Khulna University of Engineering & Technology

Report on:

Brikkhayon

Course No: CSE 3200

Course Title: System Development Project

Submitted by

1. Saurav Roy
   Roll:1807084

2. Dabbrata Das
   Roll: 1807109

Department of Computer Science and Engineering

Khulna University of Engineering & Technology

Supervised by

Nazia Jahan Khan Chowdhury

Assistant Professor

Department of Computer science and Engineering

Khulna University of Engineering & Technology

Department of Computer Science and Engineering

Khulna University of Engineering & Technology

Contents

1. Objectives

- To develop and build a cross platform application using dart programming language in Flutter UI Toolkit which maintains a tree-based structure.

- To develop the website using No-SQL Firebase database to get the facilities of online data storage and perform creation, retrieve, update and storage operation.

- To develop the application with Artificial Intelligence based features in some fields like chatbot, voice recognition searching.

- To implement location-based searching for finding plants depending on the range of the areas from current location.

- To develop a platform in which all the users can get their desired plants easily and can be familiar with all kinds of plants.

- To develop admin panel from where the admin can monitor and take actions against users if they perform any problematic or illegal activities.

2. Introduction

2.1. Brikkhayon

Brikkhayon is an e-commerce application which focuses only on the plants sector. Here users can buy or sell their desired plants. Their data is stored in the Firebase database. Firebase authentication is used to log in, register, verify and sign out users. It is a feature enriched application which is developed keeping user comfort in mind. It is a very easy to use application for all forms of users.

2.1.1. Functionalities

- User authentication
- User verification
- Online data storage
- Category based sorting
- Location based searching
- Voice-recognition based searching
- Search suggestions
- Admin panel and authentication
- Basic database operations
- Basic e-commerce functionalities

2.1.1.1. Location based searching

Location based searching is implemented using *google_maps_flutter*, *geolocator* and *geocoding* dependency. Geocoding is used to derive the current location of the user. Then the distance of the user and the plants are measured using a function. The user can filter the distance. If the distance to a plant is less than the input distance, then the plant is added to a list and hence showed in a list view.

2.1.1.2. Voice Based Searching

Google *speech-to-text* flutter package is used to convert text to speech. Then the plants having name containing that converted string as suffix or prefix are showed to the user.

2.1.1.3. Search suggestions

flutter_typeahead dependency is used to implement this function. It provides a suggestion popup when any input string is provided.

2.1.1.4. Authentication using various providers

Gmail, Facebook and Twitter authentication are implemented by applying various dependencies. Authentication state is also handled. So, a user doesn't need to log in every time if he wants to use the app frequently.

2.2. Flutter

Flutter is an open-source UI software development kit created by Google. It is used to develop cross-platform applications for Android, iOS, Linux, macOS, Windows, Google Fuchsia, and the web from a single codebase. Flutter is an open-source framework by Google for building beautiful, natively compiled, multi-platform applications from a single codebase.

Flutter offers many ready-to-use widgets (UI) to create a modern application. These widgets are optimized for mobile environments and designing the application using widgets is as simple as designing HTML.

2.2.1. Advantages of Flutter

- Same UI and Business Logic in All Platforms
- Reduced Code Development Time
- Increased Time-to-Market Speed

- Similar to Native App Performance
- Custom, Animated UI of Any Complexity Available
- Own Rendering Engine
- Simple Platform-Specific Logic Implementation
- The Potential Ability to Go Beyond Mobile

2.2.2. Disadvantages of Flutter

- The apps made with Flutter tend to be weighty ones.
- While Flutter is popular, it has not been around long enough to have a huge resource base. Therefore, your team will need to write a lot of stuff from scratch.
- Dart is not a popular language and if you want to work with Flutter you will have to learn how to use it.

2.3. Firebase

Firebase is a set of hosting services for any type of application (Android, iOS, Javascript, Node.js, Java, Unity, PHP, C++ ...). It offers NoSQL and real-time hosting of databases, content, social authentication (Google, Facebook, Twitter and Github), and notifications, or services, such as a real-time communication server.

It is an app development platform that helps to build and grow apps and games users love. Backed by Google and trusted by millions of businesses around the world.

2.3.1. Advantages of Firebase

- Authentication libraries
- Realtime Database and Firestore Database

- Google analytics
- Cloud storage
- Cloud Messaging for Cross-Platform Apps
- Swift and Secured Web Hosting
- Higher Accessibility to Machine Learning APIs
- Optimized App Performance
- Push Notifications for Target Messaging

2.3.2. Disadvantages of Firebase

- Firebase users are locked in on the platform
- The free plan offers only basic features that lack the advanced functions that simplify and expedites all development tasks. For many, the self-hosting option is more cost-efficient.
- Firebase service is blocked in China and other countries that block Google services.
- Do Not Support SQL Database
- Document size limited to 1MB
- Lack of native aggregation of query
- Slow queries undermine database performance.
- The query pattern is not very flexible.

3. Procedure

The project is developed by using flutter UI Toolkit with dart programming language. Here there are some linear steps which are implemented to the project.

- After clicking the icon of the project, it will show animated splash screen and redirect to another screen from where we can sign in and signup of the project.
- If an account is not registered, we must register before login and we should verify the email address for registration to check email's validation.
- After signing in, home screen will appear with all the plants from all users and we can get to know more about the plants simply by clicking the plants and redirecting to the product detail page.
- In the product detail page, a buyer can find products, owner email address and the location of the plants through google map by which the buyer can communicate with the owner of the plant and can trace the plant addresses.
- There is an add to cart button to add the plant before buying and a user can select multiple plants from cart page. Then from cart page, a user can finally place the order.
- In the application every user has their own profile. From there, they can upload their plants and can modify them as well.
- In settings page, users can reset their password, user name and can see their uploaded plants and logout also.
- The user is provided with four categories of plant choices which must be selected before uploading a plant and they are used as keys to view the plants categorically.
- There is an admin panel in the application from where an administrator can monitor all the users and their uploaded plants and can remove any user or plants with respect to illegal action.
- The main focus of the project is handling "search" functionalities in different ways. Those are voice recognition, location-based searching and searcj suggestions.
- *dialogflow* API is used to get chatbot facilities and the reply messages can be dynamically trained from here.

- In the project, all the data are fetched from Firebase Firestore and Realtime database and can be easily managed because of being NoSQL database.

So, by doing all of the procedures and implementations the application might be run successfully without any issues.

4. Flow Chart

The flowchart provides a complete visualization of the application page and activity transitions.

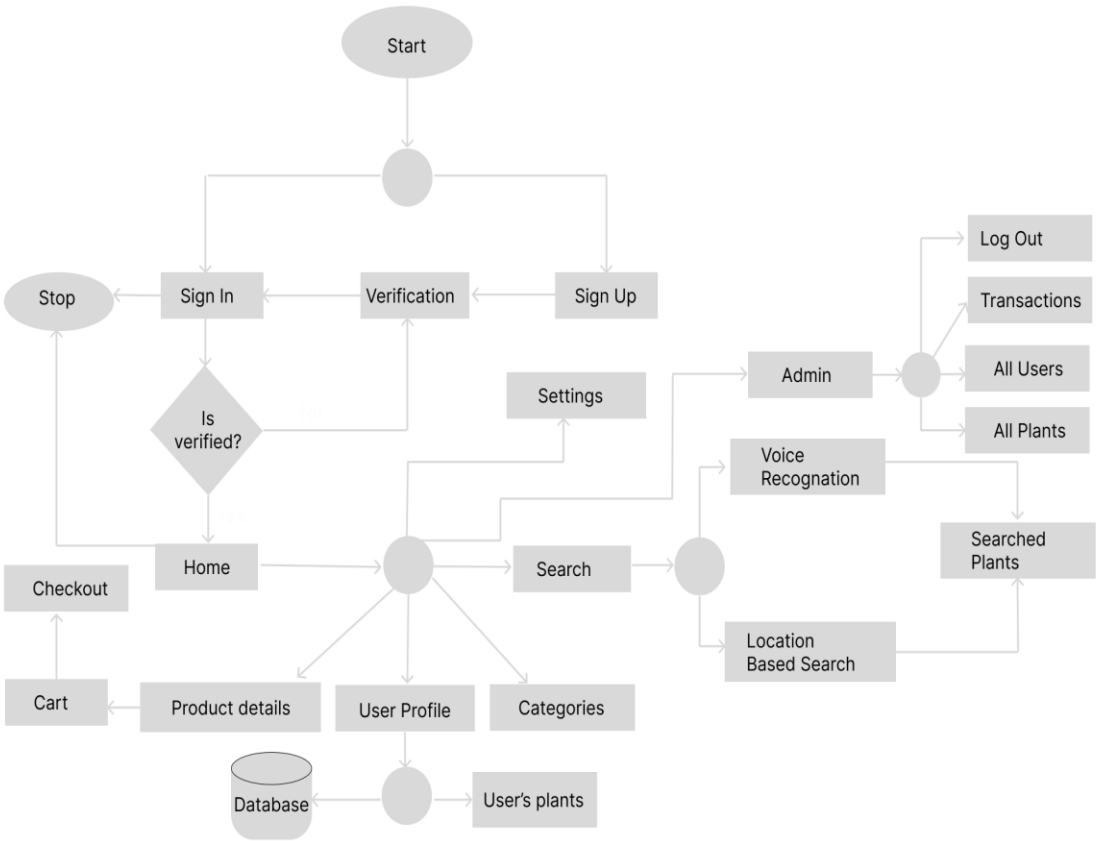

Fig 4.1: Flow chart of Brikkhayon application

5. Schema Diagram

The design of the database is called schema and more formally it means the structural view of the database. A schema diagram only shows us the database design. It does not show the actual data of the database. Schema can be a single table or it can have more than one table which is related. The schema represents the relationship between these tables.

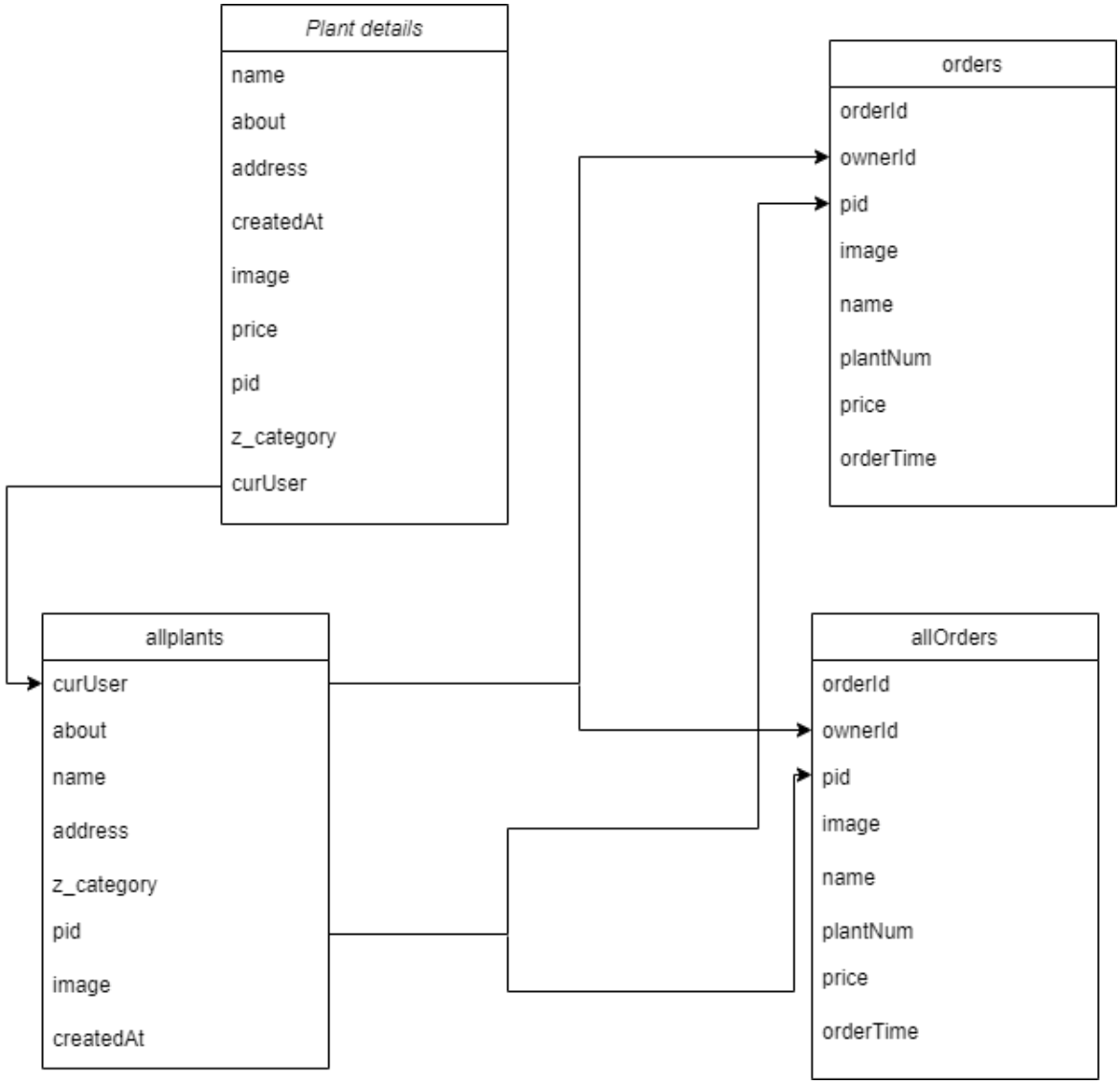

Fig 5.1: Schema diagram of Brikkhayon application

6. Detailed Visualization

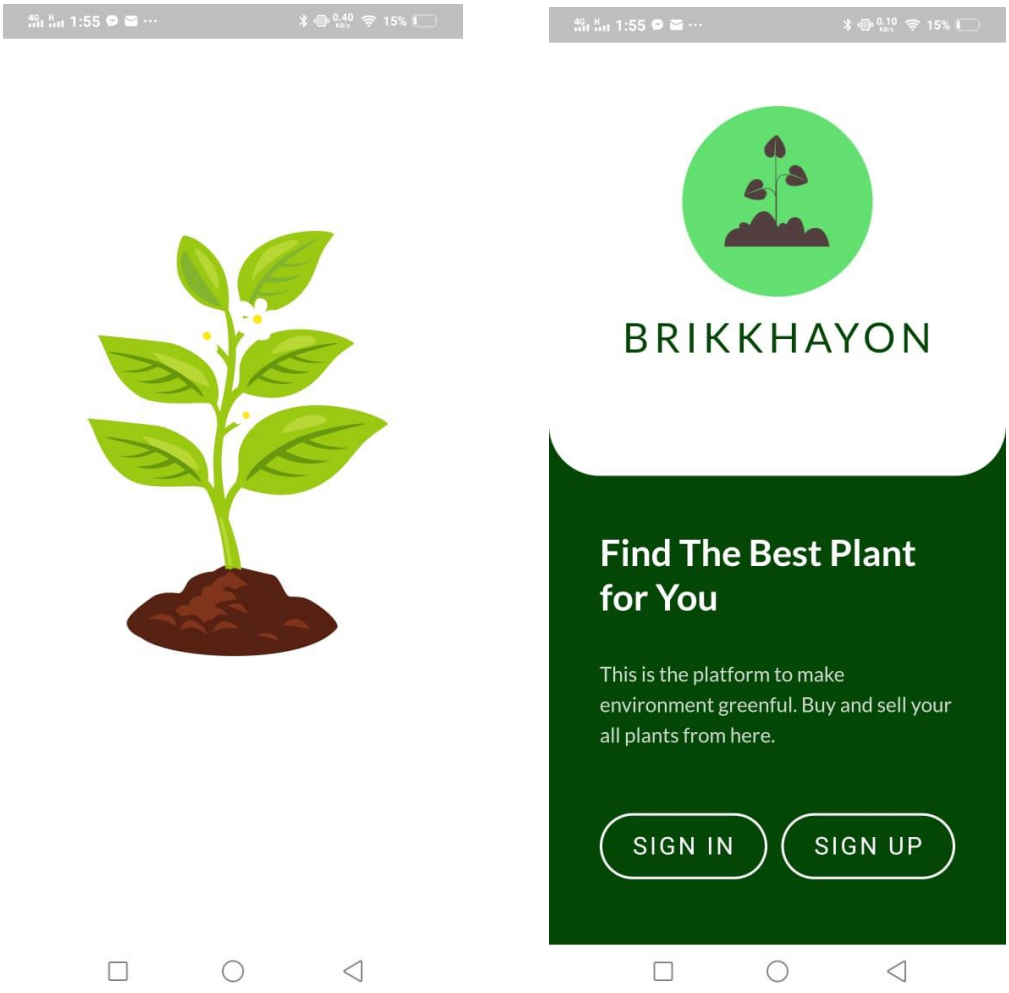

Fig 6.1: Animated splash screen

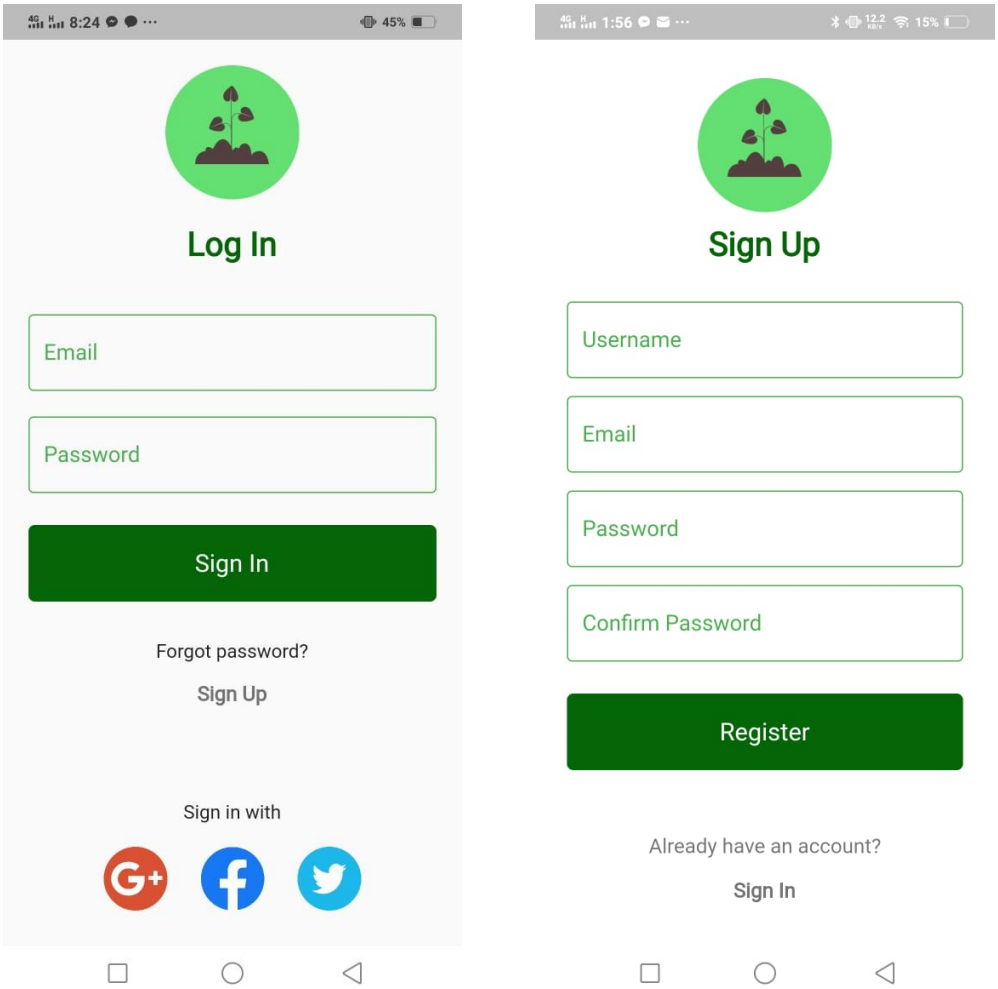

Fig 6.2: Login and registration

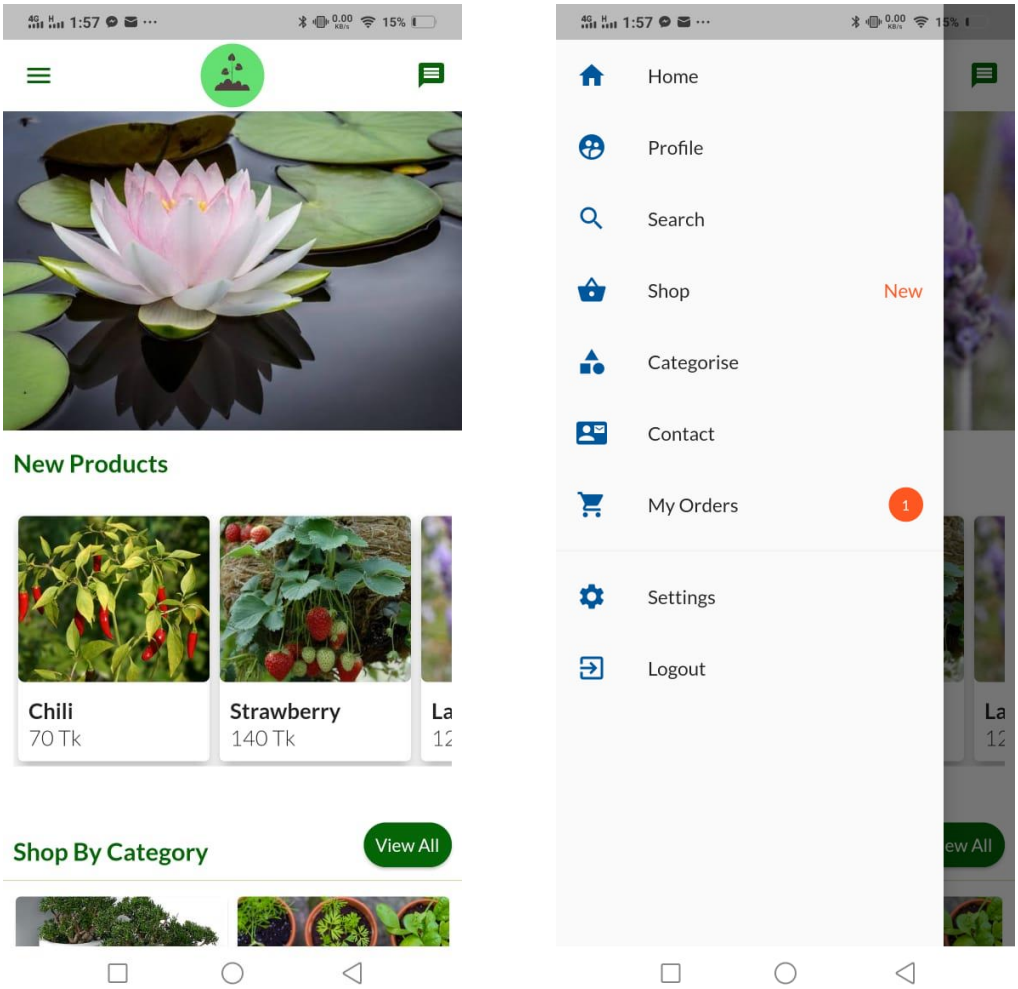

Fig 6.3: Home screen with app drawer

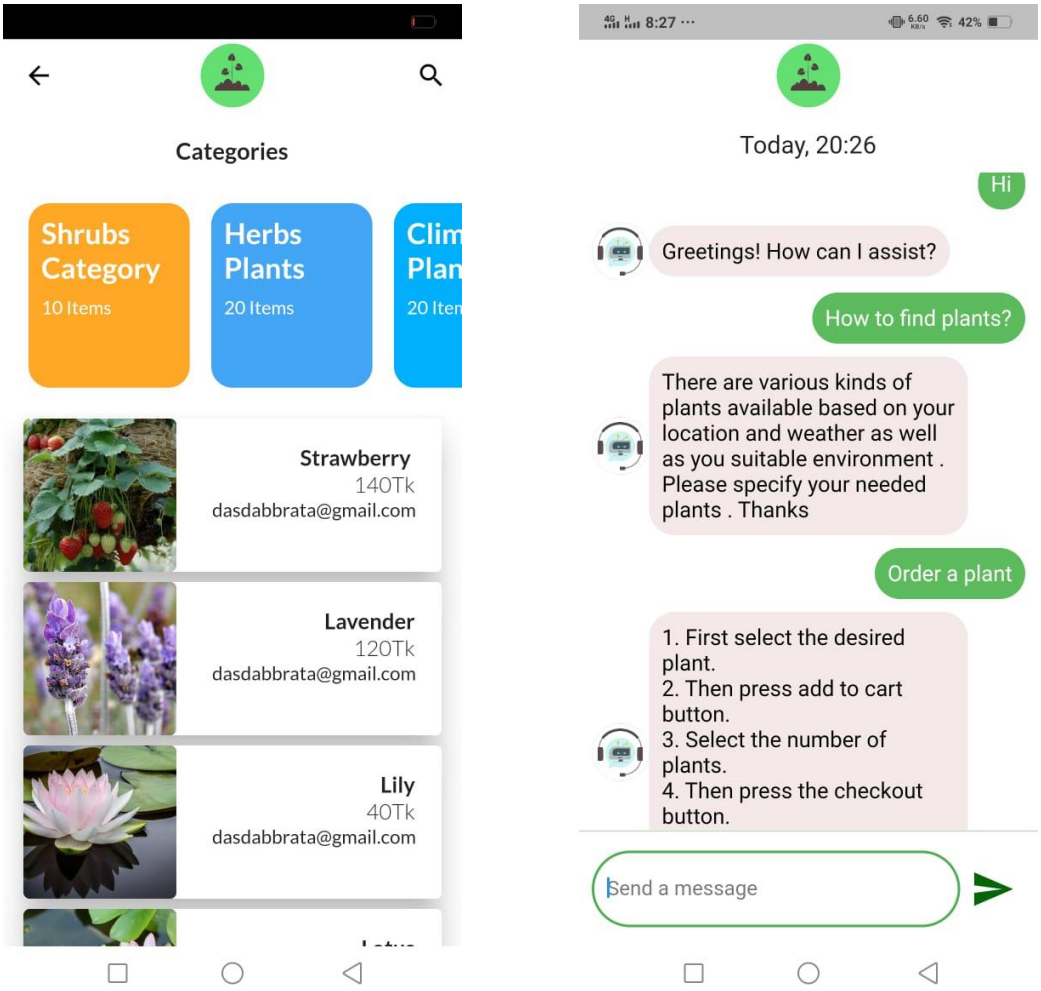

Fig 6.4: Categories and Chatbot

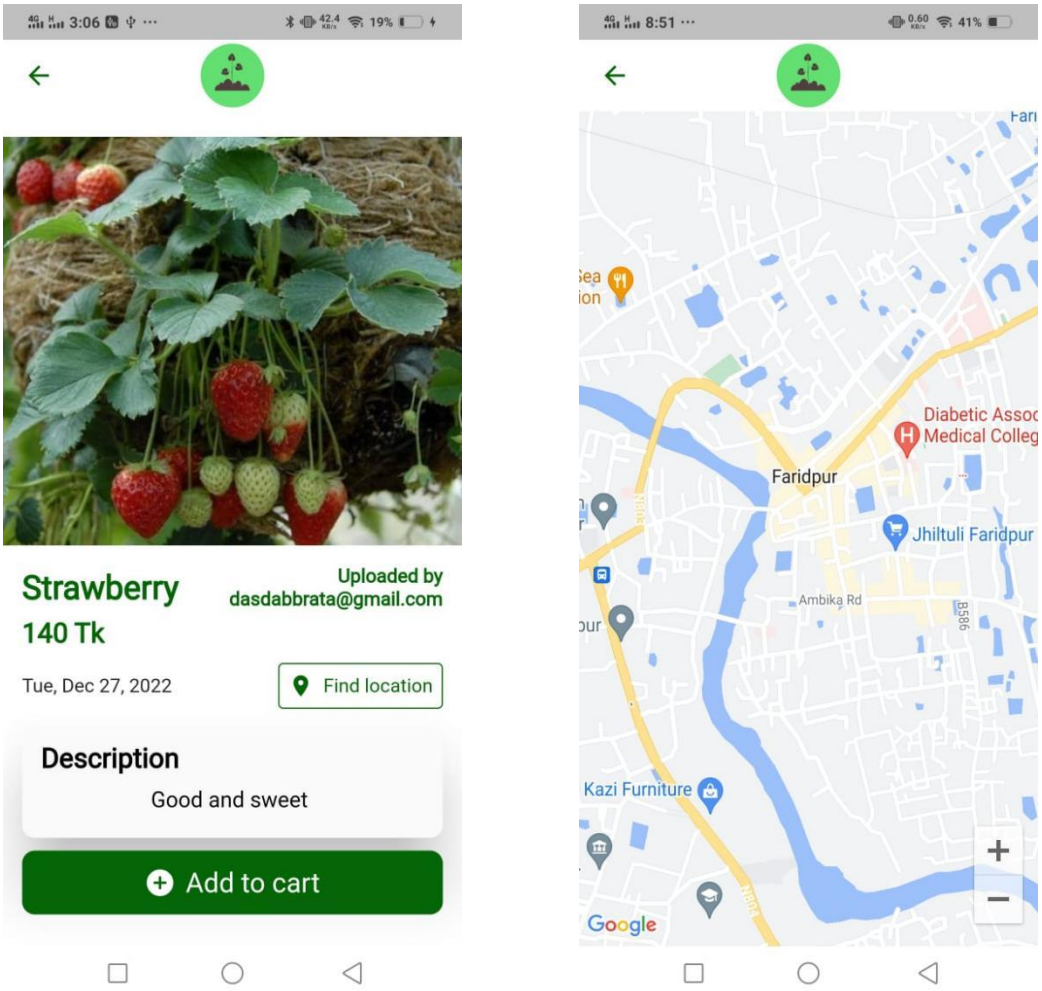

Fig 6.5: Product details and plant (product) location

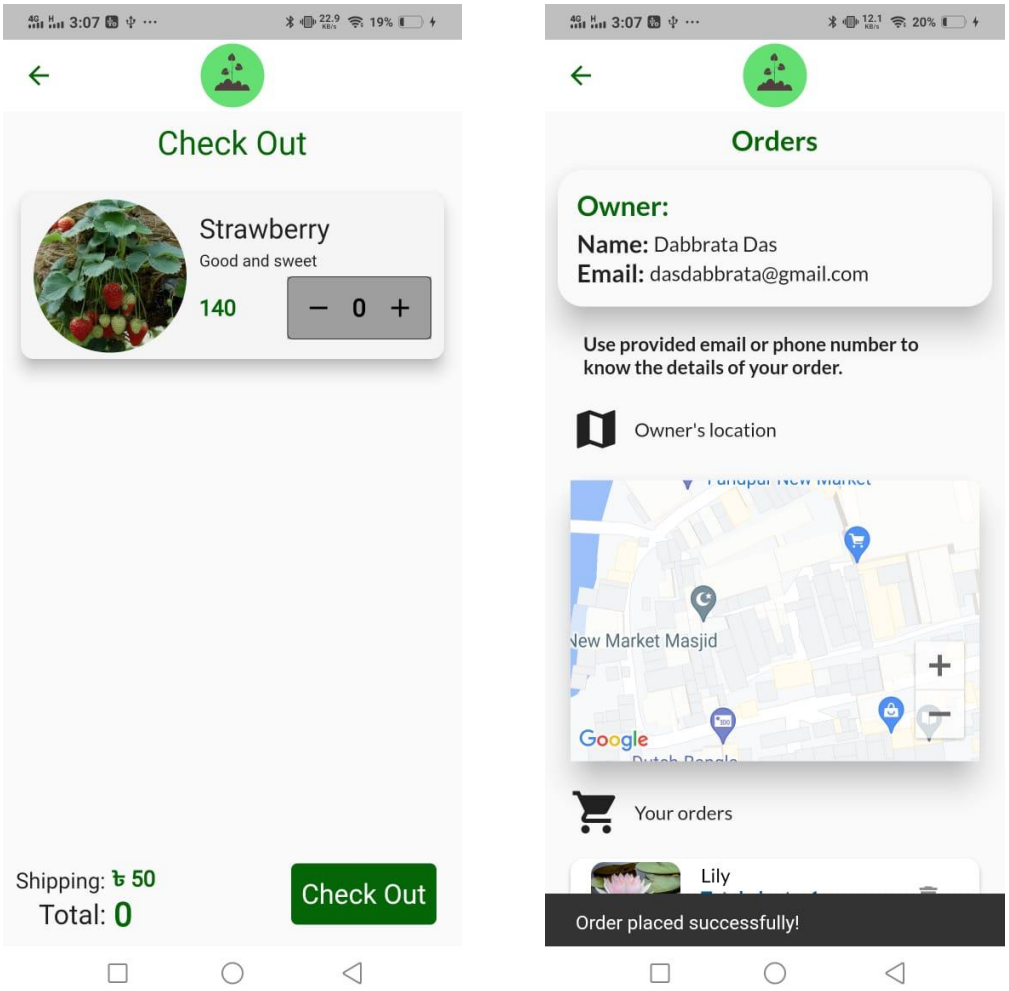

Fig 6.6: Add to cart and place order

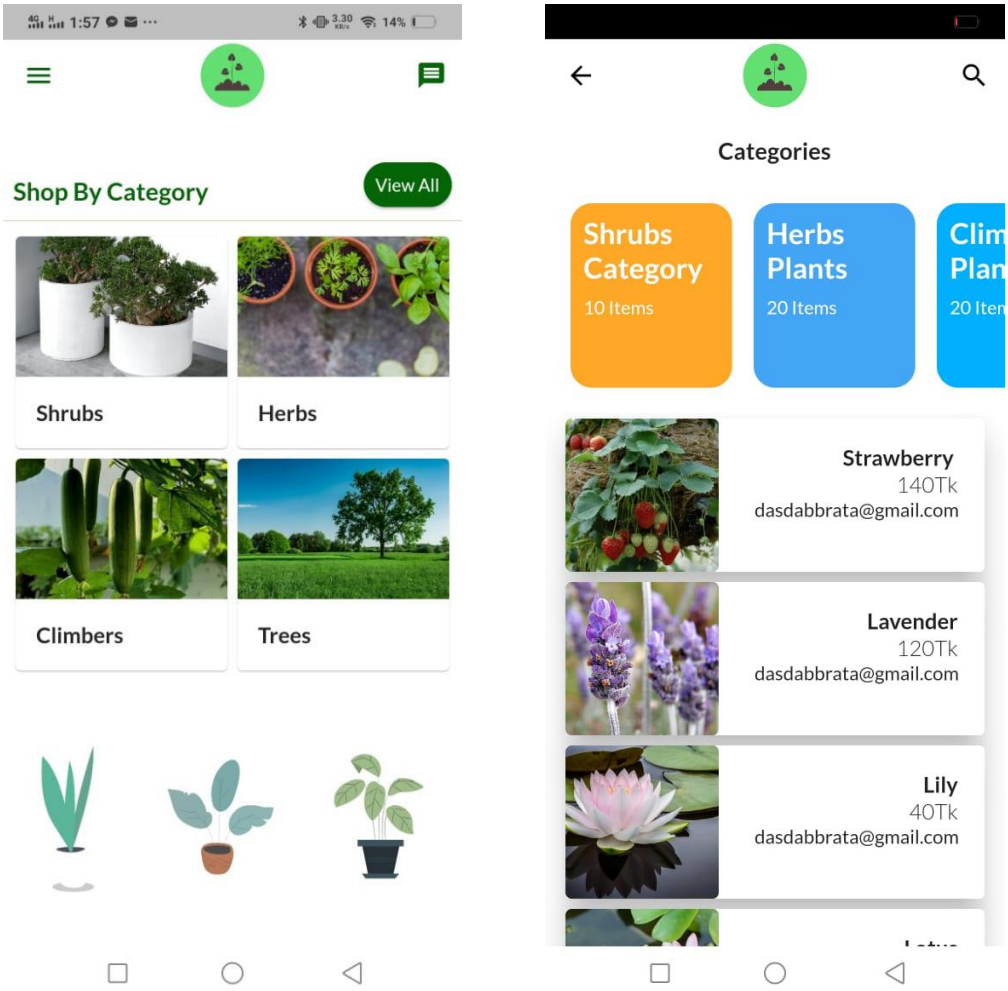

Fig 6.7: Different types of categories

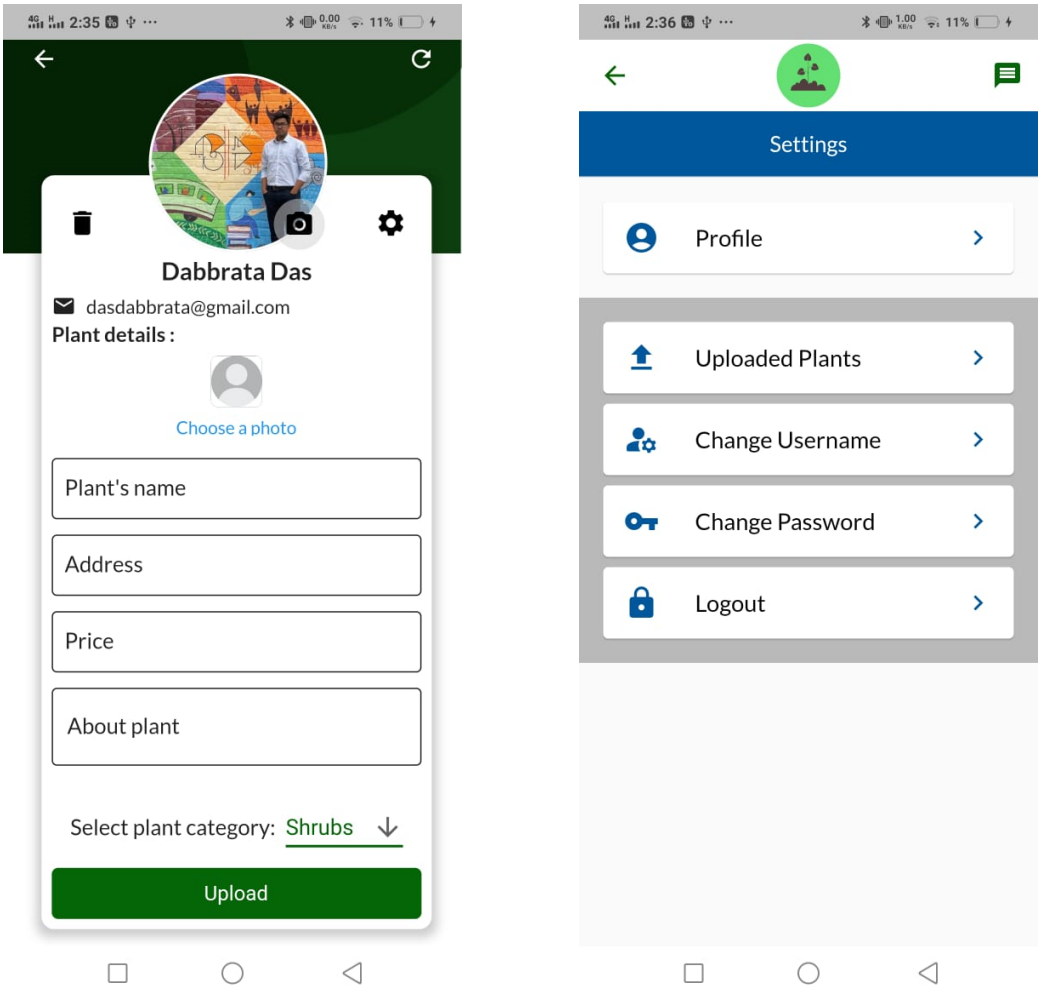

Fig 6.8: User profile and settings

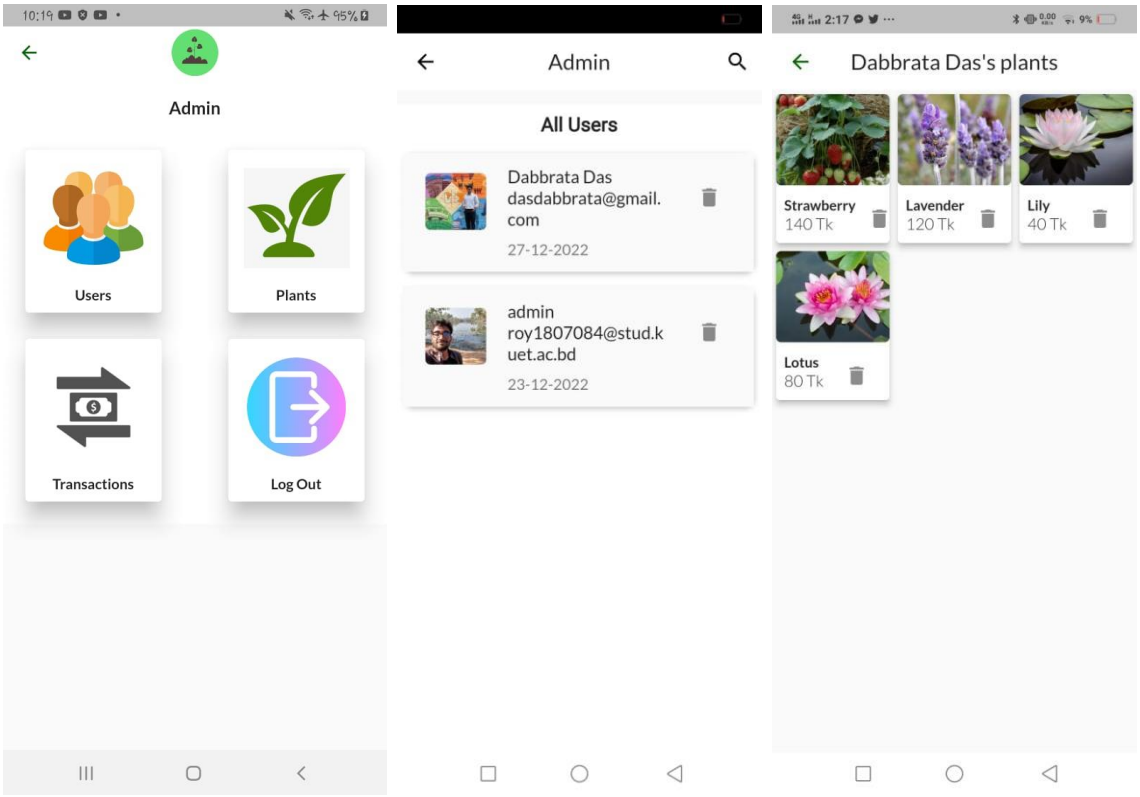

Fig 6.9: Admin and admin features

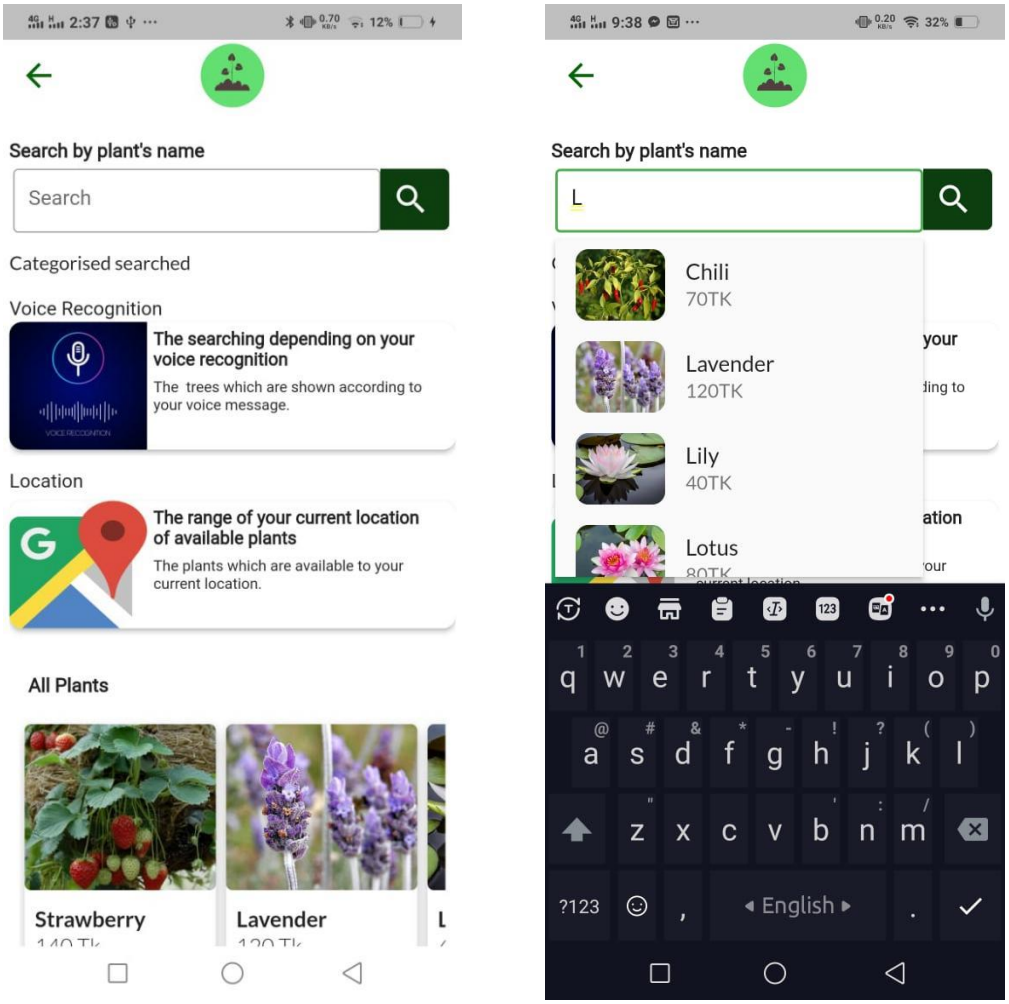

Fig 6.10: Search functionalities

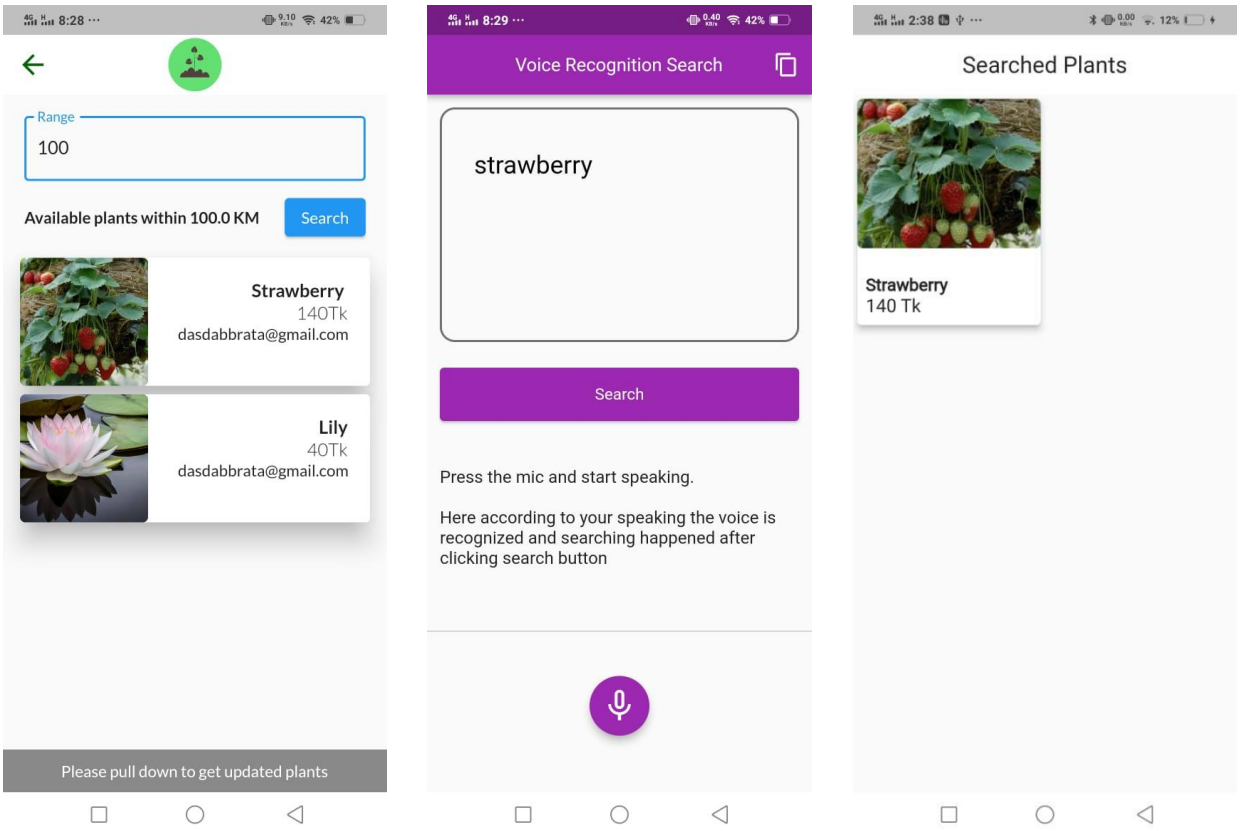

Fig 6.11: Location and voice-recognition based searching

7. Target vs. Actual Accomplishments

7.1. Target

Our target was to build a user-friendly e-commerce application. The application was supposed to contain features that could make searching products easier. Also, we wanted to apply a payment gateway procedure for business purposes. We wanted to make the application more interacting by using different auth providers. Location-based and voice-based searching was one of our main focuses. Another target was to compare the sells of plants and find out the best-selling plants. This would be available to both the user and the admin.

7.2. Actual Accomplishments

We were successful in most of the cases. We successfully completed our location and voice-based searching. We successfully implemented multiple authentication types.

However, we couldn't implement the payment gateway due to limited resources and paid integrations. We also couldn't add comparison bar charts and best-selling plants feature. But we were successful in most of the cases.

8. Limitations

- Since the app is completely internet dependent, internet connection is a must which might become problematic.
- We were unable to add payment gateway functionality due to limited resources.
- Some pages may load slower due to data fetching latency. This is dependent on the internet connectivity speed.

9. Discussion & Conclusion

Developing this app was very interesting and challenging. We encountered lots of bugs, and many times we were stuck. Often, we were presented with challenging situations. We had a hard time trying to implement payment gateway and some other problems. We were not entirely successful. But we overcame most of the problems. We tried to make the app as user-friendly as possible. There is no end to development. But we are satisfied with the output of our hard work and efforts. We are confident that this app will provide a great environment for business. And we are proud that we could take part in something that is helpful for saving the environment. We overcame most of the challenges and enjoyed developing the project. We are sure that this knowledge and experience will help us in future.

10. References

- https://pub.dev/publishers/flutter.dev/packages
- https://docs.flutter.dev/get-started/codelab
- https://www.youtube.com/watch?v=jwlgHLHFIjc&t=852s
- https://stackoverflow.com/
- https://dialogflow.cloud.google.com/#/agent/brikkhayonflutterbotntpu/intens
- https://firebase.google.com/docs/database
- https://invotech.co/blog
- https://developers.google.com/maps