



PROGRAMMING FOR MOBILE DEVICES (CCS21503)

Received Date : 23 MAY 2025
Submission Due Date : 12-16TH May /17-18TH May 2025 (for BL student)
Lecturer : Dr. JAMAL ABDULLAHI NUH
Weightage : 30%
Semester : February 2025

Instruction to students:

- This is a GROUP assignment.
- Complete this cover sheet and attach it to your assignment (first page).

Student declaration:			
<p><i>I declare that:</i></p> <ul style="list-style-type: none">• <i>This assignment is my/our own work</i>• <i>I/we understand what is meant by plagiarism</i>• <i>My lecturer has the right to deduct my marks in case of:</i><ul style="list-style-type: none">- <i>Late submission</i>- <i>Any plagiarism found in my assignment.</i>			
NAME: KAVINESH A/L EASY SELVAN MATRICS NO: 012023071517	NAME: SHAMALAN A/L GDP NAVARATNA RAO MATRICS NO: 012023071189	NAME: AMR TALAT AYAYDAH MATRICS NO: 012023090852	NAME: SAMUEL A/L AARON MATRICS NO: 012023071711

1. Introduction

After successfully developing the front-end elements in Assignment 2, Assignment 3 of the SkillXchange – A Skill-Sharing Network mobile app deals with integrating front-end and back-end features as a whole. We were looking to develop user-friendly interfaces and multi-activity navigation using Android Studio and Java in Assignment 2. And now, with Assignment 3, we add real-time capability, secure user administration, and dynamic data manipulation to transform SkillXchange into a fully functional mobile application.

By developing the SkillXchange app, we aimed to create one place where members could swap their knowledge and discover new ones shared by others. The application lets people register, log into their accounts, add or update their profile, post what they do and check other members' skills. While the previous exercise displayed the appearance and structure of the app, the current assignment centers on integrating Firebase to handle sign-ins, store all data and immediately update users.

Our primary task in this project is to develop a complete, working, and scalable back-end system with seamless communication with the front-end. This includes Firebase SDK initialization in Android Studio, performing CRUD operations on skill posts and user profiles, session securing, and providing a responsive experience. The project also seeks to incorporate best practices in application modular design, data handling, and usability in creating a helpful and informative application. We also wished to complete and test all principal features to fulfill expectations of reliability, security, and user satisfaction.

SkillXchange is our way of using the knowledge from our courses in real situations. It supports individuals helping each other by allowing for skill discovery and personal development through the community experience. Finishing the app in this assignment highlights that we can design and build mobile apps, integrate Firebase and focus on users, three important skills in current software development. This project includes what we learned in the Mobile Application Development course and gives us the confidence to work on bigger mobile projects later.

2. Front-End Section

2.1 Tools Used

SkillXchange is a Skill Sharing Network that has been developed primarily with the use of Android Studio, as the official Integrated Development Environment (IDE) for Android App Development. Android Studio provides a strong and cognizable niche and helped us in managing our design, coding, and testing phases. The platform provided a user-friendly environment, lay-out editor with drag-and-drop features, and in-built development tools to give us a great UI experience, definitely responsive and interesting, in line with our goal to deliver a clean and seamless experience for all users.

Among the many advantages built into Android studio, Gradle-based build is one of the essential tools that helped us in field work while building the application. Gradle handles our app's structure and dependency management, enabling us to compile and build the project without a hitch. Then, Gradle also gives great freedom to modify the layout configurations in doing it for different screen sizes and devices without going into the core application logic. Hence, layout adaptability maintains usability and responsiveness on a level especially considering different Android device versions and, in these cases, meeting layout consistency and responsiveness standards.

Another major feature used was the Android Emulator, which allowed our testing to be performed across several virtual Android devices. This tool was paramount in spotting any layout inconsistencies and performance issues as early as in the development phase. It helped simulate real-time user interactions and allowed rapid iteration of the UI for improvement of navigation structure and screen transitions. Moreover, we ensured through the emulator that our interface was consistent and functional across various screen dimensions, contributing to smooth user interaction and navigation usability.

2.2 Design Decisions

SkillXchange – A Skill-Sharing Network was created with many reflective design choices for a more engaging and user-friendly experience. We decided to make a clean and simple platform for users to move around the application easily. The user can easily find key features in the app without being overwhelmed, whether browsing skill categories, posting a new skill, or connecting with another person. This approach actually puts the user first and helps reduce the learning curve for the first-time user, thereby promoting seamless interaction via the app.

On the other hand, the prime color palette is blue and white because it has been chosen to represent professionalism, trust, and clarity, much akin to the values espoused by collaborative learning and sharing. These colors provide a sort of clean visual canvas for better readability and enhance the polished appearance of the interface. For utility, we have put fifth support colors, such as neon purple and pink, white, blue, and black, which help consolidate user interaction flow. We primarily made the app background black, adding neon purple and pink lighting to it. The result is an interface that looks fantastic, draws attention from users and still has a professional appeal. White borders were added to dialog boxes and key user interface parts so they would be clearly visible. Blue, white and black are the main colors used for text and they help you notice any important messages without being too distracting. Using colors wisely makes the platform attractive and easy to use for all users.

Moreover, we then magnified the prominence of the visual aspect of the application's communication by interfacing with skill-relevant images and meaningful icons in various app sections. By this, it helps users identify skill types in a split second while acting as a lively, engaging addition to the Interface. We managed to keep design consistency with the use of the same font family for all text elements; the font choice promotes readability and aligns with a clean, modern aesthetics that is inviting and easy to the eyes.

Beyond the scheme of colors and pictorial aspects, SkillXchange – A Skill-Sharing Network is also incorporating the usability and functionality of the site. Another major applicant of minor aesthetics at the interface was trying to embed some simple and intuitive navigation patterns in smoothing users' journeys. Some of the app's key functions, like posting a skill, browsing skills, navigation and CRUD operations have buttons placed so they're quickly accessible. As a result, users could go right from opening the app to getting a simple introduction and using it more.

2.3 User Interface Components

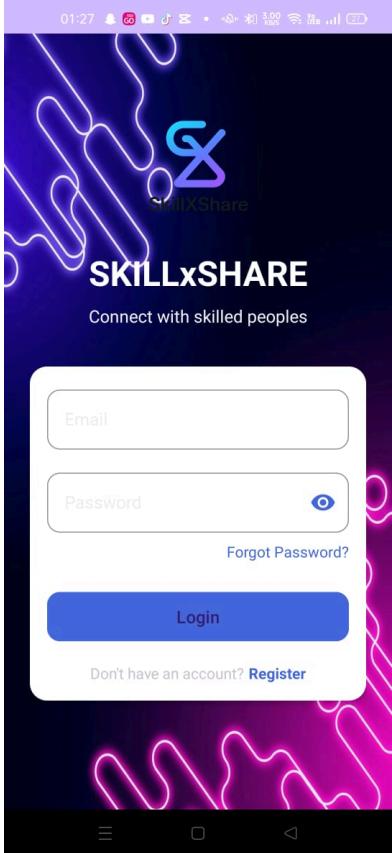
The SKILLxSHARE application exhibits a carefully crafted and colorful UI that highlights modern aesthetics with its dark background and neon-themed visualizations. It allows for a user-engaging experience, complementing the techie, very-futuristic feel of the application. The login and registration screens are clean and functional, with all essential fields, i.e., email, password, full name, address, and bio, displayed in friendly form elements. In addition, the login screen comes with the eye icon toggle for password visibility and a "Forgot Password?" link for recovery, thus increasing the usability. Below, a blue "Login" button and a "Register" link offer crystal-clear calls-to-action for user interaction so that new and returning users alike could comfortably use the platform.

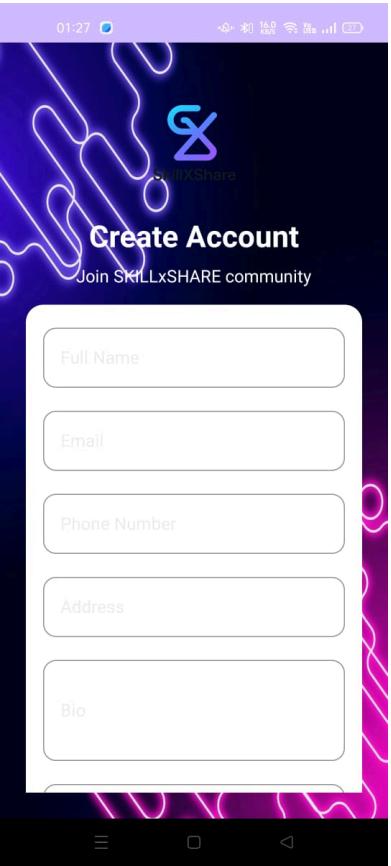
A key component of the user interface is navigation, and SKILLxSHARE has a bottom navigation bar with icons for the Home, Search, Profile, and About categories. This allows users to move between the main features of the application with ease and without having to look for answers. When it comes to displaying user data, such as name, email, mobile number, address, and bio, the Profile area is brilliant. Additionally, skills are displayed as tags that may be easily added or removed using the "+ ADD SKILL" button. The option "Edit Profile" and "Logout" buttons, which provide a great lot of control over account settings while maintaining a responsive and clean user interface, retain the app's interactivity friendliness.

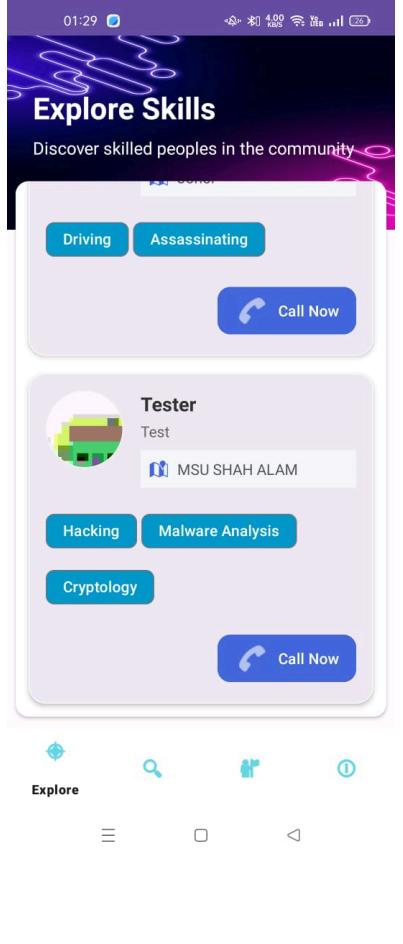
There exist the "Explore Skills" and "Find Skills" pages that create this core functionality of connecting the learner with the skills-givers. Both pages contain a search bar with filtering options ("By Skill" and "By Name"), along with trending skills listed in pill-shaped buttons and user profile cards showing people's names, locations, and skillsets. Each profile card features a "Call Now" button, enabling instant interaction, which means a very quick real-time connection between two individuals is possible.

Last but not least, the "About" page is informative and branding. It declares SKILLxSHARE's mission and vision, emphasizing knowledge sharing and community growth. A clear list of features, contact details, and credits to developers creates transparency and trust with the user. Consistent design language, dark background, neon highlights, and bold fonts ties the pages together, confirming the brand identity and navigation through the app instinctive and interactive.

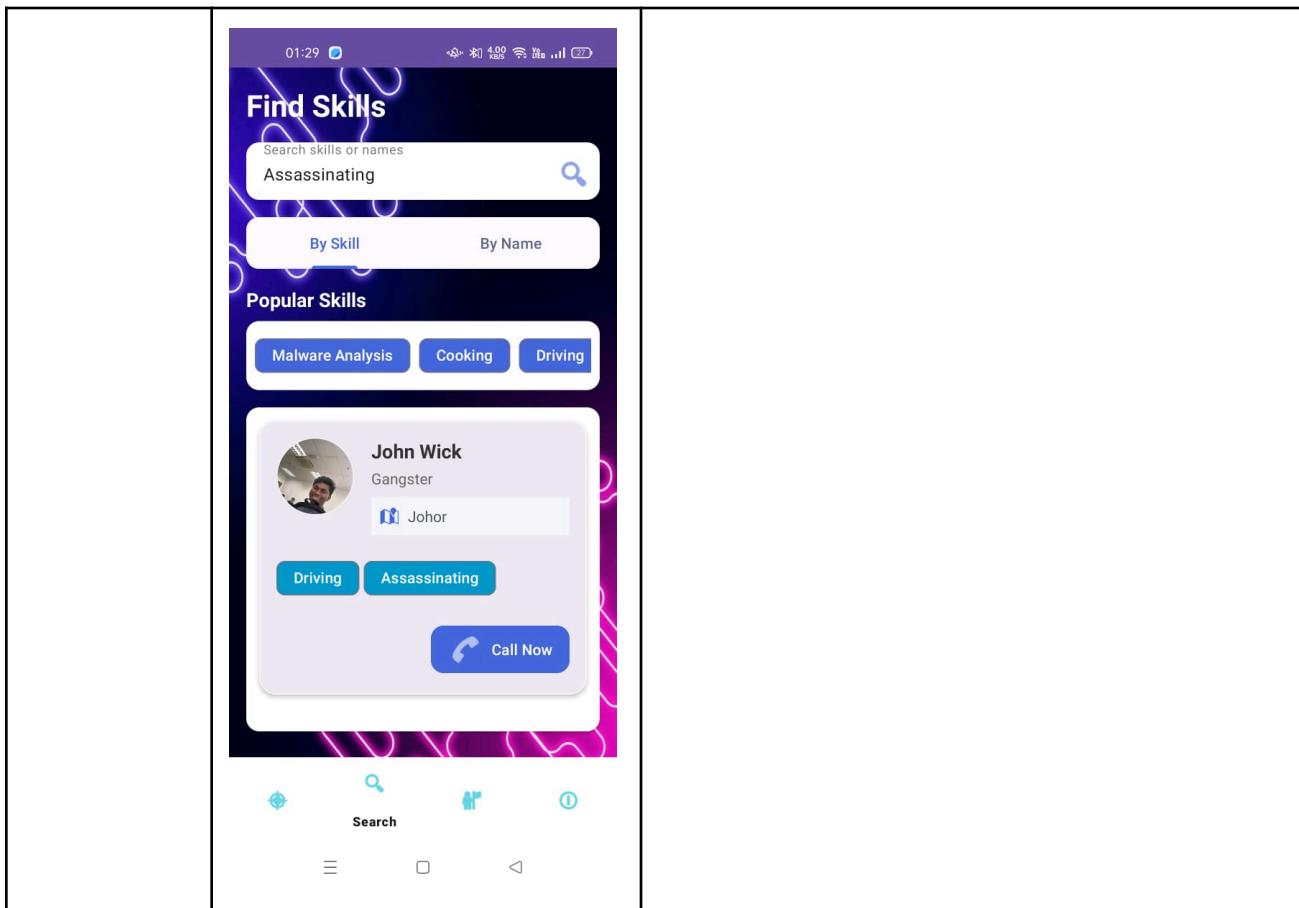
2.4 User Interface

Screen	Screenshots	Details
Login Page		<p>The login screen is shown in this image for the SKILLxSHARE mobile app created to link people with skill holders. The design of the screen is modern and vibrant, with a black background and neon graphics. At the top, the logo and name "SKILLxSHARE" are very distinct, with the tagline "Connect with skilled peoples" underneath (a grammatical error exists here - it should read "people"). Below it, users are asked to enter their email ID and password to log into the app, with the password field allowing one to toggle visibility via an eye icon, which is a good design touch. There is also a "Forgot Password?" link in case users wish to recover their password, with a blue "Login" button next to it that can be clicked to access one's account. At the bottom lies the "Register" link option for new users, all of which is easy to navigate and fun to look at, keeping the very tech-savvy theme of this app intact.</p>

Signup Page		The screen comes with an image of creating a "Create Account" screen for SKILLxSHARE mobile app wherein new users come to join the community. The design is consistent with the futuristic art style of the app; dark backgrounds with neon light effects and the SKILLxSHARE logo sit atop. Further, the text "Join SKILLxSHARE community" attempts to draw the user in below the header. Full Name, Email, Phone Number, Address, and Bio constitute the essentials in the form for user details; the fields are arranged to offer a nice clean layout, thus an easy and quick registration process. This screen allows new users to start using the platform and connecting with one another based on skills and interests.
--------------------	--	--

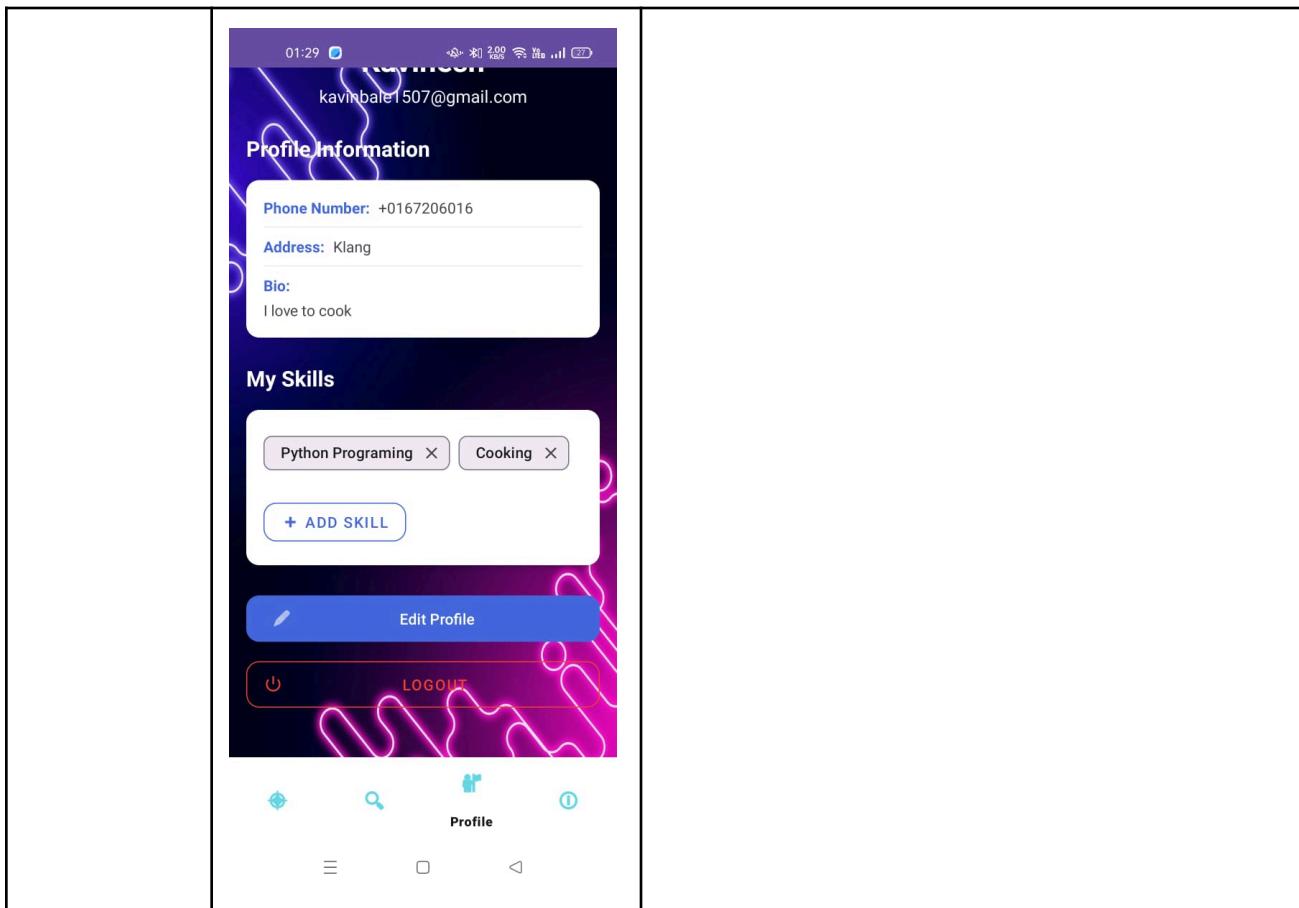
Explore Skills Page		<p>The screenshot is of the "Explore Skills" page of a mobile skill-sharing application, likely of the "SkillXange" project. The UI has a modern feel with a neon pink and purple background that facilitates a creative and interactive atmosphere. There is a header on top, "Explore Skills," which invites users to discover skilled individuals in the community. Below the header are user profile cards displaying a user's name, affiliation (e.g., "MSU SHAH ALAM"), and a list of blue skill tags like "Hacking," "Malware Analysis," and "Cryptology." A large blue "Call Now" button for immediate contact is available for easy user interaction. Prominent use of white in the UI is in favor of simplicity, and black text is easy to read. The layout and color options together reinforce an engaging and intuitive user experience.</p>
----------------------------	--	---

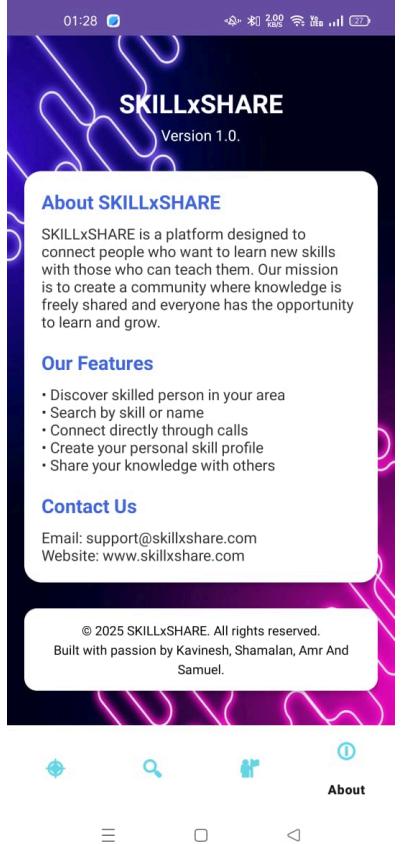
Find Skills Page		<p>The screenshot is from the "Find Skills" page of the mobile application, which is designed to enable users to search for skills offered in the community. The UI follows a neon theme with a purple and pink gradient background that enhances visual interaction. There is a bold title, "Find Skills," at the top and below it is the search bar where users type in skill titles or keywords. There are two filter buttons, "By Skill" and "By Name," below the search bar that allow users to filter their search results. Under "Popular Skills" are trending skills like "Malware Analysis," "Cooking," and "Driving," each in blue pill-shaped buttons.</p> <p>The second picture shows an app interface utilized to find individuals with some skills. Below is a user's profile, with "John Wick," whose occupation is listed as "Gangster" and country of origin is Johor. His skills are "Driving" and "Assassinating," and there is also a "Call Now" button for urgent calling. The layout is very basic with a black background and glow-like visual effects.</p>
-------------------------	--	--



**User Profile
Page**


The screenshot is of the "Profile" page of the user Kavinesh from the same mobile application. It contains the profile photo, name, and email (kavinbale1507@gmail.com) of the user at the top, with an edit profile option. Below that, the "Profile Information" field shows the user's phone number, address (Klang), and bio, "I love to cook." The "My Skills" field shows "Python Programming" and "Cooking" as Kavinesh's skills, with the option to add additional skills using the "+ ADD SKILL" button. The app follows a consistent dark and neon-based theme, with simple navigation icons at the bottom for different sections such as Home, Search, Profile, etc. This screen allows users to arrange and present their personal and professional information in an effective way within the app. Two important interactive buttons are also present on this page: a blue "Edit Profile" button to modify user details and a red "Logout" button to safely log out from the account. The overall user interface carries on the eye-catching neon-theme appearance with a dark backdrop, creating a modern, tech-savvy ambiance. The interface is user-friendly, with simple navigation to profile management and account control features.



About Us Page	 <p>The screenshot shows the "About" section of the SKILLxSHARE mobile application. At the top, it says "SKILLxSHARE Version 1.0." Below that is a section titled "About SKILLxSHARE" which states: "SKILLxSHARE is a platform designed to connect people who want to learn new skills with those who can teach them. Our mission is to create a community where knowledge is freely shared and everyone has the opportunity to learn and grow." Under "Our Features", there is a bulleted list: "Discover skilled person in your area", "Search by skill or name", "Connect directly through calls", "Create your personal skill profile", and "Share your knowledge with others". The "Contact Us" section provides an email address (support@skillxshare.com) and a website (www.skillxshare.com). At the bottom, it says "© 2025 SKILLxSHARE. All rights reserved. Built with passion by Kavinesh, Shamalan, Amr And Samuel." Navigation icons at the bottom include a diamond, a magnifying glass, a person icon, and an "About" icon.</p>	<p>The final photograph displays the "About" section of the SKILLxSHARE mobile app (Version 1.0). The page indicates a brief overview of the function and features of the app. SKILLxSHARE has been described as a platform that aims to connect individuals who want to learn new skills and individuals who are willing to share them with others. Its mission is to contribute towards developing a community where knowledge is exchanged without charge and anyone can learn and self-improve. Key things include finding experts near you, search by name or skill, connect by direct call, construct your own skills profile, and share with others. The department also provides contact details: an email address (support@skillxshare.com) and the site (www.skillxshare.com). Lastly, there is a credit note that the app was built with love by Kavinesh, Shamalan, Amr, and Samuel, with a nod to one-eyed eagle symbol one to the right for the honor of using their work: © 2025 SKILLxSHARE. The consistent neon and dark color scheme is what the rest of the app utilizes, and bottom navigation icons render users reachable between sections.</p>
----------------------	---	---

3. Back End Section

3.1 Back-End Architecture and Technologies Used

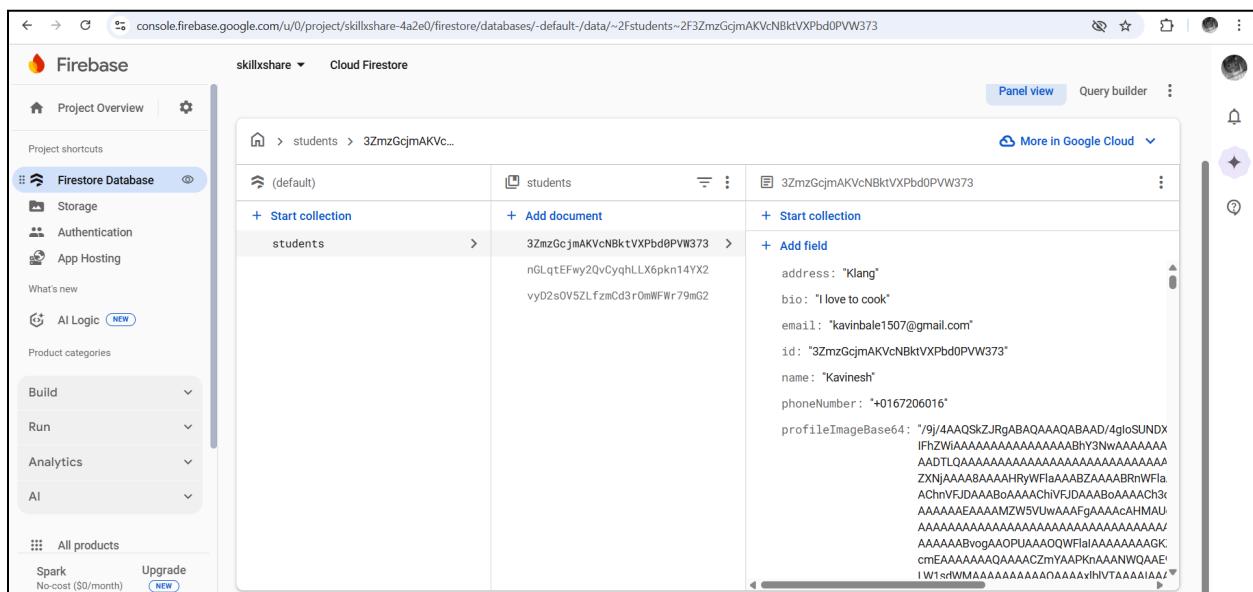
We used Firebase on the back end of our app, as it is a cloud-based Backend-as-a-Service (BaaS) platform. Using serverless technology, we did not have to worry about the servers or infrastructure of the backend.

- **Firebase Authentication:**

Authentication in Firebase ensures safe and easy management of user account registration, signing in and sessions. It guarantees that users can use the app without problems.

- **Cloud Firestore:**

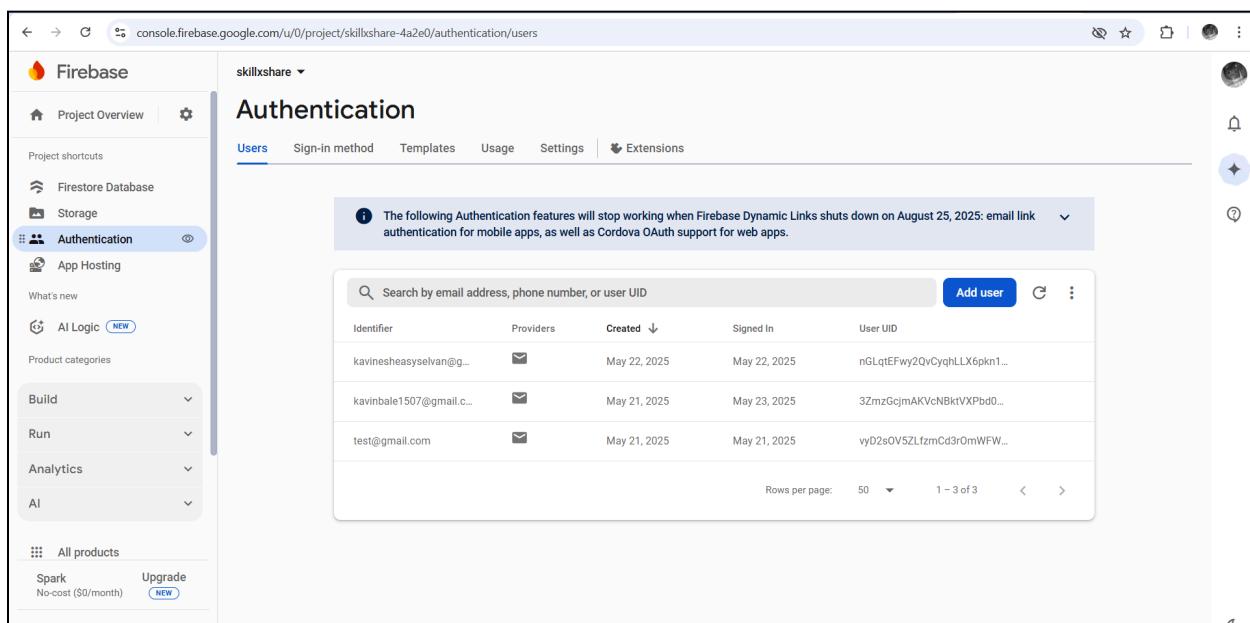
We rely on Cloud Firestore to keep and get user profile data. Here you'll find skills, contact details and a short bio about the user. This architecture allowed for a backend that is flexible, safe and can be maintained and it integrates nicely with the Android app.



The screenshot shows the Firebase Cloud Firestore interface. On the left, there's a sidebar with 'Project Overview' and several tabs: 'Firestore Database' (selected), 'Storage', 'Authentication', 'App Hosting', 'AI Logic', 'Build', 'Run', 'Analytics', and 'AI'. Below these are sections for 'All products', 'Spark' (No-cost (\$0/month)), and 'Upgrade (NEW)'. The main area shows a database structure under 'skillxshare' project. A collection named 'students' contains a document with the ID '3ZmzGcjmA...'. The document fields include:

```

address: "Klang"
bio: "love to cook"
email: "kavinbale1507@gmail.com"
id: "3ZmzGcjmA...VX...373"
name: "Kavinesh"
phoneNumber: "+0167206016"
profileImageBase64: "iVBORw0KGgoAAAQ... (long base64 string)"
  
```



The screenshot shows the Firebase Authentication console for a project named 'skillshare'. The left sidebar includes links for Project Overview, Firestore Database, Storage, Authentication (selected), App Hosting, AI Logic, Product categories, Build, Run, Analytics, and AI. The main area is titled 'Authentication' and has tabs for Users, Sign-in method, Templates, Usage, Settings, and Extensions. A message at the top states: 'The following Authentication features will stop working when Firebase Dynamic Links shuts down on August 25, 2025: email link authentication for mobile apps, as well as Cordova OAuth support for web apps.' Below this is a table listing three users:

Identifier	Providers	Created	Signed In	User UID
kavinesheasy selvan@g...	✉️	May 22, 2025	May 22, 2025	nGLqtEFwy20vCyghLLX6pkn1...
kavinbale1507@gmail.c...	✉️	May 21, 2025	May 23, 2025	3ZmzGcjmAkvCNbktVXPbd0...
test@gmail.com	✉️	May 21, 2025	May 21, 2025	vyD2sOV5ZLfmCd3rOmWFW...

At the bottom of the table, there are buttons for 'Rows per page' (set to 50), '1 - 3 of 3', and navigation arrows.

3.2 How the Front-End Communicates with the Back-End

Firebase's SDK for Android allowed us to build our app without depending on additional third-party servers or custom REST APIs for communication. We have applied the following in our implementation.

- **Authentication Flow:**

AuthService, within our application, takes responsibility for checking user credentials using Firebase and managing their active session. At app startup, the system looks at whether the user is authenticated, then leads to the login page or the home screen if needed.

- **Data Operations:**

Our Student models are turned into key-value maps by the `toMap()` method so they can be added to Firestore under the document IDs of each user. Data from Firestore documents is first parsed and then mapped into Student objects using the `fromMap()` method for proper and safe data usage.

- **Real-time Data Synchronization:**

Because Firestore uses real-time updates, the app automatically shows the latest user data to users, without making them refresh the app each time.

3.3 Detailed Data Flow

1. App Startup and Authentication Check:

After the application opens, Firebase services are automatically set up to establish the server environment. Then, the app uses the current user's input to check their authentication in AuthService. Should a valid session be detected, the user will be taken immediately to the dashboard. If that's not the case, the app goes to the login screen to request that the user provide their credentials.

2. User Authentication:

Those who use the app are asked to register or log in using Firebase Authentication. It stores and checks user information, verifies access with a token and preserves secure sessions during all app use.

3. Profile Creation and Updates:

As soon as authentication is successful, the user's data is managed in Cloud Firestore. A key-value map of the Student model is produced using toMap() and then saved to the user's Firestore account under their own unique ID. Personal information, bio, skills and a profile image encoded as Base64 are included in a profile.

4. Skills Management:

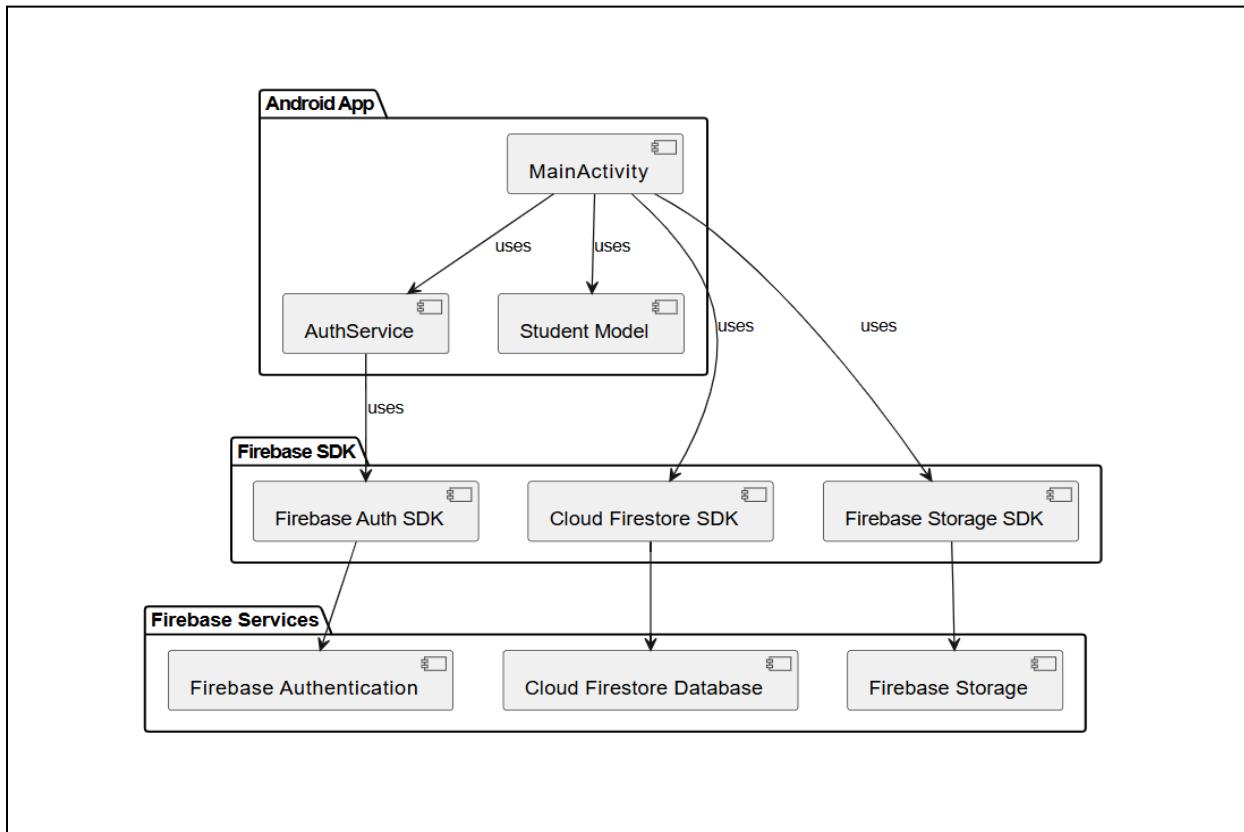
Individual user skills are organized as strings in the skills field under the Firestore document. Whenever a user adds or removes skills via the app, the local Student object is updated and the new information is sent to Firestore right away.

5. Data Retrieval and UI Population:

When either the profile or dashboard screens show up, the app fetches the user's Firestore document. The information from the raw document is put back into a Student object using the

fromMap() method. The object is used to fill the UI with relevant information, so the users can view their skills and profile at any moment.

Component	Technology / Tool	Role / Purpose
Authentication	Firebase Authentication SDK	Secure user login, registration, and session management
Database	Cloud Firestore	Storage of user profiles, skills, and app data
Front-End Communication	Firebase SDK	Direct client-side communication with Firebase services



The diagram clarifies how the Android part of the app works with the Firebase section. MainActivity within Android App handles the user screen and turns over authentication work to AuthService which in turn uses the Firebase Authentication SDK to ensure login security and

session details. The Student model stores user information and makes use of the Cloud Firestore SDK to keep and get that information. The Firebase Storage SDK allows users to upload and handle profile photos. Every Firebase SDK talks directly to its Firebase cloud service which allows for constant data updates and no need for server maintenance. Because of this design, SkillShare's app front-end and back-end link smoothly, making the user experience consistent throughout the app.

4. Challenges And Solutions

Challenge 1: Backend Integration Difficulties

The integration of backend services emerged as a primary technical hurdle we faced when implementing the call button interactive functions. Our connection to Firebase operates properly for authentication and data storage but problems emerged when we attempted to add real-time features requiring external intents or permissions (like voice or video calls). After studying the Android documentation and consulting multiple online references, the call button does not execute its expected action.

Solution :

Through our previous experience with Firebase and asynchronous operations during the Final Year Project (FYP) we formed a solid base for working with these technologies. We used our past exposure to detect problems and solve other backend issues more quickly. The call feature's complex nature and our time constraint prevented us from achieving complete functionality. We will return to this feature for post-submission improvements in upcoming versions of the application.

Challenge 2: OneDrive-Related Build Issues

We encountered a crucial problem with OneDrive service during our project execution. Our project suffered major operational breakdowns because it relied on a shared OneDrive folder during the final build stage. The development process stopped the APK file release for our application because Android Studio failed to discover assets which matched OneDrive files or were incorrectly linked to the service. The submission process faced high levels of pressure since the release procedure stopped multiple times.

Solution :

We took steps to protect our files by relocating key resources to offline storage before reengineering our system to stop depending on cloud-based synchronization of folders. The process of relocating to a completely local development setup enabled us to build a functional version yet caused significant delays which affected our deployment schedule. From this

experience we understand the importance of refraining from using cloud-synced directories including OneDrive when maintaining local path congruence during development operations.

Challenge 3: Limited Development Time

The academic semester brought forth a substantial challenge when students attempted to handle multiple responsibilities. Our team had to split time between developing SkillXShare and finishing the Final Year Project (FYP) which needed full dedication of time and effort. The team faced high pressure because of tight deadlines and concurrent deliverables which created limitations for in-depth testing and feature debugging and enhancement.

Solution :

Dr. Jamal's timely intervention through a project extension helped us effectively handle our workload. The extra time enabled us to concentrate on fixing unresolved issues, improve the design and confirm proper operation of essential features. Through the extended period we managed to release a more polished and optimized version of SkillXShare.

Challenge 4: Front-End Redesign Requirement

We recognized that our first frontend design became excessively complicated and unintuitive when we started developing the project. Our navigation structure together with the layout design made feature integration difficult while causing user experience problems during testing. The system design made it challenging to use components because the styling did not match the rest of the application interface thus affecting devices compatibility.

Solution :

After identifying these issues we concluded that the front-end interface required a complete redesign. Our team implemented a layout simplification together with color contrast enhancements and a more user-friendly structure that delivered faster development and improved user experience. The redesigned front end enhanced the app appearance through neon-themed branding while maintaining user accessibility.

5. Conclusion

As a result, working on the SkillXchange – A Skill-Sharing Network app has given us a thorough and valuable education in learning all aspects of mobile app development. We managed to use both front-end and back-end approaches with Android Studio and Firebase, connecting design and real implementation in the project. We found out how to create secure user processes, store data in real-time and adjust our user interface depending on user interaction by merging Firebase Authentication and Cloud Firestore. We became familiar with working with delayed operations, organizing data and running several activities and fragments on Android.

This project deepened our understanding of important development practices such as modular coding, separation of concerns, and responsive design. Moreover, it enhanced our ability to collaborate effectively as a team, especially in debugging issues, resolving challenges such as Firebase connectivity and UI redesign, and completing deliverables within tight deadlines. We also developed better problem-solving and time management skills, especially when juggling this project alongside our Final Year Project (FYP).

Looking ahead, there is significant potential to further improve the SkillXchange app. Future improvements could include implementing chat or messaging functionality between users, adding a rating or feedback system for posted skills, supporting media uploads for skill demonstrations, and enhancing the user interface for even better accessibility and visual appeal. Security measures can also be expanded to include multi-factor authentication, and performance optimizations can be made as the app scales. Ultimately, SkillXchange lays the foundation for a robust, community-driven platform, and this assignment has equipped us with the practical knowledge and confidence to build even more advanced and impactful applications in the future

Github Link

<https://github.com/Kavinesh1507/SkillxchangePMD.git>

References

1. Coding With T. (2020, March 14). 04(A) - Modern Dashboard UI Design Android Studio Tutorial - Cardview in Android Studio. YouTube.
<https://www.youtube.com/watch?v=NkPpxGh7YM>
2. User authentication using Firebase in Android. (2019, November 7). GeeksforGeeks.
<https://www.geeksforgeeks.org/user-authentication-using-firebase-in-android/>
3. Wellitsabhi. (2024). Skillswap/backend at main · Wellitsabhi/Skillswap. GitHub.
<https://github.com/Wellitsabhi/Skillswap/tree/main/backend>
4. Brown, George. 8 Aug. 2014, "Android Go to next Page." Stack Overflow,
[stack overflow.com/questions/25203386/android-go-to-next-page](https://stackoverflow.com/questions/25203386/android-go-to-next-page)
5. (2025). Youtu.be. <https://youtu.be/DWIGAkYkpg8?si=9XNSTFAonlsf9foV>

Rubric for Assignment 3: Back-End Programming & Mobile Interface Development

Criteria	1-2 (Standards Not Met)	3-4 (Standards Barely Met)	5-6 (Meets Standards)	7-8 (Exceeds Standards)	9-10 (Outstanding)	Marks
Introduction (3)	Lacks clarity or relevance to the project goals.	Objectives vaguely mentioned with minimal project context.	Provides an adequate overview with some detail on goals.	Clear objectives, covering front and back-end development in context.	Highly detailed introduction with clear project goals and outcomes.	
Front-End Design (5)	UI design is incomplete, poorly structured, and not functional.	Minimal design with little attention to usability and consistency.	Functional UI, but lacks visual appeal or consistency.	Well-designed UI, user-friendly, responsive, and meets project requirements.	Exceptionally designed UI with excellent usability, responsiveness, and consistency.	
Back-End Functionality (10)	Basic or incomplete back-end with little or no CRUD functionality.	Some back-end functionality is present, but incomplete or faulty CRUD operations.	Functional back-end with all CRUD operations, but minor issues exist.	Full back-end implementation with working CRUD operations and minor issues.	Outstanding back-end with well-structured, flawless CRUD operations and additional features (e.g., user authentication).	
Data Handling & CRUD Operations (20)	No clear data management, incomplete CRUD methods, or poorly functioning.	CRUD methods implemented but lack optimization or clarity in handling data.	CRUD functions are correctly implemented but need improvements in structure or efficiency.	Clear and effective implementation of CRUD methods, ensuring efficient data handling.	Highly organized, efficient CRUD functions with clear separation of concerns, optimal data handling.	
Authentication System (5)	No authentication implemented or non-functional.	Basic authentication exists but has security issues or lacks key features.	Authentication implemented but may have minor issues (e.g., insecure, limited error handling).	Fully functional user authentication (signup, login) with minor flaws.	Secure and well-implemented authentication system with all required features and proper error handling.	

Back-End Integration (10)	No proper integration between the front and back end, or only partially working.	Integration is present but lacks efficiency or consistency.	Basic integration between the front and back end with some functionality issues.	Seamless integration of front-end and back-end with good functionality.	Excellent integration, with smooth communication between front-end and back-end, using best practices.	
Report Structure & Clarity (5)	Report is disorganized, lacks details, and is hard to follow.	Report is partially structured but lacks clarity in key sections.	Adequately structured report, covering key sections but lacking in-depth analysis.	Well-structured and clear report, covering all required sections with appropriate depth.	Exceptionally clear, well-organized, and in-depth report, covering all sections with comprehensive detail.	
Screenshots & Documentation (5)	No screenshots or poorly documented.	Screenshots are present but poorly explained.	Screenshots included with explanations but lacking detail.	Clear screenshots with explanations, well-documented process.	Detailed documentation with relevant, well-explained screenshots throughout.	
Challenges & Solutions (5)	No challenges mentioned, or no solutions provided.	Challenges mentioned but solutions are unclear or ineffective.	Some challenges described with adequate solutions.	Challenges well-documented with clear and practical solutions.	Comprehensive breakdown of challenges with innovative and effective solutions.	
Prototype Submission (2)	Project files are incomplete or not functional.	Project files are submitted but contain errors or lack functionality.	Project files are functional with some minor issues.	Complete project files, fully functional and meeting requirements.	All project files submitted are well-organized, fully functional, and exceed requirements.	
Individual Evaluation (30%)						
1) Task Completion 2) Quality of Work 3) Able to explain clearly 4) Problem-Solving & Technical skills 5) Explain things using Android/programming/OOP/Mobile terms						

Total Marks: 100 (Weight: 30%)

Member	Score
KAVINESH A/L EASY SELVAN	
SHAMALAN A/L GDP NAVARATNA RAO	
AMR TALAT AYAYDAH	
SAMUEL A/L AARON	