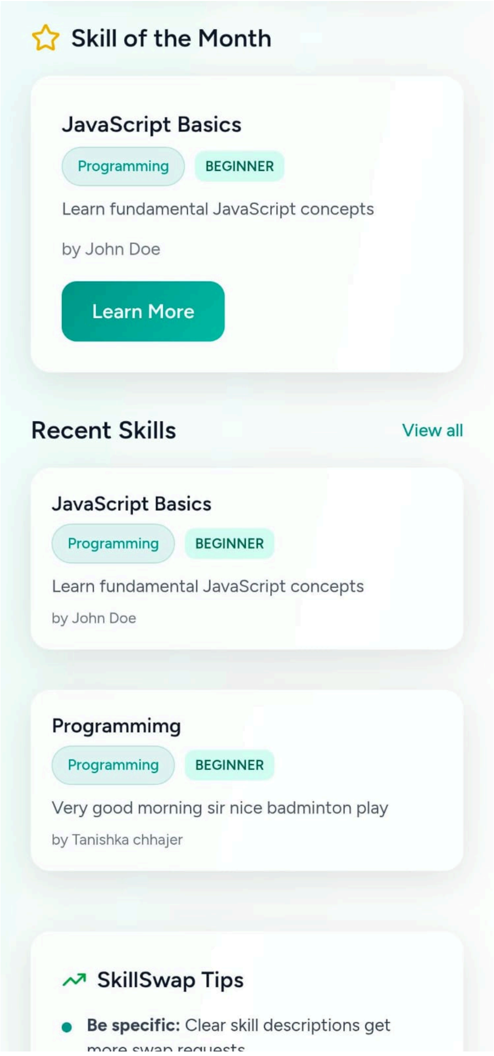
Learn fundamental JavaScript concepts

by John Doe

Programmimg

Very good morning sir nice badminton play

by Tanishka chhajer



JavaScript Basics

Learn fundamental JavaScript concepts

by John Doe

Recent Skills

JavaScript Basics

Learn fundamental JavaScript concepts

by John Doe

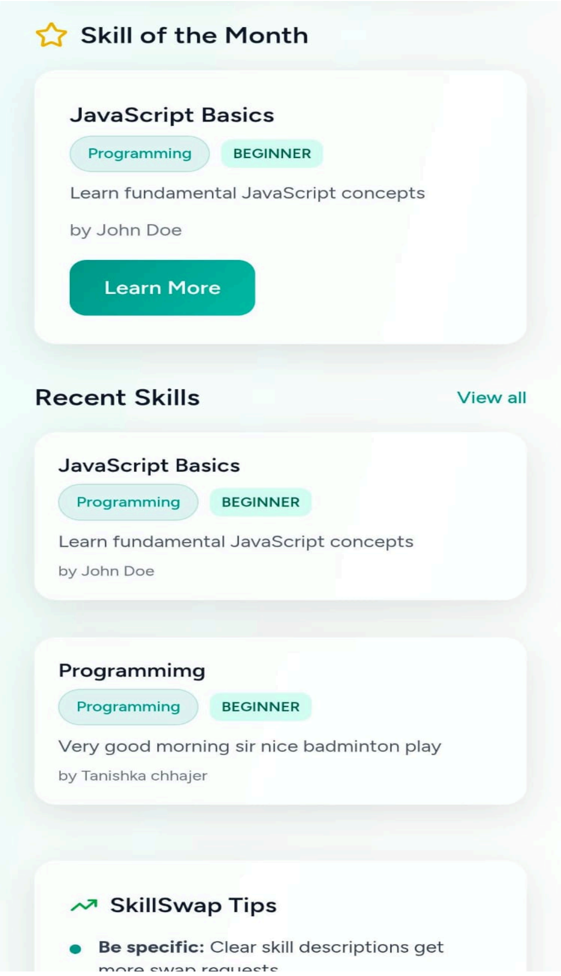
Programmimg

Very good morning sir nice badminton play

by Tanishka chhajer

SkillSwap Tips

- **Be specific:** Clear skill descriptions get more swap requests



☆ Skill of the Month

JavaScript Basics

Programming BEGINNER

Learn fundamental JavaScript concepts

by John Doe

Learn More

Recent Skills View all

JavaScript Basics

Programming BEGINNER

Learn fundamental JavaScript concepts

by John Doe

Programmimg

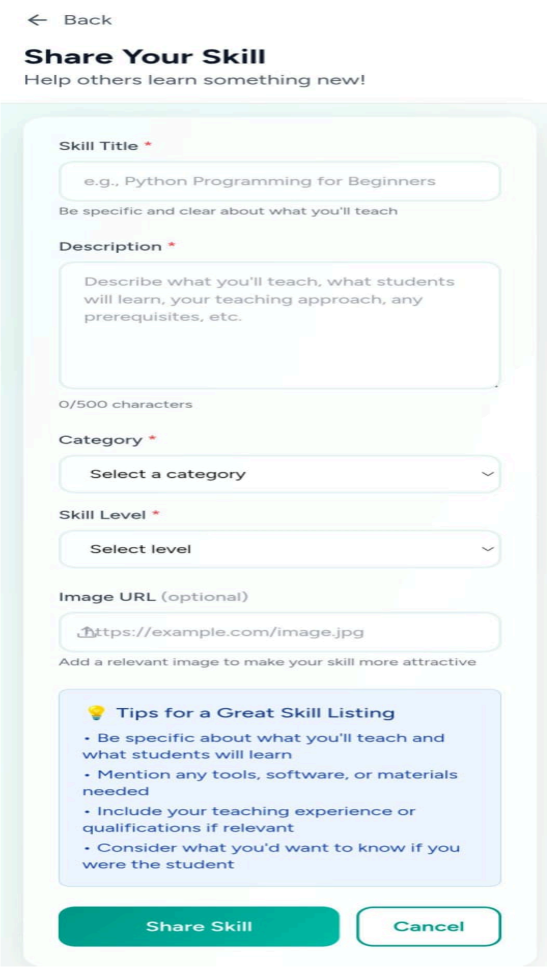
Programming BEGINNER

Very good morning sir nice badminton play

by Tanishka chhajer

SkillSwap Tips

- **Be specific:** Clear skill descriptions get more swap requests



← Back

Share Your Skill

Help others learn something new!

Skill Title *

e.g., Python Programming for Beginners

Be specific and clear about what you'll teach

Description *

Describe what you'll teach, what students will learn, your teaching approach, any prerequisites, etc.

0/500 characters

Category *

Select a category

Skill Level *

Select level

Image URL (optional)

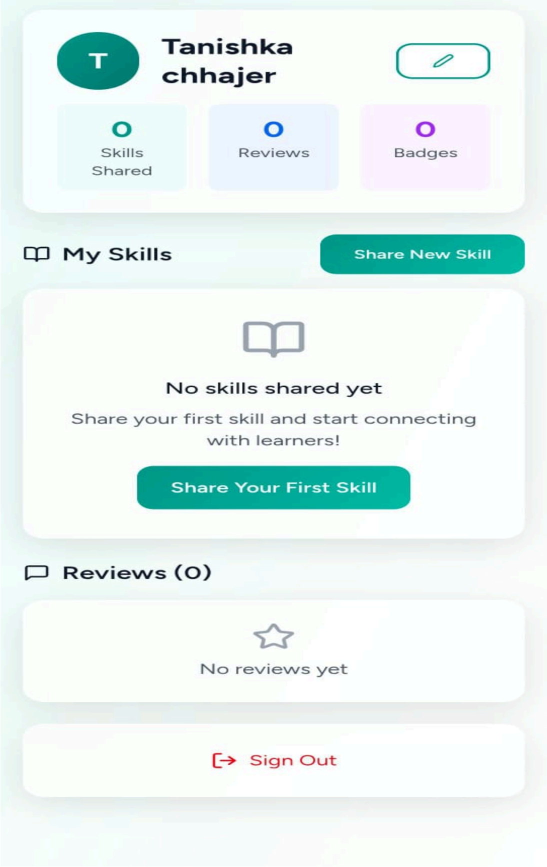
https://example.com/image.jpg

Add a relevant image to make your skill more attractive

💡 Tips for a Great Skill Listing

- Be specific about what you'll teach and what students will learn
- Mention any tools, software, or materials needed
- Include your teaching experience or qualifications if relevant
- Consider what you'd want to know if you were the student

Share Skill Cancel



T

Tanishka chhajer

0 Skills Shared

0 Reviews

0 Badges

My Skills

Share New Skill

No skills shared yet

Share your first skill and start connecting with learners!

Share Your First Skill

Reviews (0)

No reviews yet

Sign Out

CONCLUSION

SkillSwap bridges the gap between learners and experts by creating a vibrant, community-driven skill exchange platform.
With its intuitive design, robust backend, and rich set of features—from skill listings to mentorship—SkillSwap encourages collaboration, continuous learning, and personal growth.
By leveraging Flutter and Firebase, the app delivers a seamless cross-platform experience, ensuring real-time interaction and scalability.
SkillSwap is not just an application—it's a step toward building stronger, more connected communities where knowledge is shared and everyone has the opportunity to learn and grow.