Ayurvedic (Herbal) plant identification system Aayu mobile application – Implementation

Technical Report · May 2021

DOI: 10.13140/RG.2.2.35700.48009

6 authors, including:

Prasandika Balage
University of Westminster
3 PUBLICATIONS **0** CITATIONS
SEE PROFILE

Viraj Lakshitha Bandara
University of Westminster
2 PUBLICATIONS **0** CITATIONS
SEE PROFILE

Aruna Randika
University of Westminster
2 PUBLICATIONS **0** CITATIONS
SEE PROFILE

Anuja Dassanayake
University of Westminster
2 PUBLICATIONS **0** CITATIONS
SEE PROFILE

Ayurvedic (Herbal) plant identification system
Aayu mobile application - Implementation

Bhagya Prasandika[1] (Project Leader, Back-end dev.), Viraj Lakshitha Bandara[1] (Data science dev.), Aruna Randika[1] (Front-end dev), Anuja Jude[1] (Data science dev.), Minura Kariyawasam[1] (Back end dev.), Adeepa Jayawardana[1] (Front end dev.),

1 *Informatics Institute of Technology In collaboration with University of Westminster UK*

Department of Computer Science and Engineering

Table of Contents

List of Figures

List of Tables

Abbreviation

- REST - Representational state transfer
- API - Application Programming Interface
- SVN - Subversion
- IDE - integrated development environment
- GPU - Graphics processing unit
- CPU - central processing unit
- CSV - comma-separated values
- 2D - two-dimensional
- CNN - convolutional neural network
- AWS - Active Amazon Web Services
- RESTful - representational state transfer
- JARs - Java ARchives
- EC2 - Amazon Elastic Compute Cloud
- JSON - JavaScript Object Notation
- DevOps - development and operations
- IEEE - the Institute of Electrical and Electronics Engineers
- FR1- Functional Requirements
- NFR - Non-Functional Requirements
- IIT - Informatics Institute of Technology

Chapter 1 – Implementation

1.1 Chapter Overview

In the preceding chapter, the discussion prioritized to be carried on about the design decisions of the project. This chapter will extend the previous discussion into the level of implementation of the authentic prototype. This chapter will also focus on programmatic decisions taken by the developers working on the implementation of each section, categorized to be as frontend, backend and the data science section. Furthermore, steps taken to avoid blunders and confusion during the implementation process will be considered profoundly.

1.2 Overview of the prototype

The front end was developed by java native, using the IDE Android Studio. The Front end will mainly consist of Aayu Scan, Quiz, Map, About , Help Pages. For instance, when a user scans a leaf and adds location, later the user will be able to search that plant name in the map and find the location of that plant.Once the plant leaf image is captured or uploaded, the models will identify the leaf with higher accuracy and related details for that plant will be displayed to the user. And about the quiz, the user can face 10 different randomized questions with answers where after each question, the correct answer will be displayed and at the end the final score will be displayed. Ayu about and Aayu help will give some facts about the project Aayu and guide the user to how to use the mobile application, respectively.

1.3 Technology selections

1.3.1 Language / Framework Selection

1.3.1.1 Front-end - Java native

Team Aayu first developed the front end of the Aayu mobile application using React-Native. With the ongoing process, the team faced some challenges when connecting the data science component, backend with the frontend. To pass an image taken, NumPy Array was used. Since the NumPy array isn’t available in Java, there wasn't a possible method to transfer the image to the NumPy array. After some research was done, team Aayu was able to connect the components and transfer the image using Java Bitmap which is available in Android Studio. Team Aayu explained the issue to the higher authorities and justified the fact that for this mobile application Aayu, Java native development suits way better than React-Native.

Eventually, the front end developer team of Aayu was able to build up a user-friendly, responsive front end by using Java language and the IDE, Android Studio.

1.3.1.2 Back-end - Java and Spring Boot

Aayu mobile application used Java as the back-end development language and Java-Spring Boot to develop REST APIs of the application. Spring framework is one of the most popular and efficient application development frameworks of Java. It provides a powerful and flexible path to configure database transactions. By using Java-Spring boot for the back-end development process, team Aayu was able to reduce development time and increase the quality and productivity of the application.

1.3.1.3 CI/CD - Jenkins

Continuous Integration and Deployment is the most important part of DevOps that is used to integrate various steps of the CI/CD automation. So, for that aayu team chose Jenkins as the project continues integration tool. Jenkins is an open-source software build in Java with various plugins for continuous Integration purpose. It uses an agent-based architecture, primarily deployed on-premises that can run on the cloud also.

- Limitless integrations: Jenkins can integrate with nearly any outside program used for developing applications. EX: Docker, Kubernetes, Maven, Java

- Plugins: There are a lot of plugins to apply with Jenkins to integrate with Git, SVN, Selenium and many more tools.

- Huge community: Jenkins has a lot of community, to resolve the issues and help others when the integration.

1.3.1.2 Data Science - Python

After evaluating the available programming languages, Python was selected as the main programming language for the implementation of the data science component of the Aayu project. Python was selected mainly due to the following reasons.

1. Availability of libraries and tools such as Scikit-learn and Jupyter.

2. Python is increasingly used in both industry and academic settings as it is extremely powerful and flexible

3. For the most of open-source and industrial project used the python for implementing models

4. High community support to overcome issues during implementation, and easy to solve the issue in python.

5. Availability of tutorials and support with both Machine Learning and Deep Learning tasks with Python Environment

Libraries and Packages Selections

1. **OpenCV** - OpenCV is an open source library, used for computer vision, machine learning and mostly in image processing.It plays a major role in real-time operations which is very important in image processing for classification models.

2. **Keras** - Keras is a powerful and easy-to-use free open source Python library for developing and evaluating deep learning models. Here Keras is used to implement the deep learning convolutional neural network from scratch, data augmentation and also for the pre-processing of the image data.

3. **Scikit Learn** - SciKit Learn is a popular python machine learning library which offers a number of classification, regression and image processing algorithms. To avoid time wasting by re-implementing the algorithm SciKit Learn provides all the pre-build algorithms. Here SciKit-Learn is used to split the dataset into the training and testing sub data sets.

4. **Matplotlib** - Matplotlib is a plotting library for Python used for plotting graphs and visualizing the results. Matplotlib is used in this project to visualize the prediction accuracy of the model by using the confusion matrix in the Jupyter notebook which was used for the implementation of the model.

5. **Tensorflow** - TensorFlow Library is an Open-Source Google Library that's available on GitHub. It is one of the more famous libraries that is used for the Deep Neural Networks. The primary reason behind the popularity of TensorFlow is the sheer ease of creating and deploying heavy applications using TensorFlow. Keras library is also run on top of TensorFlow framework. It is made with the focus of understanding deep learning techniques, such as creating layers for neural networks, using the existing APIs for hands-on projects and mathematical details.

1.4 Implementation of the data science component

1.4.1 Why is python suitable for data science components?

As artificial intelligence and machine learning are being applied in various fields in the corporate sector, health sector as well as in day-today activities demand for artificial intelligence grows rapidly.For the implementation of machine learning,it needs stable, powerful programming language. After doing a lot of background research and gathering advice from the industry experts, python was identified as the most suitable language for machine learning.

When using the python programming language it has a great library ecosystem, low entry barriers, flexibility, and is also platform independent etc.Python supports Keras for deep learning. It allows fast calculations and prototyping, as it uses the GPU in addition to the CPU of the computer and TensorFlow for working with deep learning by setting up, training, and utilizing artificial neural networks with massive datasets, SciKit-image for the data pre-processing, as well as Pandas for high-level data structures and analysis. It allows merging and filtering of data and gathering data from other external sources like CSV files and also Matplotlib for creating 2D plots, histograms, charts, and other forms of data visualization.
Python programming language builds with the everyday English language, and that makes it easy to learn. Its simple syntax and attributes are allowed to work with complex systems, making the clear relationship between the system elements.

Considering all things that were mentioned above it becomes clear that python is undoubtedly the most suitable language for the implementation of the data science components of the Aayu Project.

1.4.2 Why is CNN most suitable for image recognition among other neural networks?

In 2010 The Imagenet Large Scale Visual Recognition Challenge was introduced as a means of testing the effectiveness of different computer vision algorithms by researchers from Stanford and Princeton.

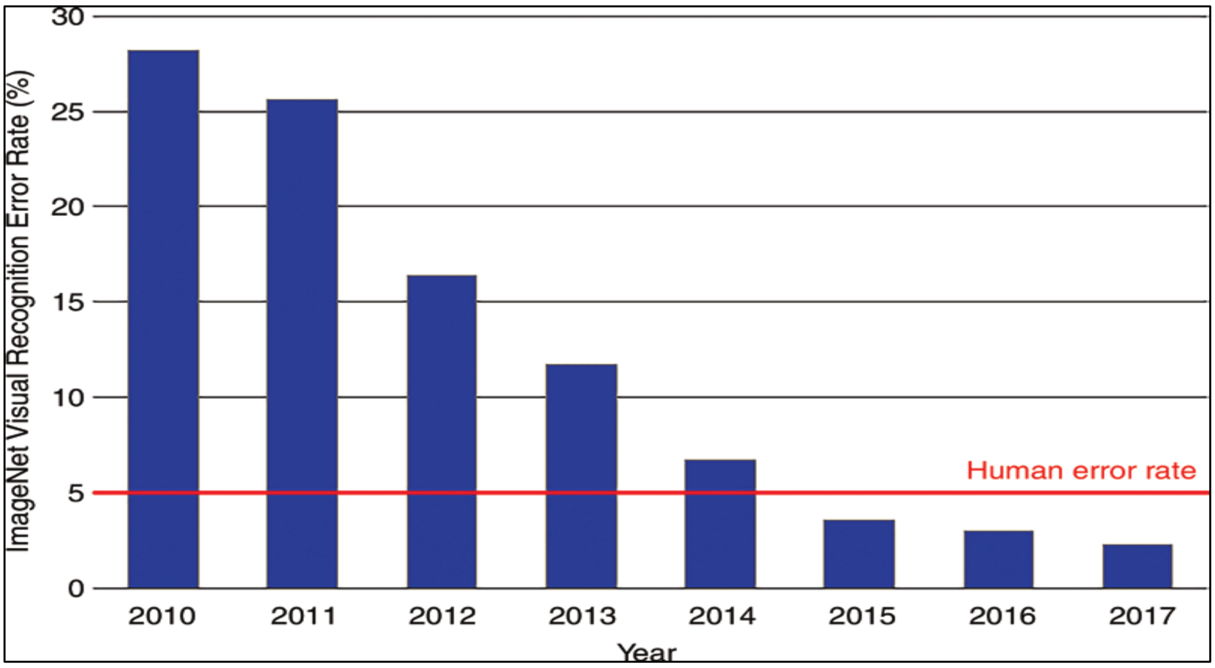


Figure 1 - ImageNet Visual Recognition error rate

The above graph shows the error rate of the top algorithms entered to the competition from its inception up until 2015. When observing the graph we can see that even though the error rate had a downward trend from the first 2 years in 2012 it was drastically reduced. This was the year that AlexNet, a convoluted neural network, was introduced. This proves that Convoluted Neural Networks are the most suitable technology for the purpose of image classification.

1.5 Implementation of the backend component

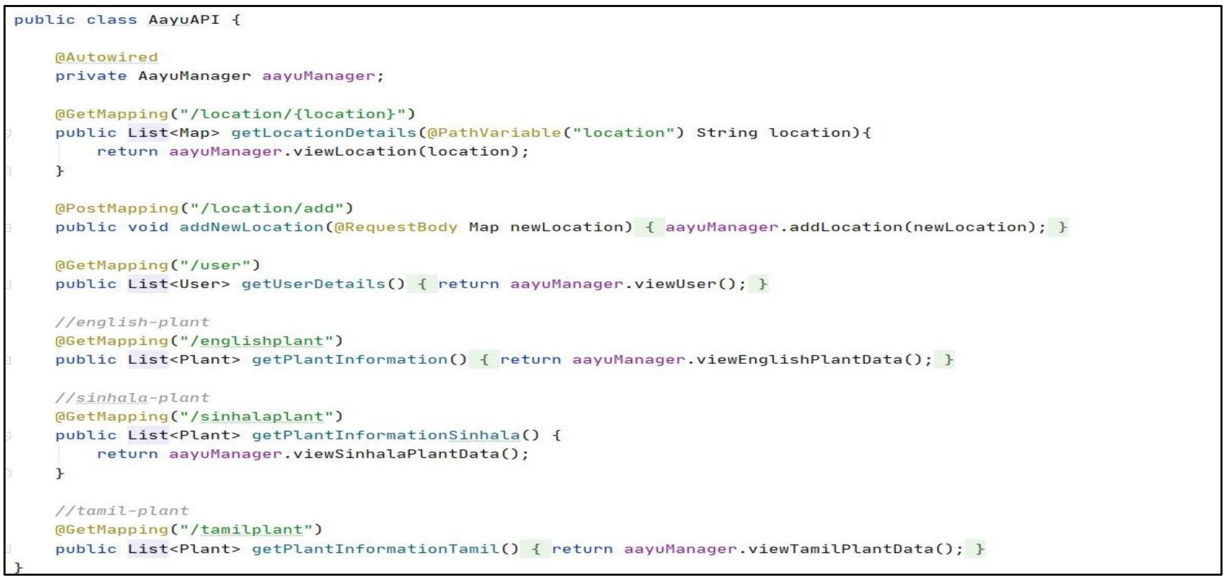
1.5.1 Implementation of the backend component

Back-end and server-side implementation of the Aayu mobile application has been done in several stages. Aayu team has selected Java-Spring Boot as the backend framework and Amazon Web Services (AWS) as the cloud computing platform.

1.5.2 Backend framework (Java-Spring Boot)

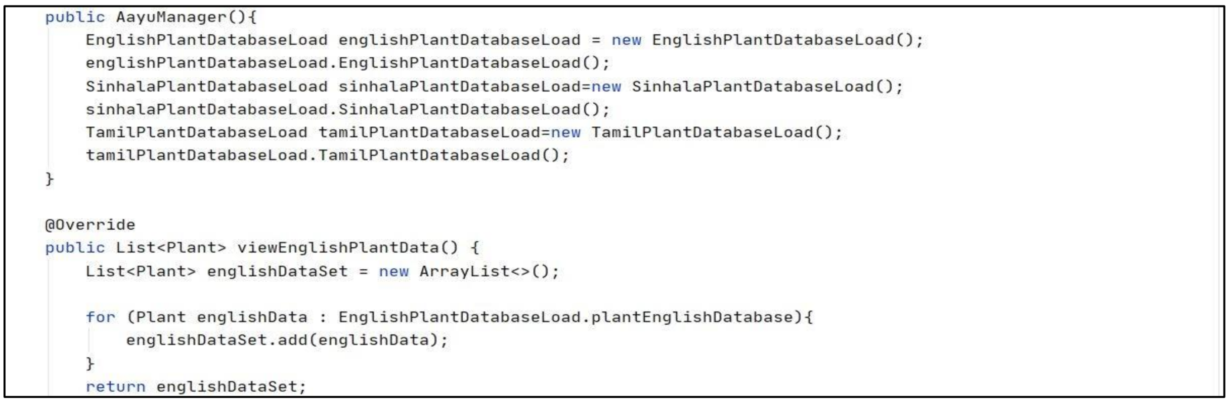
Spring framework is known as the most popular and efficient application development framework of Java. Spring Boot is a module of Spring frameworks. Spring Boot is ideal for RESTful services and it allows the development of stand-alone applications with a minimum number of configurations.

Spring Boot provides a powerful and flexible path to configure database transactions. As mentioned earlier, the project team has selected AWS as the cloud computing platform and it is also highly supportive for Spring Boot. Furthermore, Spring Boot has a concept of starter in the pom.xml file that Spring Boot can download the dependencies JARs based internally on Spring Boot requirement. Features of Spring Boot reduces development time and increases the quality and productivity of the application. By considering all the above factors, Aayu used Spring Boot to develop REST APIs of the application. (Zhou, Li, Luo and Chou, 2014)



```java
public class AayuAPI {

    @Autowired
    private AayuManager aayuManager;

    @GetMapping("/location/{location}")
    public List<Map> getLocationDetails(@PathVariable("location") String location){
        return aayuManager.viewLocation(location);
    }

    @PostMapping("/location/add")
    public void addNewLocation(@RequestBody Map newLocation) { aayuManager.addLocation(newLocation); }

    @GetMapping("/user")
    public List<User> getUserDetails() { return aayuManager.viewUser(); }

    //english-plant
    @GetMapping("/englishplant")
    public List<Plant> getPlantInformation() { return aayuManager.viewEnglishPlantData(); }

    //sinhala-plant
    @GetMapping("/sinhalaplant")
    public List<Plant> getPlantInformationSinhala() {
        return aayuManager.viewSinhalaPlantData();
    }

    //tamil-plant
    @GetMapping("/tamilplant")
    public List<Plant> getPlantInformationTamil() { return aayuManager.viewTamilPlantData(); }
}
```

Figure 2 - Application Programming Interface(API) links



```java
    public AayuManager(){
        EnglishPlantDatabaseLoad englishPlantDatabaseLoad = new EnglishPlantDatabaseLoad();
        englishPlantDatabaseLoad.EnglishPlantDatabaseLoad();
        SinhalaPlantDatabaseLoad sinhalaPlantDatabaseLoad=new SinhalaPlantDatabaseLoad();
        sinhalaPlantDatabaseLoad.SinhalaPlantDatabaseLoad();
        TamilPlantDatabaseLoad tamilPlantDatabaseLoad=new TamilPlantDatabaseLoad();
        tamilPlantDatabaseLoad.TamilPlantDatabaseLoad();
    }

    @Override
    public List<Plant> viewEnglishPlantData() {
        List<Plant> englishDataSet = new ArrayList<>();

        for (Plant englishData : EnglishPlantDatabaseLoad.plantEnglishDatabase){
            englishDataSet.add(englishData);
        }
        return englishDataSet;
```

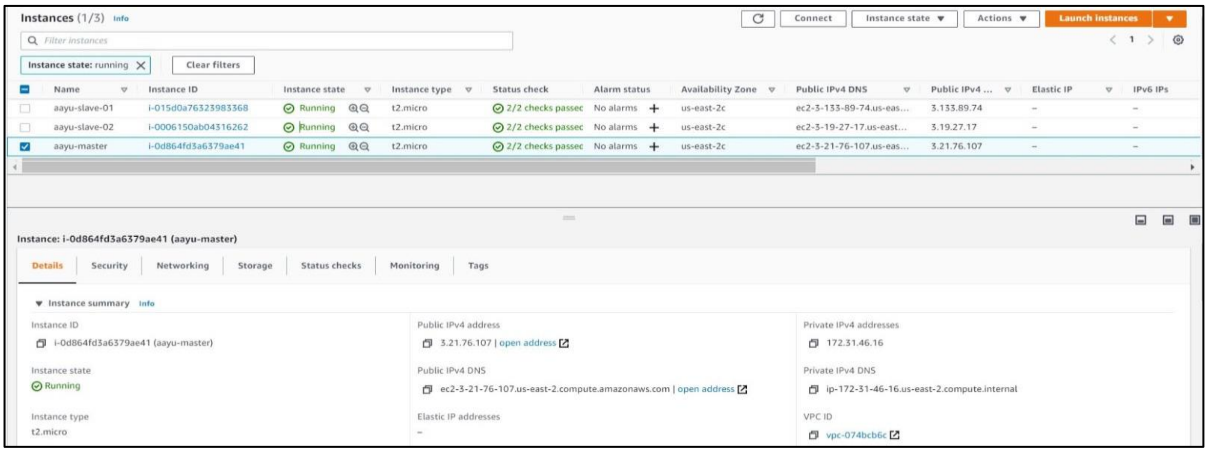
Figure 3 - Aayu back-end - returning plant information

1.5.3 Cloud computing platform

1.5.3.1 Amazon Web Services

Aayu team used the EC2 service to deploy the Aayu mobile application in AWS. Amazon Elastic Compute Cloud is a component of Amazon.com's Amazon Web Services cloud computing platform that allows users to rent virtual machines to run computer applications. (Jackson et al., 2010)

Therefore, the Aayu back-end (Java-Spring Boot application) process runs in the EC2 service of the AWS, and some information (Plant information) is stored in the same space as JSON files. All settings are created according to security standards.


Instances (1/3) Info

Instance state: running | Clear filters

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP	IPv6 IPs
aayu-slave-01	i-015d0a76323983368	Running	t2.micro	2/2 checks passec	No alarms	us-east-2c	ec2-3-133-89-74.us-eas...	3.133.89.74	–	–
aayu-slave-02	i-0006150ab04316262	Running	t2.micro	2/2 checks passec	No alarms	us-east-2c	ec2-3-19-27-17.us-east...	3.19.27.17	–	–
aayu-master	i-0d864fd3a6379ae41	Running	t2.micro	2/2 checks passec	No alarms	us-east-2c	ec2-3-21-76-107.us-eas...	3.21.76.107	–	–

Instance: i-0d864fd3a6379ae41 (aayu-master)

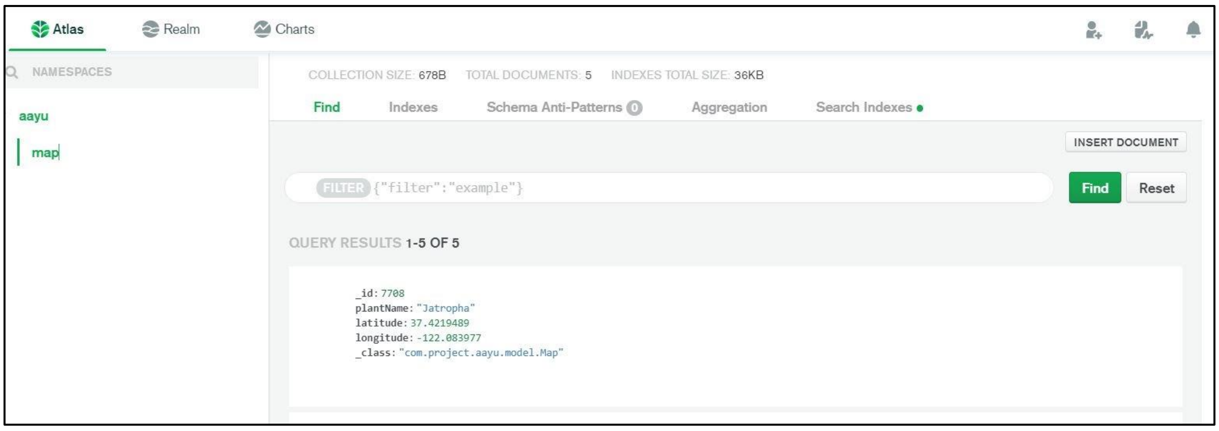
Instance summary Info

Instance ID: i-0d864fd3a6379ae41 (aayu-master)	Public IPv4 address: 3.21.76.107 | open address	Private IPv4 addresses: 172.31.46.16
Instance state: Running	Public IPv4 DNS: ec2-3-21-76-107.us-east-2.compute.amazonaws.com | open address	Private IPv4 DNS: ip-172-31-46-16.us-east-2.compute.internal
Instance type: t2.micro	Elastic IP addresses: –	VPC ID: vpc-074bcb6c

Figure 4 - AWS console - Running instances

1.5.3.2 MongoDB

MongoDB is cross-platform document-oriented database software. It is a NoSQL database software that works with JSON-like documents and optional schemas. Aayu team used MongoDB to store user information and map data information of the Aayu application.


NAMESPACES: aayu / map

COLLECTION SIZE: 678B   TOTAL DOCUMENTS: 5   INDEXES TOTAL SIZE: 36KB

FILTER {"filter":"example"}

QUERY RESULTS 1-5 OF 5

```
_id: 7708
plantName: "Jatropha"
latitude: 37.4219489
longitude: -122.083977
_class: "com.project.aayu.model.Map"
```

Figure 5 - Stored information of mongoDB

1.6 Implementation of the front-end component

It is way more clear that Java is a platform-independent language, backed by a huge community and with support for legacy support libraries. When it comes to android app development, by using java, it becomes more possible to write codes to easily manage and control android devices.

Java comes with so many libraries that can be used to fulfill almost any kind of requirement. These libraries provide functionalities for data structure,math,graphical implementations,data science and related models etc.

Java supports multi-threading. That is very important when it comes to fetching data from APIs and running Data Science Models real time. Java can also handle big processes without facing crashes whatsoever, that is an added benefit too.

Java is very easy to learn with so many resources and backed by a huge community. As a language Java has fluent English-like syntaxes and less and less special characters.Android Studio is a very powerful and easy to use IDE and that also supports many plugins,libraries and dependencies.

1.7 Deployments/CI-CD Pipeline

Aayu application is a full-stack mobile application that contains two deep learning models to identify the ayurvedic value of a plant. So as a scalable mobile application, aayu followed the deployment strategy on AWS (Amazon web services), which is the best on-demand cloud service in the industry, used to deploy powerful and scalable applications.

1.7.1 CI/CD pipeline

Continuous integration (CI) is a development practice where developers can merge their changes into the main branch and then automated the build and test process, providing test feedback as well.

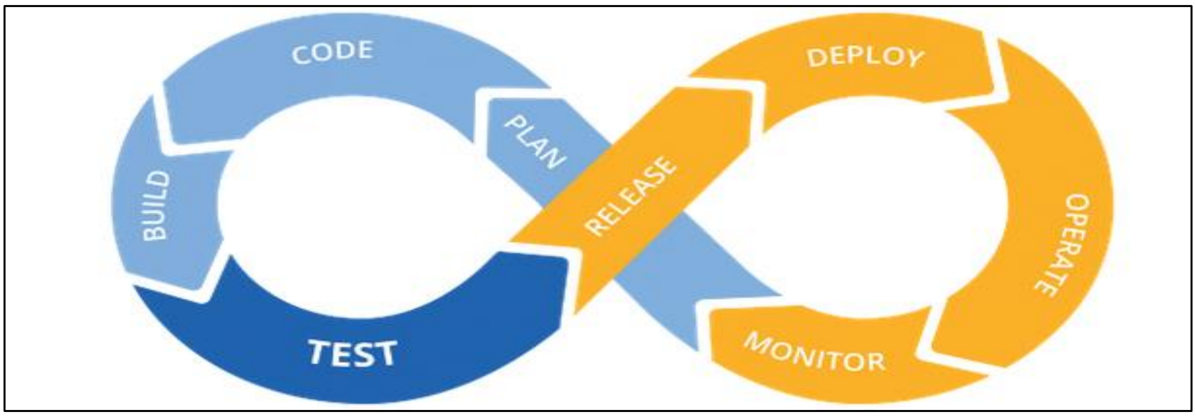


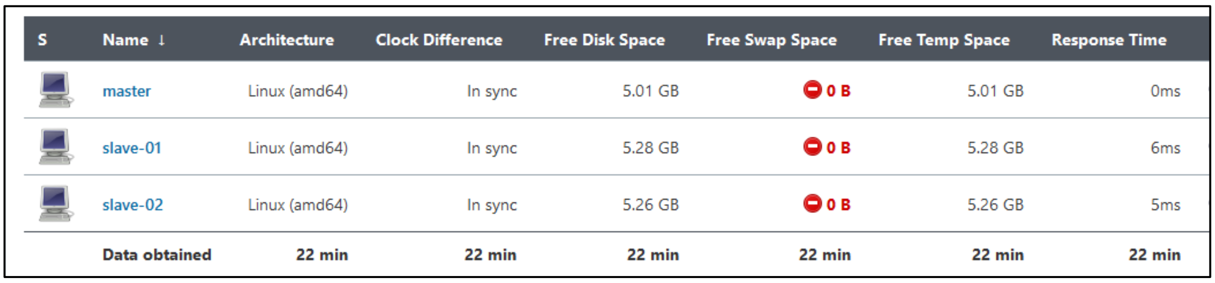
Figure 6 - DevOps Architecture

Continues delivery (CD) is a process of automating the application, version control system to end-users. Besides the help of CD, software development teams can build, test, and go to the production level faster and more frequently. Furthermore, it helps the organization to be more agile, innovative, and startups.

1.7.2 How Aayu applied CI/CD pipeline

AWS is one of the best on-demand cloud services providers in the modern tech world. Plus, with the help of Jenkins, aayu creates a CI/CD pipeline to effect code changes automatically build, tested, and prepare to deploy on the production environment more instantly.

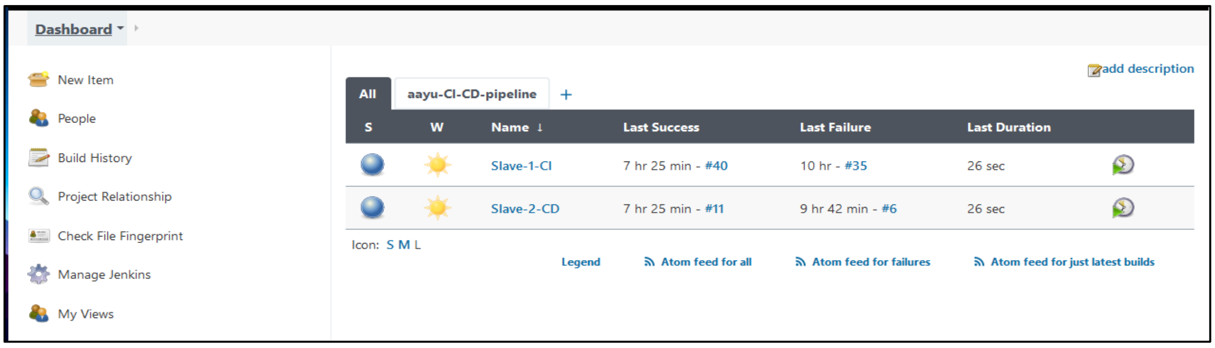
During the implementation of the aayu pipeline, Aayu used popular master-slave architecture to maintain the build quality in the CI/CD pipeline. Aayu used the master server to handle most of the key components of the pipeline. Scheduling build jobs monitor the slaves (possibly taking them online and offline as required). Plus, the master instance of Jenkins can also execute build jobs directly. Parallelly, aayu used salve servers to execute different jobs under different conditions. Slave servers can run different operating systems and the job of slave servers do as they are told to, which includes executing the build, testing and deployment as well.



S	Name ↓	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	master	Linux (amd64)	In sync	5.01 GB	⛔ 0 B	5.01 GB	0ms
	slave-01	Linux (amd64)	In sync	5.28 GB	⛔ 0 B	5.28 GB	6ms
	slave-02	Linux (amd64)	In sync	5.26 GB	⛔ 0 B	5.26 GB	5ms
	Data obtained	22 min	22 min	22 min	22 min	22 min	22 min

Figure 7 - Jenkins Slave-01, Slave-02 agents

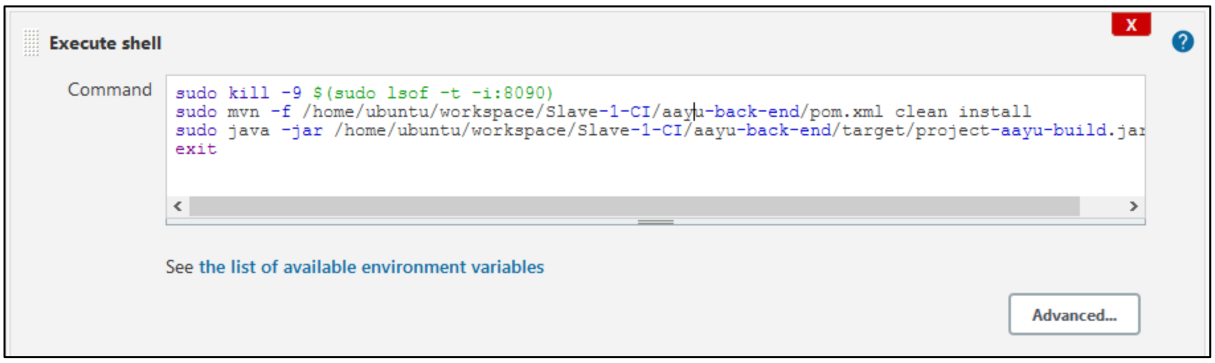
Every change that occurred on the aayu repository gets notified by Jenkins and it starts to build the project rapidly. One of the slave servers runs the build and the test executing and if the test execution is done stably, then the deployment server starts to deploy the packages. The team has chosen two separate ubuntu servers for the integration and deployment component. One for testing and if the test build was successful then it deploys on the slave-01 server as well. Following that post-build will execute to start the slave-2 production-level server. This is the server aayu used for deploying the production level application.



S	W	Name ↓	Last Success	Last Failure	Last Duration
●	☀	Slave-1-CI	7 hr 25 min - #40	10 hr - #35	26 sec
●	☀	Slave-2-CD	7 hr 25 min - #11	9 hr 42 min - #6	26 sec

Figure 8 - Jenkins jobs that use to build the api and the deloy

Slave-1 is fully accountable for executing the continuous integration part. The team created a separate job called Project Slave-1-CI that automates the primary testing part using shell commands.



```
sudo kill -9 $(sudo lsof -t -i:8090)
sudo mvn -f /home/ubuntu/workspace/Slave-1-CI/aayu-back-end/pom.xml clean install
sudo java -jar /home/ubuntu/workspace/Slave-1-CI/aayu-back-end/target/project-aayu-build.jar
exit
```

Figure 9 - Shell command that use for build and deploy the api

Following code responsible for building maven dependencies, run the aayu build jar on the tomcat server. If the tomcat server starts without any error, post-build action will start the execution of the Project Slave-2-CD jobs. Project Slave-2-CD is accountable for deploying the API on the production server, this shell commands run only if the job-01 executes successfully on the agent-01 slaver.

After a lot of configurations and implementation aayu team came up with the fully automated CI/CD pipeline based on the Master slave architecture.

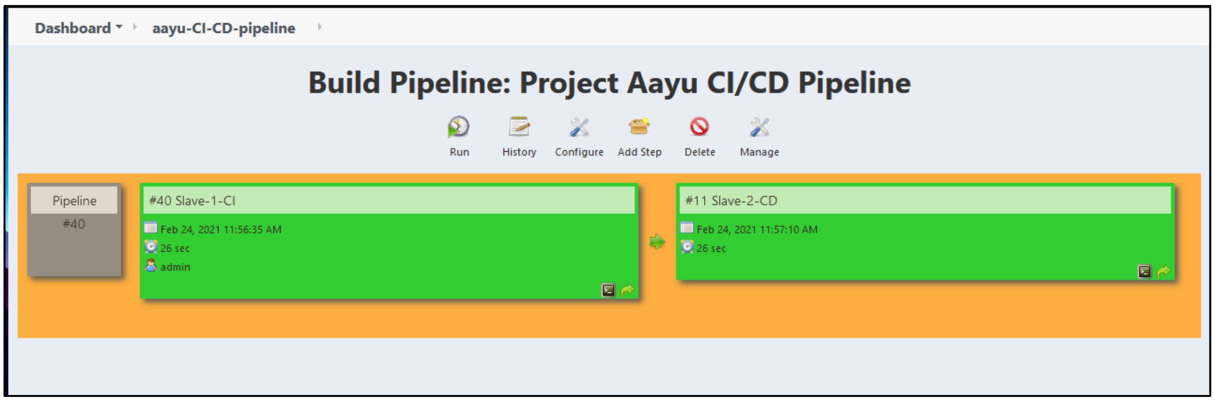


Figure 10 - Aayu CI/CD pipeline view

1.8 Chapter Summary

This chapter focuses on the implementation phase of the Aayu project. Firstly, the prototype of the system is discussed. The reasoning and the advantages and disadvantages of the technology stack which has been selected has been presented. Thereafter a detailed discussion of the implementation of the data science, backend and frontend components are carried out. Finally, the deployment and the CI/CD pipeline are examined.

Chapter 2 – Testing

2.1 Chapter Overview

This chapter solely focuses on the testing and the evaluation aspect of the Aayu mobile application. The core motivation of this phase is to verify that the app performs in the intended way. Here we conduct multiple types of testing procedures which evaluate the mobile app from different angles, such as performance testing, usability testing, compatibility testing to mention a few.

2.2 Testing Criteria

To analyse the test procedure, intentions, scheme and estimate the required deliverables, the necessity of the test plan template engages in a critical stage. For these testing procedures, IEEE 829 test plan template is used. In this stage functional testing, non- functional testing procedures will be taken into consideration. Expectations of these testings will be to determine whether the required outcome will be achieved from the Aayu mobile application.

2.3 Testing functional requirements

Functional requirement testing is used to substantiate the necessity of the application Aayu with the use of the black box testing. Functional requirements of the app were discussed in previous chapters and testings for those functionalities will be summarized below.

Test case	Feature Tested	Test case description	Test case status	Promised Result	Real Result	Result Status
1.0	FR1	Scan healthy plant leaf	Scan plant leaf by taking a photo from the mobile phone camera	Displaying necessary details related to the scanned plant leaf, in the user's preferred language with	7 out of 10 times, plants were identified correctly when photos taken from the phone were scanned	Pass (80%)

				nearly 80% accuracy		
2.0	FR1	Scan healthy plant leaf	Scan plant leaf by uploading a photo from the mobile phone gallery	Displaying necessary details related to the scanned plant leaf, in the user's preferred language with nearly 90% accuracy	9 out of 10 times, plants were identified correctly when photos taken from the phone were scanned	Pass (90%)
3.0	FR1	Scan diseased plant leaf	Scan a diseased plant leaf by taking a photo from the mobile phone camera	Displaying necessary details related to the scanned plant in the user's preferred language with nearly 80% accuracy	3 out of 5 times, plants were identified correctly when photos taken from the phone were scanned	Pass (80%)
4.0	FR1	Scan diseased plant leaf	Scan diseased plant leaf by uploading a photo from the mobile phone gallery	Displaying necessary details related to the scanned plant leaf, in the user's preferred language with	4 out of 5 times, plants were identified correctly when photos taken from the phone were scanned	Pass (90%)

				nearly 90% accuracy		
5.0	FR2	Suggesting Other plants with similar qualities	Suggesting plants with similar qualities to the identified plant after scanning the plant leaf	Displaying plants with similar qualities, related to the scanned plant	All the time similar plant suggestion works correctly	Pass (100%)
6..0	FR3	Searching the added plant from the Aayu Map	Search plant location using the plant name	Plant location will be displayed correctly in the map using a marker with the user name who added the plant	All the plant locations were displayed correctly when searched by the plant name.	Pass (100%)
7.0	FR5	Aayu quiz	Aayu quiz to test user's knowledge about ayurvedic plants	Quiz working correctly with randomized questions and answers	Quiz works properly with randomized questions and answers	Pass (100%)
8.0	FR4	Tri Language Selector	Display plant information using three languages	Display plant details related to the scanned plant leaf in the user	All plant details display in user preferred language (English, Sinhala, Tamil)	Pass (100%)

			(Sinhala, English, Tamil)	preferred language		

Table 1 - Functional requirements

2.4 Testing non-functional requirements

Requirement List		Priority Level	Description
NFR 1	Accuracy	Critical	*Maintenance issues should be accurately highlighted whereas non-maintenance issues should not be highlighted*
NFR 2	Performance	Critical	*The application should perform without experiencing significant lagging or other performance-related problems*
NFR 3	Reliability	Critical	*The application must perform reliably for the purpose of leaf recognition*
NFR 4	Usability	Critical	*The application must behave good user interface and user experience*
NFR 5	Scalability	Critical	*The application should be implemented with scalability and maintainability in mind also having an orderly code repository*

Table 2 - Non-functional requirements

More non-functional test cases were performed and they can be found in Appendix Chapter 2 - NFR.

2.5 Unit testing

2.5.1 Black Box testing

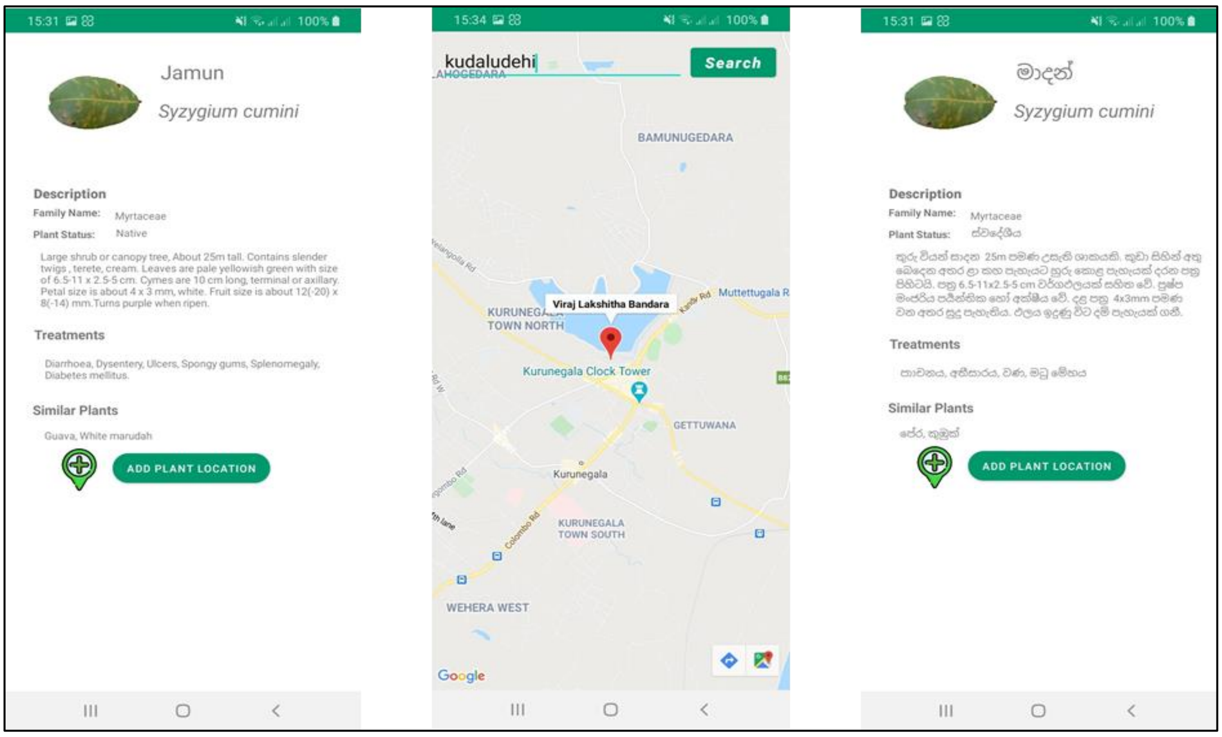
Unit tests were done during the development of the Aayu application. Test results along with screenshots are comes in Appendix Chapter 2 - Black Box testing.

2.6 Performance testing

Performance Testing is an application testing the acceleration, response time, stability, reliability of a software application under a critical workload.

Performance Testing secures the Aayu mobile application reliability on every mobile phone. Aayu application is essentially focused on an android platform only. Hence aayu team test the performance of the chosen android phones. Xiaomi Pocophone F1, Samsung Galaxy A20 and Huawei nova 7i were used to test the performance of the aayu mobile application.

Response times were also checked and optimizations were done to enhance the performance of the aayu mobile application by advancing the API quality and applying CI/CD to the aayu project. Every function continuously tested individually in the above mobile phones. No stability or reliability error was identified through the Performance testing.



Jamun

Syzygium cumini

Description

Family Name:	Myrtaceae
Plant Status:	Native

Large shrub or canopy tree, About 25m tall. Contains slender twigs , terete, cream. Leaves are pale yellowish green with size of 6.5-11 x 2.5-5 cm. Cymes are 10 cm long, terminal or axillary. Petal size is about 4 x 3 mm, white. Fruit size is about 12(-20) x 8(-14) mm.Turns purple when ripen.

Treatments

Diarrhoea, Dysentery, Ulcers, Spongy gums, Splenomegaly, Diabetes mellitus.

Similar Plants

Guava, White marudah

මාදන්

Syzygium cumini

Description

Family Name:	Myrtaceae
Plant Status:	ස්වදේශීය

තුරු වියන් සාදන 25m පමණ උසැති ශාකයකි. කුඩා සිහින් අතු බෙදෙන අතර ළා කහ පැහැයට හුරු කොළ පැහැයක් දරන පත්‍ර පිහිටයි. පත්‍ර 6.5-11x2.5-5 cm වර්ගඵලයක් සහිත වේ. පුෂ්ප මංජරිය පර්යන්තික හෝ අක්ෂීය වේ. දළ පත්‍ර 4x3mm පමණ වන අතර සුදු පැහැතිය. ඵලය ඉදුණු විට දම් පැහැයක් ගනී.

Treatments

පාචනය, අතීසාරය, වණ, මධු මේහය

Similar Plants

පේර, කුඹුක්

Figure 11 - Performance testing

2.7 Usability testing

Usability testing is used to measure the user friendliness of the software application. A small group of targeted audience asked to use the given software and asked for feedback for their usage with the application.

Aayu application is developed following the User Experience principles thouroghly. The Aayu team has gathered a group of IIT students as test subjects and given relevant instructions how to use the Application. After a certain amount of time, tests subjects were asked to give feedback on the experience with the Application.

Improvements such as font size changes, color changes were done according to the feedback. Test subjects were using different models of mobile phones with different sizes of screens and the Application performed as expected.

2.8 Compatibility testing

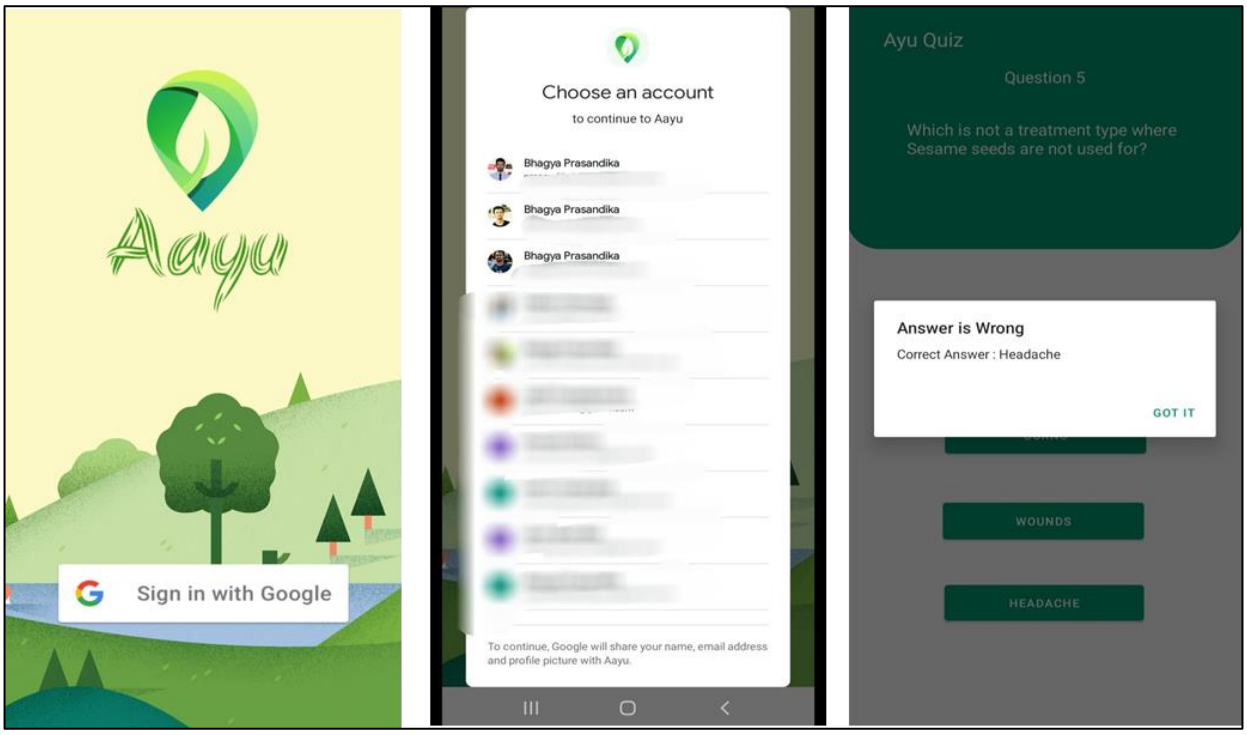
2.8.1 Screen captures of testing scenarios

Compatibility testing is a section of non-functional testing conducted on any software (mobile/web/other) to ensure the application's compatibility with the different environments or devices.

Compatibility testing makes sure that the Aayu mobile application runs well on all mobile phones. Aayu mobile application is designed as an Android application. Therefore, the Aayu team tested the application only within the Android platform. Aayu mobile application was tested using three different mobile phone brands. Samsung Galaxy A20, Xiaomi Pocophone F1, and Huawei nova 7i were used to test the application. All the functionalities and application windows worked well in the above different mobile phones.

All screens and functions are tested separately in the above mobile phones. All functions are worked fine on every screen and no error was identified during the Compatibility testing.

2.8.1.1 Main page, login and quiz pages



To continue, Google will share your name, email address and profile picture with Aayu.

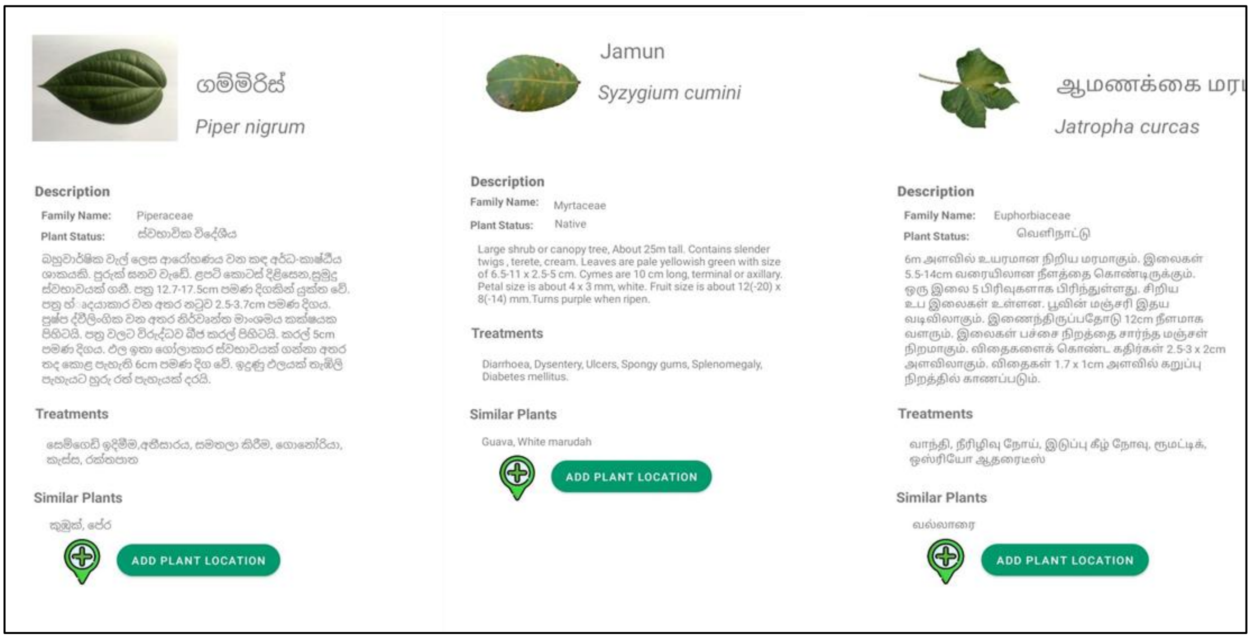
Which is not a treatment type where Sesame seeds are not used for?

Answer is Wrong

Correct Answer : Headache

Figure 12 - Main page and login page

2.8.1.3 Plant information pages



ගම්මිරිස්

Piper nigrum

Description

Family Name: Piperaceae

Plant Status: ස්වභාවික විදේශීය

බහුවාර්ෂික වැල් ලෙස ආරෝහණය වන කඳ අර්ධ-කාෂ්ඨීය ශාකයකි. පුරුක් සහිත වැඩේ. ළපටි කොටස් දිළිසෙන,සුමුදු ස්වභාවයක් ගනී. පත්‍ර 12.7-17.5cm පමණ දිගකින් යුක්ත වේ. පත්‍ර හෘදයාකාර වන අතර නටුව 2.5-3.7cm පමණ දිගය. පුෂ්ප ද්වීලිංගික වන අතර නිර්වෘන්ත මාංශමය කක්ෂයක පිහිටයි. පත්‍ර වලට විරුද්ධව බීජ කරල් පිහිටයි. කරල් 5cm පමණ දිගය. ඵල ඉතා ගෝලාකාර ස්වභාවයක් ගන්නා අතර තද කොළ පැහැති 6cm පමණ දිග වේ. ඉදුණු ඵලයක් තැඹිලි පැහැයට හුරු රත් පැහැයක් දරයි.

Treatments

සෙම්ගෙඩි ඉදිමීම,අතීසාරය, සමතලා කිරීම, ගොනෝරියා, කැස්ස, රක්තපාත

Similar Plants

කුඹුක්, පේර

ADD PLANT LOCATION

Jamun

Syzygium cumini

Description

Family Name: Myrtaceae

Plant Status: Native

Large shrub or canopy tree, About 25m tall. Contains slender twigs , terete, cream. Leaves are pale yellowish green with size of 6.5-11 x 2.5-5 cm. Cymes are 10 cm long, terminal or axillary. Petal size is about 4 x 3 mm, white. Fruit size is about 12(-20) x 8(-14) mm.Turns purple when ripen.

Treatments

Diarrhoea, Dysentery, Ulcers, Spongy gums, Splenomegaly, Diabetes mellitus.

Similar Plants

Guava, White marudah

ADD PLANT LOCATION

ஆமணக்கை மர

Jatropha curcas

Description

Family Name: Euphorbiaceae

Plant Status: வெளிநாட்டு

6m அளவில் உயரமான நிறிய மரமாகும். இலைகள் 5.5-14cm வரையிலான நீளத்தை கொண்டிருக்கும். ஒரு இலை 5 பிரிவுகளாக பிரிந்துள்ளது. சிறிய உப இலைகள் உள்ளன. பூவின் மஞ்சரி இதய வடிவிலாகும். இணைந்திருப்பதோடு 12cm நீளமாக வளரும். இலைகள் பச்சை நிறத்தை சார்ந்த மஞ்சள் நிறமாகும். விதைகளைக் கொண்ட கதிர்கள் 2.5-3 x 2cm அளவிலாகும். விதைகள் 1.7 x 1cm அளவில் கறுப்பு நிறத்தில் காணப்படும்.

Treatments

வாந்தி, நீரிழிவு நோய், இடுப்பு கீழ் நோவு, ருமட்டிக், ஒஸ்ரியோ ஆதரைடீஸ்

Similar Plants

வல்லாரை

ADD PLANT LOCATION

Figure 13 - Plant information pages

2.8.1.4 Map information page

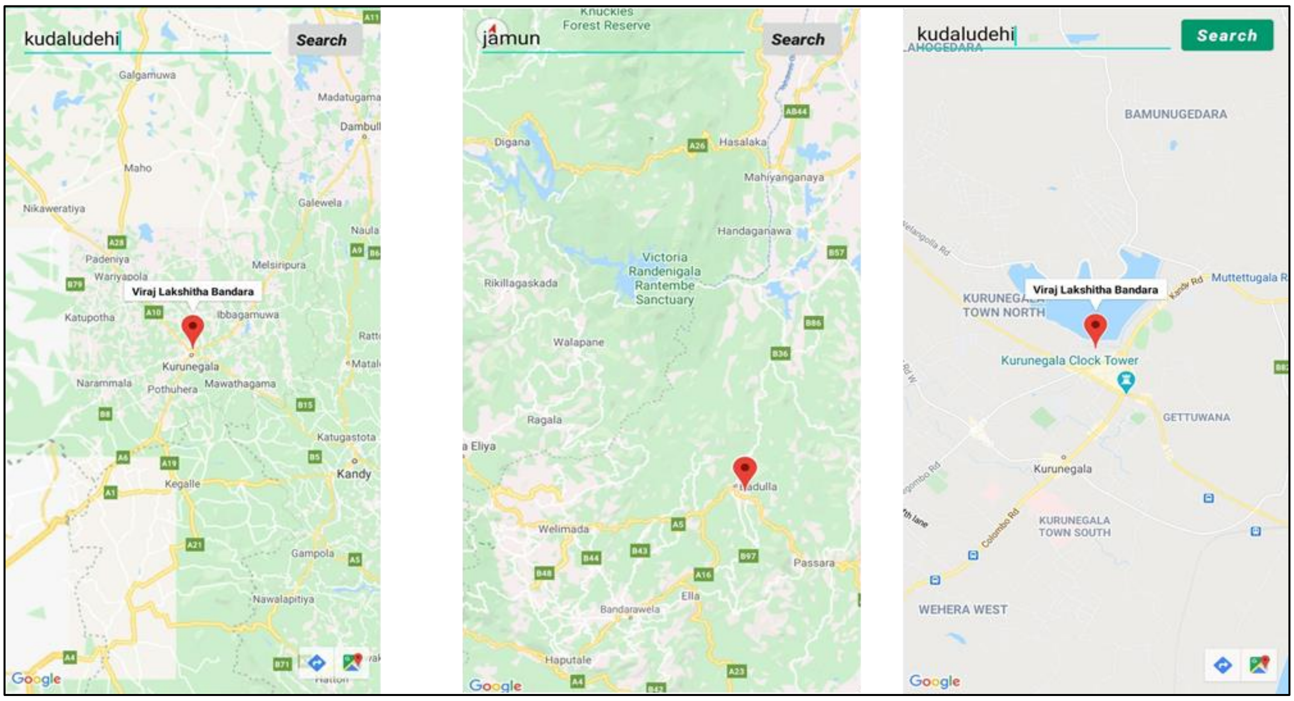


Figure 14 - Map information page

2.9 Chapter Summary

This chapter centred around the testing and validation of the Aayu project where both functional and non-functional requirements were tested at length. Due to the team following an orderly method in implementing the project, the issues and errors were minimized and all minor errors which were discovered were swiftly resolved to ensure that they don't impact the system in the future. While all errors that were discovered were resolved, certain limitations in the system which were due to certain limitations and constraints were discovered. These limitations can be overcome in future iterations of the system.

Chapter 3 – Evaluation

3.1 Chapter Overview

In the previous chapter the Aayu application was put through its paces and thoroughly tested in all its aspects where it was proved that its performance is reliable and accurate. This chapter focuses on more holistic evaluation of the application, focusing not only on quantitative evaluation but also including qualitative evaluation. The application, for the first time, is given to third parties such as end-users, domain experts and industry representatives to gather feedback and their comments are documented for any future enhancements to the application.

3.2 Evaluation methods

The prototype and processes were evaluated for this project by using the methods used during the implementation process. The prototype was evaluated to confirm the below criteria separately.

1. The front end, back end, data science component, server-side deployment are connected together.
2. All the core functions are working properly.
3. Ensure the user can use the application without any issue.

3.3 Quantitative evaluation

Evaluation is extremely important in order to make sure that everything is working up to expected standards and the results are accurate. Qualitative evaluation was done to test the accuracy, performance, reliability and usability of the mobile applications using the feedback from evaluators and domain experts to verify if all the requirements are met successfully. Each step was visualized in the code and graphs were plotted using the python matplotlib library to analyse each step in the model. This is further elaborated under the testing chapter under section 7.5. Detailed test cases and evaluation results can be found under Appendix E – Testing Chapter.

3.4 Qualitative evaluation

3.4.1 Feedback gathered from domain experts

- N.Chandrika Karunarathne (DA:Specialized in Bone fractures and Dislocations)

“ Brilliant Application. I was so amazed about this product Aayu. As of my knowledge and experience within the domain Ayurveda, I have to admit the fact that this is a necessity at the time. This app allows youngsters to get more interested within the Ayurvedic field which is a positive point within the developing society. And the availability of plant information in three main languages and the feature that shows the plant location are valuable facts to be considered. Overall it is a simple, user friendly app with valuable features. ”

3.4.2 Feedback gathered from end-users

- Rajitha Dissanayake (Student at University of Kelaniya)

“This is a cool, user friendly app with a brilliant user interface. This is a very useful mobile app for everyone within any age. I haven't used any app which can identify ayurvedic plants by scanning the plant leaf and displaying information in the three main languages available in Sri Lanka. Hope to see new features too. ”

3.4.3 Feedback gathered from industry experts

- Undisclosed (Software Engineer, Freelancer)

“Overall this is a simple, but very useful and user friendly mobile app. The developer team has put together a lot of work. Improve the scan feature a bit more to enhance the user experience. Keep up the good work”

3.5 Self evaluation

The Aayu prototype addresses the details of the concern under the problem domain chapter 1 in SRS. This prototype can only be used to identify the ayurvedic value of a plant. However, in the future, it can be upgraded to interact with other domains as well because of all the functions and the API built dynamically. Certain things regarding the project have changed because of the time restraint aayu followed.

For a second-year undergraduate scheme, the reach of the project resembles adequate. The implementation was done after considering a lot of different options. Although the tech stack has been changed because of the issues, Aayu team encountered during the implementation. The data science component operates satisfactorily for the problem identified, furthermore accuracy can be increased with future enhancements. Due to time restrictions, team aayu only focuses on the essential features identified in chapter 1 in SRS. The released aayu mobile application accomplishes reliability and testing outcomes to ensure that the application operates perfectly.

3.6 Chapter Summary

This chapter documented the evaluation process of the Aayu application, where the application was evaluated on multiple fronts. The user feedback collected was overwhelmingly positive with some advice and suggestions which would help to further improve the application. The team has also performed a critical review of their work on the application and stated their view on the overall project

Chapter 4 – Conclusion

4.1 Chapter Overview

This chapter is dedicated to discussing the conclusion of the whole project and its successful completion of the goals and objectives. Furthermore, the challenges the team faced during the implementation and design of the project will also be briefly discussed within the chapter. The ethical, Legal and Social effects of the project will also be emphasized. Limitations of the research, future enhancement of the Application will also be mentioned. A full overview of the Extra work done by the team members can be obtained from this chapter. This chapter will end with the concluding remarks.

4.2 Achievements of aims and objectives

Team Aayu was able to accomplish the promised features within the scope within the given period. Completed intentions inside the project are summarized below.

1. The commencement document of the project has defined Aayu, explaining the problem domain, project management methodologies, resources and requirements for the project, aim, and the scope within the given guidelines.

2. Literature Review focused on existing work within the domain related to Aayu, which is in this case the Ayurvedic medicinal field. Each existing work under this domain was critically analysed and the most suitable approaches for Aayu were identified.

3. Methodologies used in the project Aayu were deeply discussed in the project management chapter. Gantt chart, activity schedules were created and OOD principles were followed in the design approach, where OOP programming methodologies were followed in the implementation stage.

4. Data collection techniques carried out for the requirement gathering of project Aayu were deeply discussed in the System Requirements Specification chapter. Interview sessions were carried out with domain experts, Various types of questionnaires were sent to the general public to gather information effectively. Documentations were

thoroughly assessed, prototyping was neatly done before the analysis of functional, non-functional conditions of the system to be implemented.

5. Sequence diagrams, class diagrams, high-level architecture diagrams, model designs, wireframes that define the design and the architecture of Aayu, were included in the design chapter.

6. In the implementation chapter, team Aayu has justified the usage of different technologies using code snippets and screenshots. Also, a prototype for the mobile application was implemented in this stage.

7. The project Aayu was estimated by both quantitative and qualitative methods. The final product was illustrated to end-users, industry experts and domain experts. Evaluations from those parties have been included in the evaluation chapter.

4.3 Legal, social, ethical and professional issues

4.3.1 Legal ISSUES

4.3.1.1 Data set

Aayu team got a dataset with the website that presents free images of the ayurvedic plant without any legal regularities. The dataset does not present any sensitive images or copyright images that can be caught for legal concerns.

4.3.1.2 Privacy policy

Aayu application architecture can perform several privacy concerns. However, the Aayu team handles them entirely. The aayu mobile application only accesses the camera and the location while using the app. Aayu promises not to collect data from the user and that the collected location data is not to share with third-party applications.

4.3.1.3 Copyright issues

Aayu uses several research papers and different knowledge sources to improve the mobile application to the most suited level that can offer a solid experience to the user. All the document and other related sources credited to the original publishers and writers in the document.

4.3.2 Social issues

Aayu application mainly focuses on the English language in the implementation, although it can affect several people who do not understand the English language. Therefore the Aayu team decided to advance the application to show information in the main three languages in Sri Lanka (English, Sinhala, Tamil). Hence with the help of that upgrade, aayu does not have any critical social effect on society.

4.3.3 Ethical issues

Project Aayu was implemented in the most suitable ethical manner. Because there is no cracked or illegally downloaded software not used during the implementation of the aayu application. Aayu has a location showing a feature that can show the location of the Ayurvedic plant. So the geotagging feature can only access while using the app. Gathered location data only used for identifying the location of the plant. Aayu promises not to share location and user data with any other third party application.

4.3.4 Professional issues

Aayu application followed the highest professional standards that are used in the industry. The Aayu team ensured that all professional issues were accompanied all the time. Images and additional important sources were used to implement the data science component were obtained from publicly available sources. All questionnaires were informed about the project and how the aayu team used their response to improve the project.

4.4 Limitations of the research

Team Aayu was able to provide all of the promised features in the scope within the allocated time. Out of the scope, there are a few limitations in this mobile application. These limitations are, The user has to scan the plant leaf by placing the plant leaf in a clear white background. For now, when scanned, the details for the highest matching element will be displayed. Team Aayu will sort out and fix that in near future. Also, only the available types of leaves in the dataset can be scanned and identified. Other than these mentioned limitations, the mobile app is fully functioning with the provided features as on the scope and feature sections of the SRS.

4.5 Future enhancements

The second next stage of the project will extend to develop both Alexnet and Resnet models to identify the healthy and diseased plant leaves without clear white background and planned to develop the model to identify the plants that are only in the training dataset. It means if the user uploads an image of a plant leaf, that does not exist in the model training dataset, then it will reject without predicting a similar plant and as well it will extend the number of varieties of plants that can identify using the mobile application as the future enhancements of this project.

4.6 Extra work

Team Aayu selected to create a web application as the extra work of the project. The user can scan ayurvedic plants and can add them to the Aayu map. As the main feature of the Aayu web application, it has a feature to search added plants from the map page. Simply the user can enter the name and it will show the locations of the relevant plant. Sama location database API is used to create the above map section of the application.

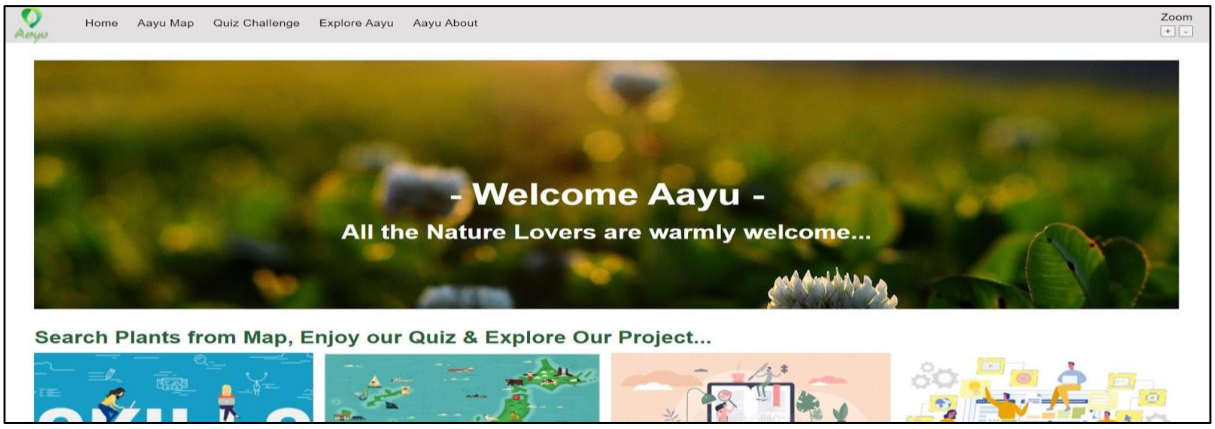

Home
Aayu Map
Quiz Challenge
Explore Aayu
Aayu About
Zoom
- Welcome Aayu -
All the Nature Lovers are warmly welcome...
Search Plants from Map, Enjoy our Quiz & Explore Our Project...

Figure 15 - Home page of the web application

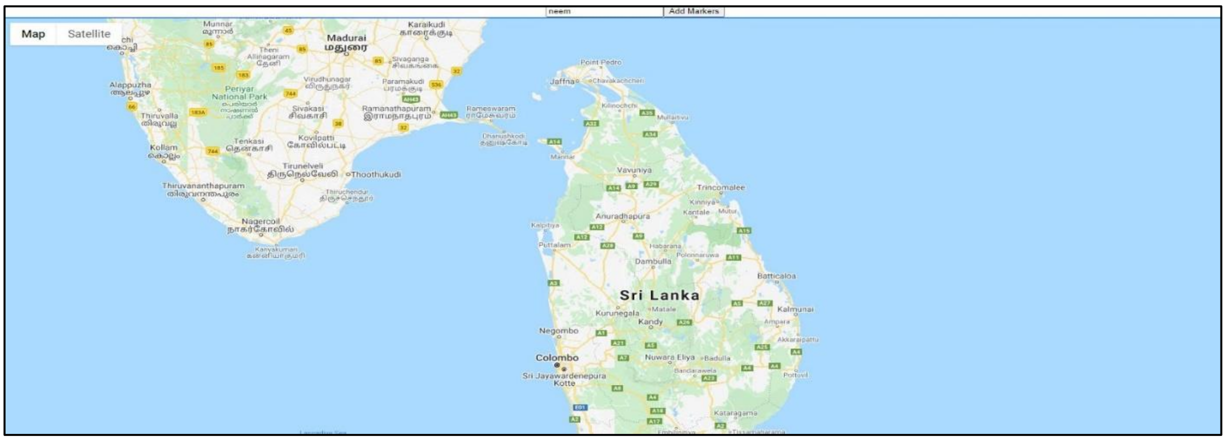
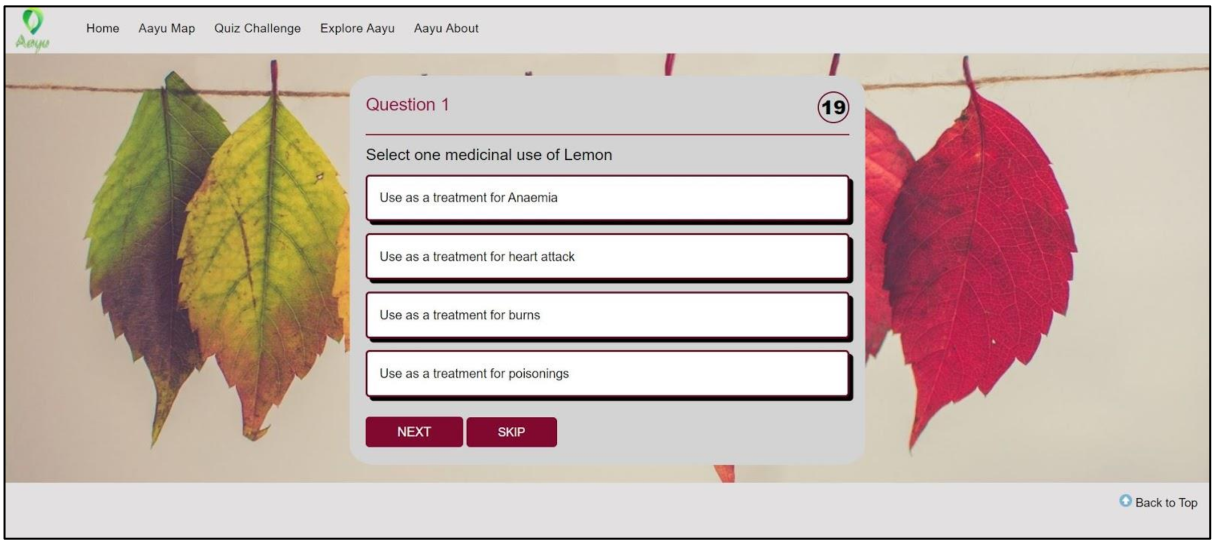


Figure 16 - Map page of the web application

Second main feature of the web application is the Quiz age. Aayu web application has an interactive quiz page with questions related to ayurvedic medicine. Users can develop his/her knowledge about ayurvedic medicine and plants by using the Quiz section.



Select one medicinal use of Lemon

Use as a treatment for Anaemia

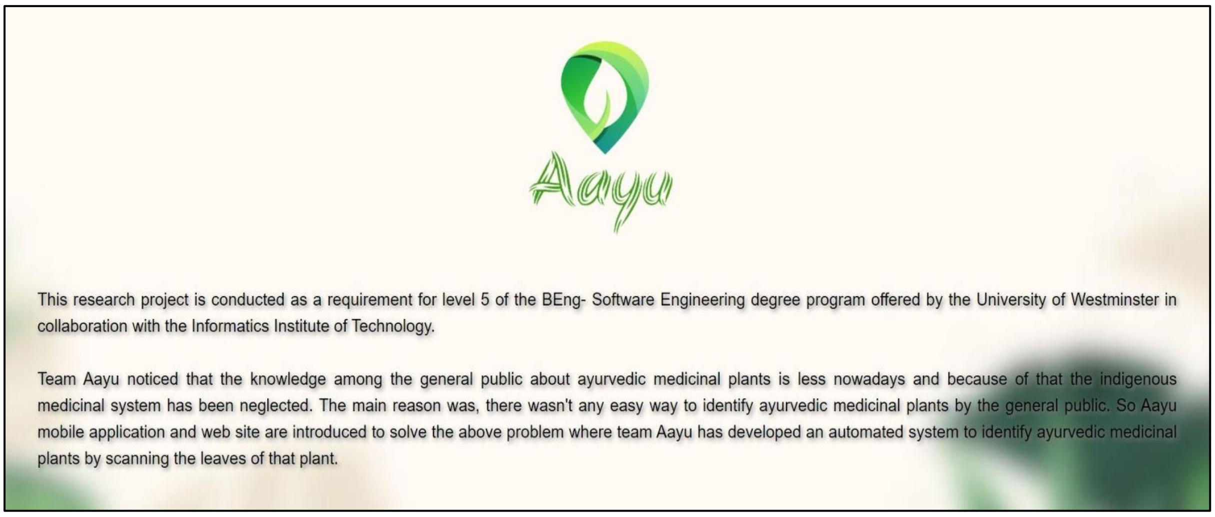
Use as a treatment for heart attack

Use as a treatment for burns

Use as a treatment for poisonings

Figure 17 - Quiz page

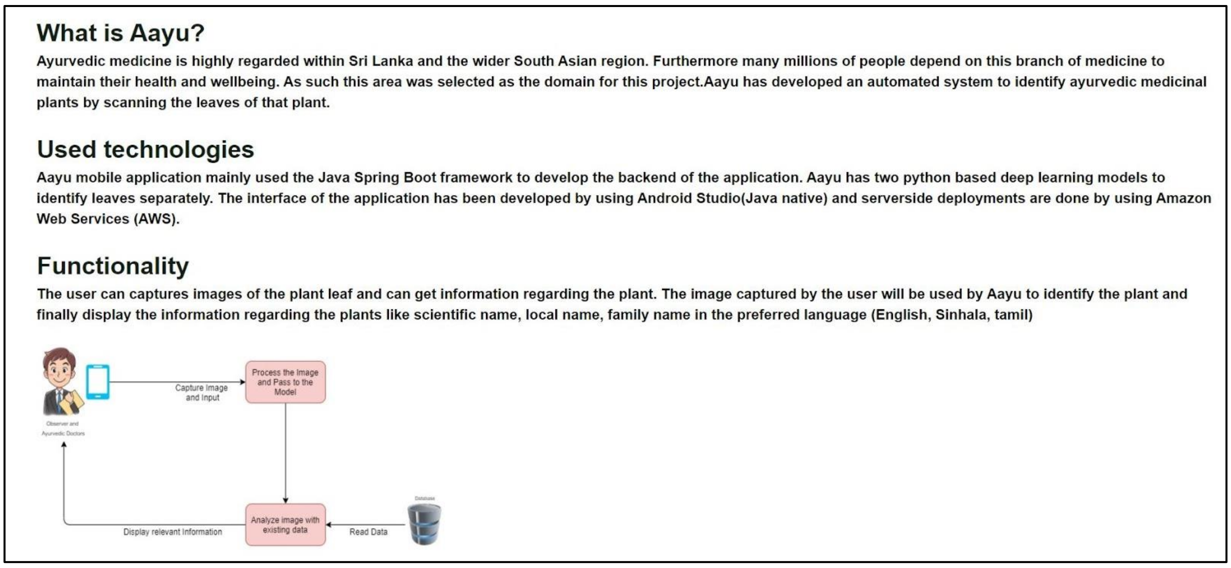
Aayu web application has "About" and "Explore" pages to give an overall knowledge to the user. Therefore, the user can get a good understanding about the project Aayu.



This research project is conducted as a requirement for level 5 of the BEng- Software Engineering degree program offered by the University of Westminster in collaboration with the Informatics Institute of Technology.

Team Aayu noticed that the knowledge among the general public about ayurvedic medicinal plants is less nowadays and because of that the indigenous medicinal system has been neglected. The main reason was, there wasn't any easy way to identify ayurvedic medicinal plants by the general public. So Aayu mobile application and web site are introduced to solve the above problem where team Aayu has developed an automated system to identify ayurvedic medicinal plants by scanning the leaves of that plant.

Figure 18 - About page



What is Aayu?

Ayurvedic medicine is highly regarded within Sri Lanka and the wider South Asian region. Furthermore many millions of people depend on this branch of medicine to maintain their health and wellbeing. As such this area was selected as the domain for this project.Aayu has developed an automated system to identify ayurvedic medicinal plants by scanning the leaves of that plant.

Used technologies

Aayu mobile application mainly used the Java Spring Boot framework to develop the backend of the application. Aayu has two python based deep learning models to identify leaves separately. The interface of the application has been developed by using Android Studio(Java native) and serverside deployments are done by using Amazon Web Services (AWS).

Functionality

The user can captures images of the plant leaf and can get information regarding the plant. The image captured by the user will be used by Aayu to identify the plant and finally display the information regarding the plants like scientific name, local name, family name in the preferred language (English, Sinhala, tamil)

Observer and Ayurvedic Doctors
Capture Image and Input
Process the Image and Pass to the Model
Database
Read Data
Analyze image with existing data
Display relevant Information

Figure 19 - Aayu explore page

4.7 Concluding remarks

From the inception of this project, the aim has always been to create an application that would help users to identify ayurvedic plants using modern technology in an accurate and user-friendly manner, thus helping to promote the knowledge and popularity of ayurvedic medicine. While there were some limitations, back at the solution that has been created, the team is confident that they have achieved the goal which they set for themselves.

References

K. R. Jackson et al., "Performance Analysis of High Performance Computing Applications on the Amazon Web Services Cloud," 2010 IEEE Second International Conference on Cloud Computing Technology and Science, 2010, pp. 159-168, doi: 10.1109/CloudCom.2010.69.
Dileep, M. and Pournami, P., 2019. AyurLeaf: A Deep Learning Approach for Classification of Medicinal Plants. TENCON 2019 - 2019 IEEE Region 10 Conference (TENCON).

W. Zhou, L. Li, M. Luo and W. Chou, "REST API Design Patterns for SDN Northbound API," 2014 28th International Conference on Advanced Information Networking and Applications Workshops, 2014, pp. 358-365.

Bibliography

Brownlee, J., 2021. Your First Machine Learning Project in Python Step-By-Step. [online] Machine Learning Mastery. Available at: <https://machinelearningmastery.com/machine-learning-in-python-step-by-step/> [Accessed 3 February 2021].

Developer.mozilla.org. 2021. SVG: Scalable Vector Graphics | MDN. [online] Available at: <https://developer.mozilla.org/en-US/docs/Web/SVG> [Accessed 5 February 2021].

Dl.ai. 2021. 7.1. Deep Convolutional Neural Networks (AlexNet) — Dive into Deep Learning 0.16.1 documentation. [online] Available at: <https://d2l.ai/chapter_convolutional-modern/alexnet.html> [Accessed 3 February 2021].

GeeksforGeeks. 2021. Residual Networks (ResNet) - Deep Learning - GeeksforGeeks. [online] Available at: <https://www.geeksforgeeks.org/residual-networks-resnet-deep-learning/> [Accessed 5 February 2021].

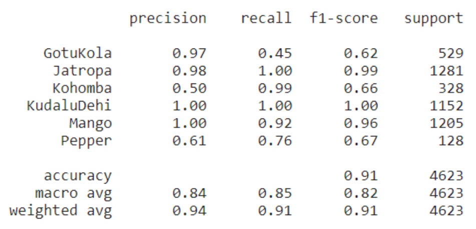
Medium. 2020. Alexnet: The Architecture That Challenged Cnns. [online] Available at: <https://towardsdatascience.com/alexnet-the-architecture-that-challenged-cnns-e406d5297951> [Accessed 17 November 2020].

Appendix

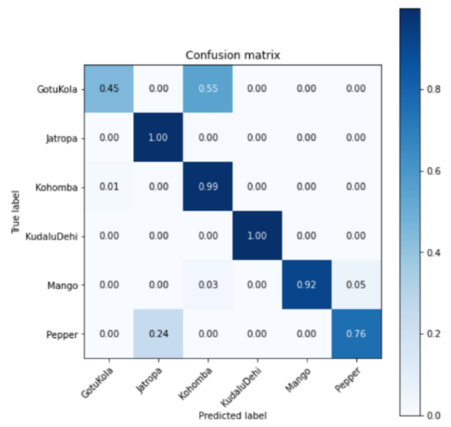
Appendix Chapter 2 - NFR

NFR 1 - Accuracy

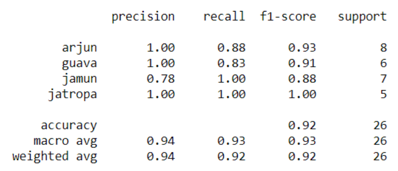
Test Case No.	Feature Tested	Description	Condition	Expected Result	Actual Result	Status
1.1	NFR 1	Alexnet - Accuracy	Multi-Class Classifier	Good Accuracy	90.67%	Pass



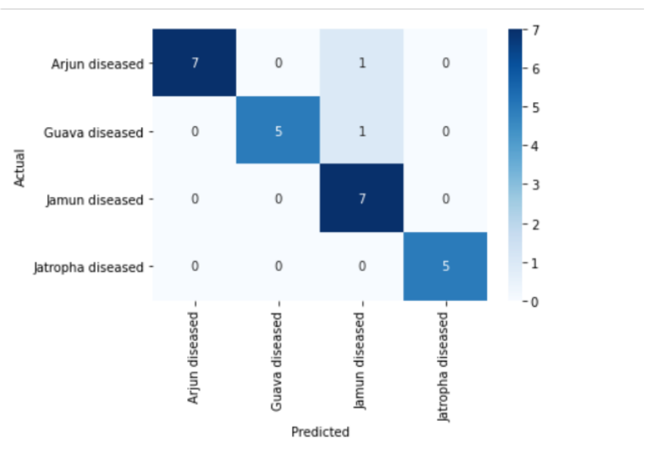
```
              precision    recall  f1-score   support

    GotuKola       0.97      0.45      0.62       529
     Jatropa       0.98      1.00      0.99      1281
     Kohomba       0.50      0.99      0.66       328
  KudaluDehi       1.00      1.00      1.00      1152
       Mango       1.00      0.92      0.96      1205
      Pepper       0.61      0.76      0.67       128

    accuracy                           0.91      4623
   macro avg       0.84      0.85      0.82      4623
weighted avg       0.94      0.91      0.91      4623
```



Test Case No.	Feature Tested	Description	Condition	Expected Result	Actual Result	Status
1.2	NFR 1	Resnet - Accuracy	Multi-Class Classifier	Good Accuracy	92.30%	Pass

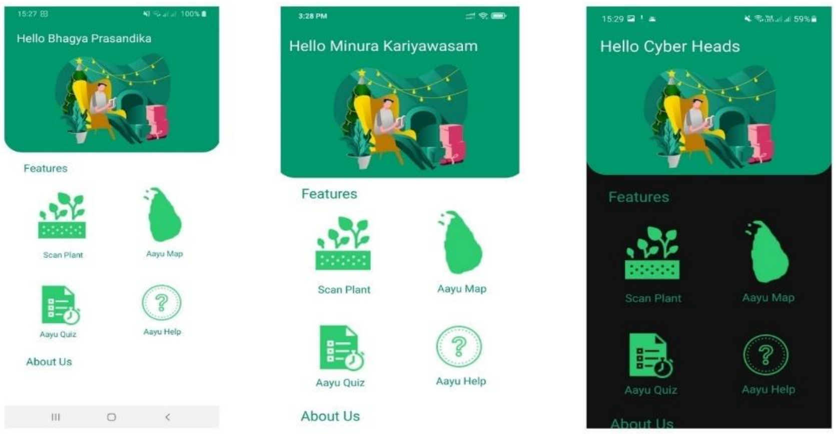


```
              precision    recall  f1-score   support

      arjun       1.00      0.88      0.93         8
      guava       1.00      0.83      0.91         6
      jamun       0.78      1.00      0.88         7
    jatropa       1.00      1.00      1.00         5

    accuracy                           0.92        26
   macro avg       0.94      0.93      0.93        26
weighted avg       0.94      0.92      0.92        26
```



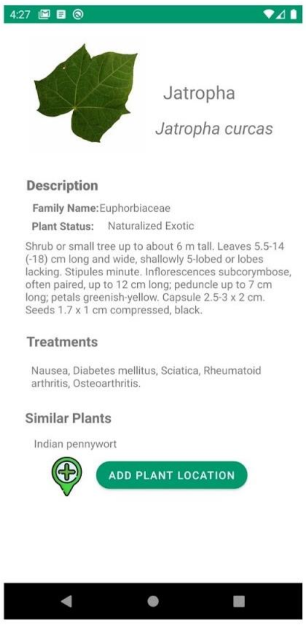
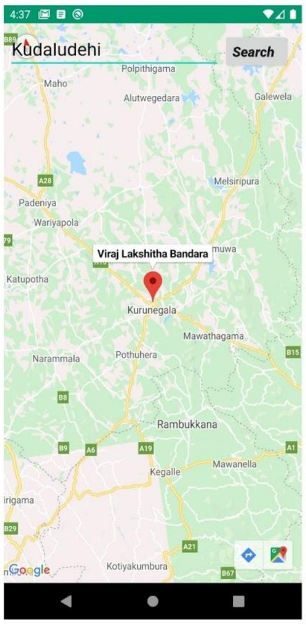
```
test accuracy: 92.3076923076923
```

NFR 2 - Performance

Test Case No.	Feature Tested	Description	Condition	Expected Result	Actual Result	Status
2.1	NFR 2	Performance	SM-A205F	Working well without lag	Working well, UI is fully responsive	Pass
2.2			POCO-F1	Working well without lag	Working well, UI is fully responsive	Pass
2.3			SM-G770F	Working well without lag	Working well, UI is fully responsive	Pass
						

NFR 3 - Reliability

Test Case No.	Feature Tested	Description	Condition	Expected Result	Actual Result	Status

3.1	NFR 3	Reliability	Identify uses of the plant	Display all the uses and other details	Display Scientific Name, English Name, Uses and etc	Pass
			Locations of plants	Display the locations of the plant	Display location of the plant and name of user added to the application	Pass
 						

4:27

Jatropha

Jatropha curcas

Description

Family Name: Euphorbiaceae

Plant Status: Naturalized Exotic

Shrub or small tree up to about 6 m tall. Leaves 5.5-14 (-18) cm long and wide, shallowly 5-lobed or lobes lacking. Stipules minute. Inflorescences subcorymbose, often paired, up to 12 cm long; peduncle up to 7 cm long; petals greenish-yellow. Capsule 2.5-3 x 2 cm. Seeds 1.7 x 1 cm compressed, black.

Treatments

Nausea, Diabetes mellitus, Sciatica, Rheumatoid arthritis, Osteoarthritis.

Similar Plants

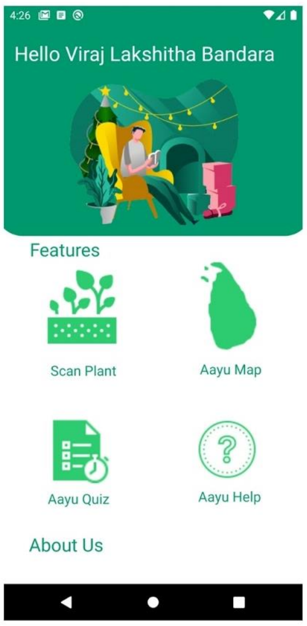
Indian pennywort

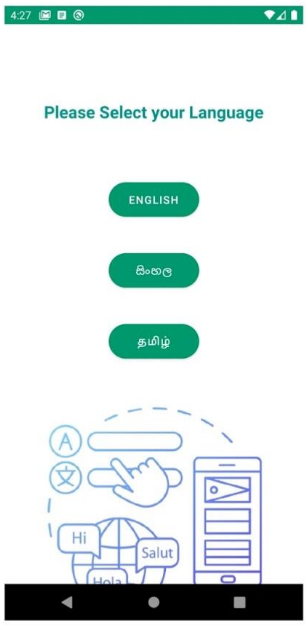
ADD PLANT LOCATION

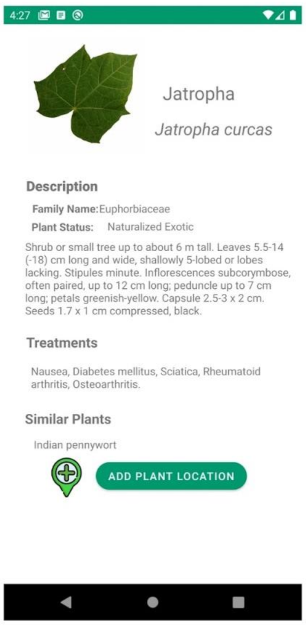
NFR 4 - Usability

Test Case No.	Feature Tested	Description	Condition	Expected Result	Actual Result	Status

4.1	NFR 4	Usability	User-friendly design	Good UI and UX	Good UI UX, Use of colourful illustrations	Pass
4.2			Responsive Design	Responsive for all android devices	Fully Responsive (> Android version 6)	Pass






Jatropha

Jatropha curcas

Description

Family Name: Euphorbiaceae

Plant Status: Naturalized Exotic

Shrub or small tree up to about 6 m tall. Leaves 5.5-14 (-18) cm long and wide, shallowly 5-lobed or lobes lacking. Stipules minute. Inflorescences subcorymbose, often paired, up to 12 cm long; peduncle up to 7 cm long; petals greenish-yellow. Capsule 2.5-3 x 2 cm. Seeds 1.7 x 1 cm compressed, black.

Treatments

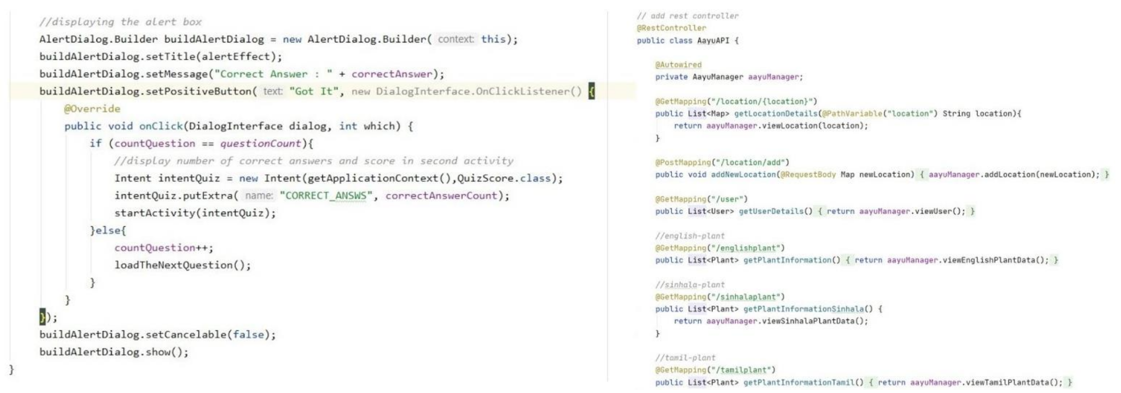
Nausea, Diabetes mellitus, Sciatica, Rheumatoid arthritis, Osteoarthritis.

Similar Plants

Indian pennywort

NFR 5 - Scalability

Test Case No.	Feature Tested	Description	Condition	Expected Result	Actual Result	Status
5.1	NFR 5	Scalability	Code comments and standards	Use code comments and good standard	Use code comments and good structure	Pass

5.2			Better file strusсture	Maintain good file structure for any update	Proper file structure and update at any time	Pass
						

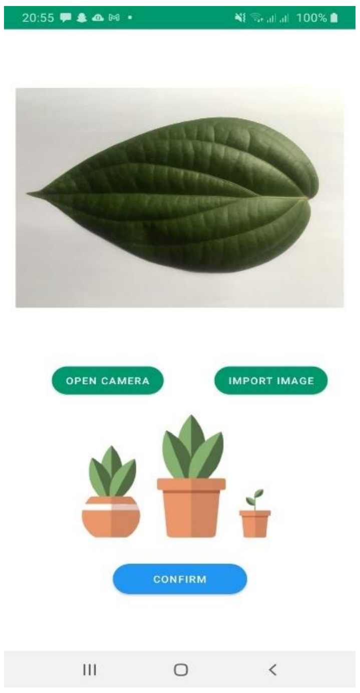
```
//displaying the alert box
AlertDialog.Builder buildAlertDialog = new AlertDialog.Builder( context: this);
buildAlertDialog.setTitle(alertEffect);
buildAlertDialog.setMessage("Correct Answer : " + correctAnswer);
buildAlertDialog.setPositiveButton( text: "Got It", new DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialog, int which) {
        if (countQuestion == questionCount){
            //display number of correct answers and score in second activity
            Intent intentQuiz = new Intent(getApplicationContext(),QuizScore.class);
            intentQuiz.putExtra( name: "CORRECT_ANSWS", correctAnswerCount);
            startActivity(intentQuiz);
        }else{
            countQuestion++;
            loadTheNextQuestion();
        }
    }
});
buildAlertDialog.setCancelable(false);
buildAlertDialog.show();
}
```

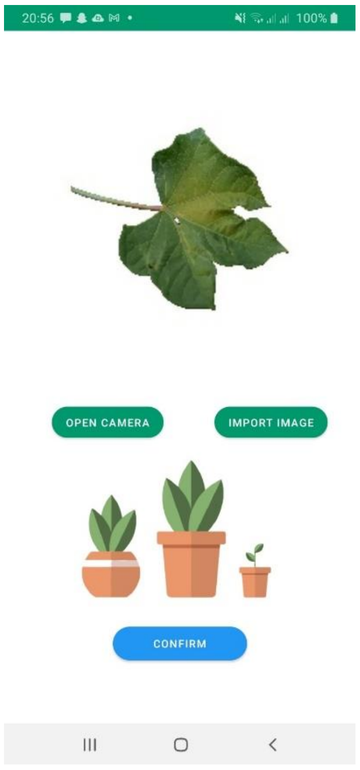
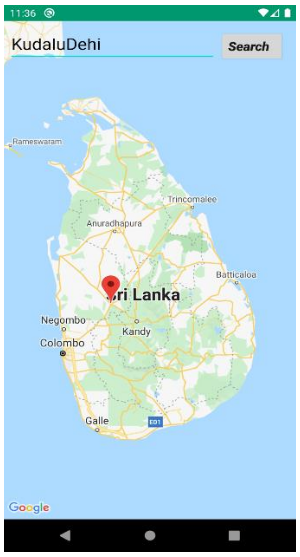
```
// add rest controller
@RestController
public class AayuAPI {

    @Autowired
    private AayuManager aayuManager;

    @GetMapping("/location/{location}")
    public List<Map> getLocationDetails(@PathVariable("location") String location){
        return aayuManager.viewLocation(location);
    }

    @PostMapping("/location/add")
    public void addNewLocation(@RequestBody Map newLocation) { aayuManager.addLocation(newLocation); }

    @GetMapping("/user")
    public List<User> getUserDetails() { return aayuManager.viewUser(); }

    //english-plant
    @GetMapping("/englishplant")
    public List<Plant> getPlantInformation() { return aayuManager.viewEnglishPlantData(); }

    //sinhala-plant
    @GetMapping("/sinhalaplant")
    public List<Plant> getPlantInformationSinhala() {
        return aayuManager.viewSinhalaPlantData();
    }

    //tamil-plant
    @GetMapping("/tamilplant")
    public List<Plant> getPlantInformationTamil() { return aayuManager.viewTamilPlantData(); }
```

Table 3: Other Non-functional requirements

Appendix Chapter 2 - Black box testing

Test Case No.	Feature Tested	Description	Condition	Expected Result	Actual Result	Status
7.1	FR 1	Scan healthy plant leaf	Using Image Capture by Camera	Capture Image using Camera	Capture Image using Camera	Pass (100%)
7.2	FR 1	Scan healthy plant leaf	Upload Image	Upload image from files	Upload image from files	Pass (100%)

						
7.3	FR 1	Scan diseased plant leaf	Using Image Capture by Camera	Capture Image using Camera	Capture Image using Camera	Pass (100%)
7.4	FR 1	Scan diseased plant leaf	Upload Image	Upload image from files	Upload image from files	Pass (100%)

						
7.5	FR 3	Searching the added plant from the Aayu Map	Find the location of the plant	Search Location on Map	Find the plant location	Pass (100%)
 KudaluDehi Search Sri Lanka Anuradhapura Trincomalee Batticaloa Negombo Kandy Colombo Galle						
7.6	FR 5	Aayu quiz	Testing knowledge of plants	Testing knowledge using Quiz	Test knowledge level	Pass (100%)

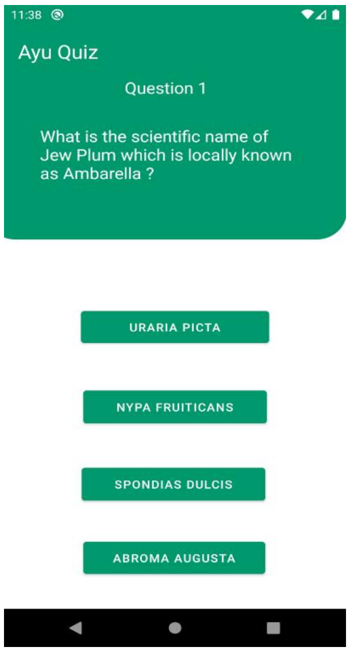
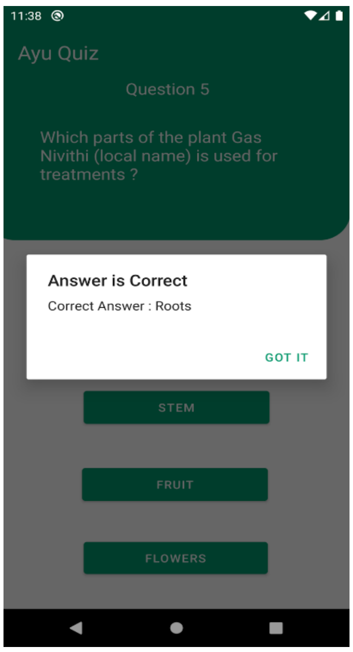
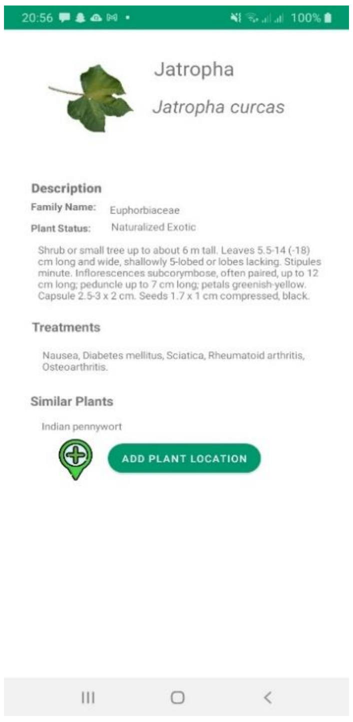
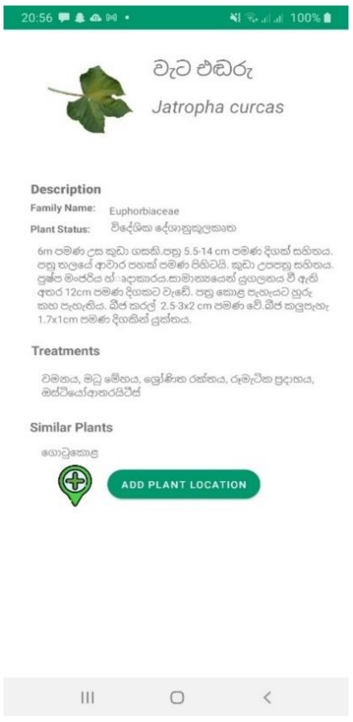
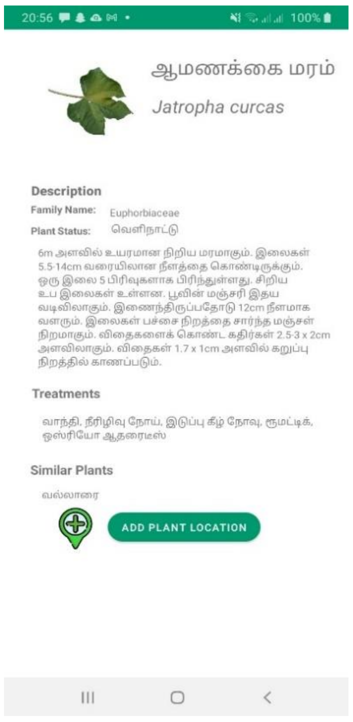
 11:38 Ayu Quiz Question 1 What is the scientific name of Jew Plum which is locally known as Ambarella ? URARIA PICTA NYPA FRUITICANS SPONDIAS DULCIS ABROMA AUGUSTA  11:38 Ayu Quiz Question 5 Which parts of the plant Gas Nivithi (local name) is used for treatments ? Answer is Correct Correct Answer : Roots GOT IT STEM FRUIT FLOWERS						
7.7	FR 4	Tri Language Selector	Plant Details in Tri Languages	Display all details of the plants in all languages	Identify and display details in tri-languages	Pass (100%)
 20:56 100% Jatropha *Jatropha curcas* Description Family Name: Euphorbiaceae Plant Status: Naturalized Exotic Shrub or small tree up to about 6 m tall. Leaves 5.5-14 (-18) cm long and wide, shallowly 5-lobed or lobes lacking. Stipules minute. Inflorescences subcorymbose, often paired, up to 12 cm long; peduncle up to 7 cm long; petals greenish-yellow. Capsule 2.5-3 x 2 cm. Seeds 1.7 x 1 cm compressed, black. Treatments Nausea, Diabetes mellitus, Sciatica, Rheumatoid arthritis, Osteoarthritis. Similar Plants Indian pennywort ADD PLANT LOCATION  20:56 100% වැට එඬරු *Jatropha curcas* Description Family Name: Euphorbiaceae Plant Status: විදේශික දේශානුකූලකෘත 6m පමණ උස කුඩා ගසකි.පත්‍ර 5.5-14 cm පමණ දිගක් සහිතය. පත්‍ර තලයේ ආවාර පහක් පමණ පිහිටයි. කුඩා උපපත්‍ර සහිතය. පුෂ්ප මංජරිය හෘදාකාරය.සාමාන්‍යයෙන් යුගලනය වී ඇති අතර 12cm පමණ දිගකට වැඩේ. පත්‍ර කොළ පැහැයට හුරු කහ පැහැතිය. බීජ කරල් 2.5-3x2 cm පමණ වේ.බීජ කලුපැහැ 1.7x1cm පමණ දිගකින් යුක්තය. Treatments වමනය, මධු මේහය, ශ්‍රෝණිත රක්තය, රූමැටික ප්‍රදාහය, ඔස්ටියෝආතරයිටිස් Similar Plants ගොටුකොළ ADD PLANT LOCATION  20:56 100% ஆமணக்கை மரம் *Jatropha curcas* Description Family Name: Euphorbiaceae Plant Status: வெளிநாட்டு 6m அளவில் உயரமான நிறிய மரமாகும். இலைகள் 5.5-14cm வரையிலான நீளத்தை கொண்டிருக்கும். ஒரு இலை 5 பிரிவுகளாக பிரிந்துள்ளது. சிறிய உப இலைகள் உள்ளன. பூவின் மஞ்சரி இதய வடிவிலாகும். இணைந்திருப்பதோடு 12cm நீளமாக வளரும். இலைகள் பச்சை நிறத்தை சார்ந்த மஞ்சள் நிறமாகும். விதைகளைக் கொண்ட கதிர்கள் 2.5-3 x 2cm அளவிலாகும். விதைகள் 1.7 x 1cm அளவில் கறுப்பு நிறத்தில் காணப்படும். Treatments வாந்தி, நீரிழிவு நோய், இடுப்பு கீழ் நோவு, ரூமட்டிக், ஒஸ்ரியோ ஆதரைடீஸ் Similar Plants வல்லாரை ADD PLANT LOCATION						

Table 4: Black box testing