

FINAL REPORT

CSCE 606

Apr 30, 2022

Team ArecaNut Farm Informatics

Team Members

Sai Shreyashi Penugonda

Rimsha Maredia

Siri Pranitha Namburi

Sindura Saraswathi

Feras Khemakhem

1 SUMMARY

In India, Areca nut is cultivated in 400,000 hectares of land with production of nearly 500k tones. It has both commercial and economical importance and is widely exported in nearby countries. Since it is spread across a wide area having multiple plots of several acres of land, it is hard to manually monitor and maintain various agricultural aspects of the farms for farmers. There are various aspects of farming which need to be monitored such as irrigation, fertigation, rain days and water available in tanks.

Our application focuses on solving this problem. We have created an application where any farmer can easily create an account and monitor farming activities. Once the user logs into the account there are various options available to enter irrigation days and frequency, fertigation days and frequency, rainfall days and tank size based on different farm names. Our application regularly sends reminders to farmers based on their input of frequency on the day of irrigation and fertigation. Our application is not only limited to arecanut farmers but also for other farmers who want to easily monitor their farms.

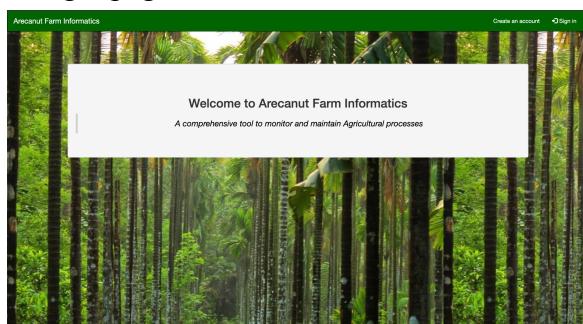
2 STORIES DESCRIPTION

2.1 ITERATION 1

- **Home page UI (*Fully Implemented*)**

Points assigned : 3 points

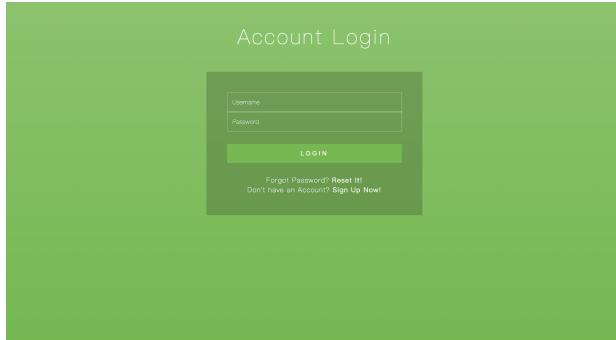
Implemented main page for the application from where users can navigate to sign up or login page.



- **Login page UI (*Fully Implemented*)**

Points assigned : 3 points

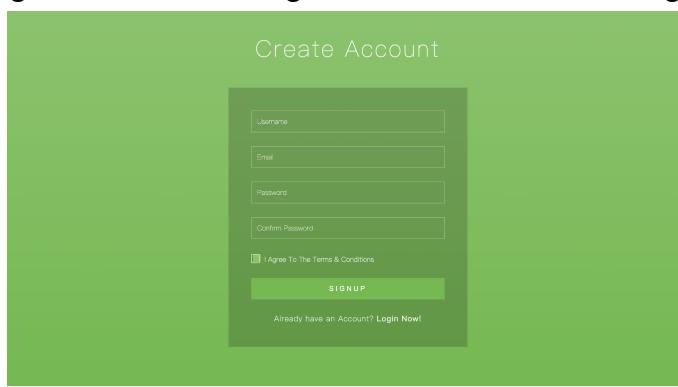
Page where user can login using user ID and password.



- **Creation of User Account UI (*Fully Implemented*)**

Points assigned : 3 points

Page where users can register an account for future logging into the application.



- **Storing User Data (*Fully Implemented*)**

Points assigned : 3 points

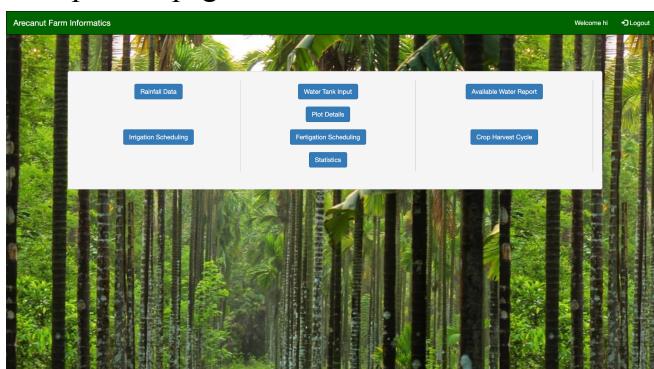
Created and connected AWS Sql database to store and fetch user credentials.

2.2 ITERATION 2

- Landing Page: Post Login (Fully Implemented)

Points assigned : 2 points

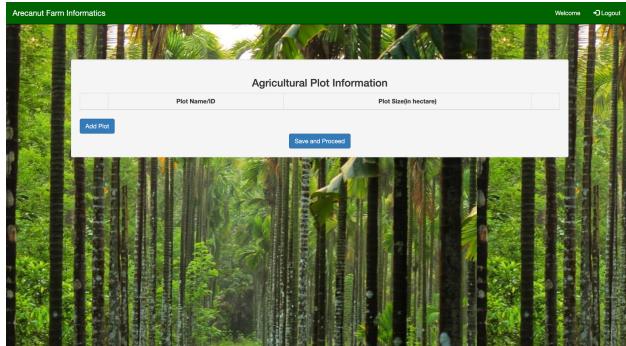
Main landing page after logging into the account. It displays all the buttons to navigate to respective pages.



- Plot Input (**Fully Implemented**)

Points assigned : 3 points

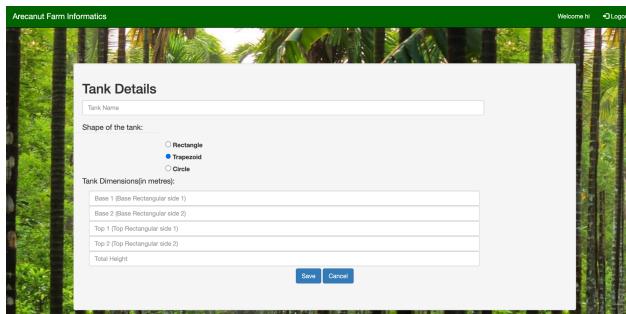
Form where user can view current plot's information and add more plots using name, plot size as parameters to do calculations for future analysis.



- **Tank Shape Input (*Fully Implemented*)**

Points assigned : 3 points

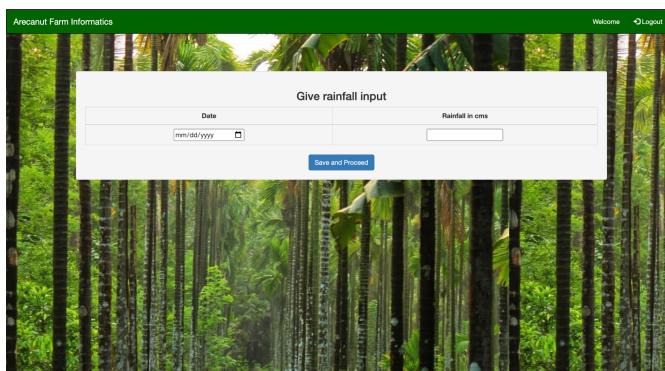
Form to add tank details including tank shape(radio button choice) and measurements to calculate volume of water stored.



- **Rainfall Data input (*Fully Implemented*)**

Points assigned : 3 points

Created page to capture rainfall data to future analysis.



- **Automating Irrigation Schedule: User Input (*Fully Implemented*)**

Points assigned : 2 points

Page where users can enter time, start date and frequency to get reminders based on schedule.

2.3 ITERATION 3

- **Implement Flask-mail using smtplib for scheduling email and algorithm to get correct date to send email (*Fully Implemented*)**

Points assigned : 3 points

Set up flask-mail to send emails to the users.

- **Validation of Input data: Irrigation Schedule input (*Fully Implemented*)**

Points assigned : 1 points

Added validation to avoid the use of same plot names for more than one plots.

- **Display Existing Plots (*Fully Implemented*)**

Points assigned : 2 points

Created page to display existing plots by fetching data from users.

The screenshot shows a web application interface titled "Arecanut Farm Informatics". At the top right, there are "Welcome hi" and "Logout" buttons. The main content area has a title "Agricultural Plot Information". Below it is a form with two fields: "Plot Name/ID" containing "1" and "ram" in a text input, and "Plot Size(in hectare)" containing "42069" in another text input. To the right of these inputs is a red "Remove" button. At the bottom left of the form is a blue "Add Plot" button, and at the bottom right is a blue "Save and Proceed" button. The background of the page features a photograph of a dense forest of palm trees.

- **Connect to DB: Tank Input (*Fully Implemented*)**

Points assigned : 1 points

Store tank input data in database

- **Validation of Input data(Rainfall Input) (*Fully Implemented*)**

Points assigned : 1 points

Added validation to rainfall input page to avoid users from adding dates from past and to ensure that rainfall input is numeric.

- **Display Existing irrigation Schedule (*Fully Implemented*)**

Points assigned : 2 points

Added page to display the existing schedule to display and verify the schedule.

The screenshot shows a web application interface titled "Arecanut Farm Informatics". At the top right, there are "Welcome hi" and "Logout" buttons. The main content area has a title "Set Irrigation Schedule". Below it is a form with four fields: "Plot Name/ID" containing "ram ranch", "Start Date" showing "03/27/2021", "Start Time(HH:MM AM/PM)" showing "07:30:01 PM", and "End Time(HH:MM AM/PM)" showing "07:30:02 PM". To the right of these inputs is a field "Frequency in Number of Days" containing "1". At the bottom left of the form is a blue "Save and Proceed" button. The background of the page features a photograph of a dense forest of palm trees.

2.4 ITERATION 4

- **Create email template (*Fully Implemented*)**

Points assigned : 2 points

Created a proper HTML template for sending reminders to the users



- **Create Cron Job (*Fully Implemented*)**

Points assigned : 1 points

Created a cron job to run schedule.py to send emails to the users

- **Unit test Coverage (*Fully Implemented*)**

Points assigned : 3 points

Conducted unit test coverage for all pages.

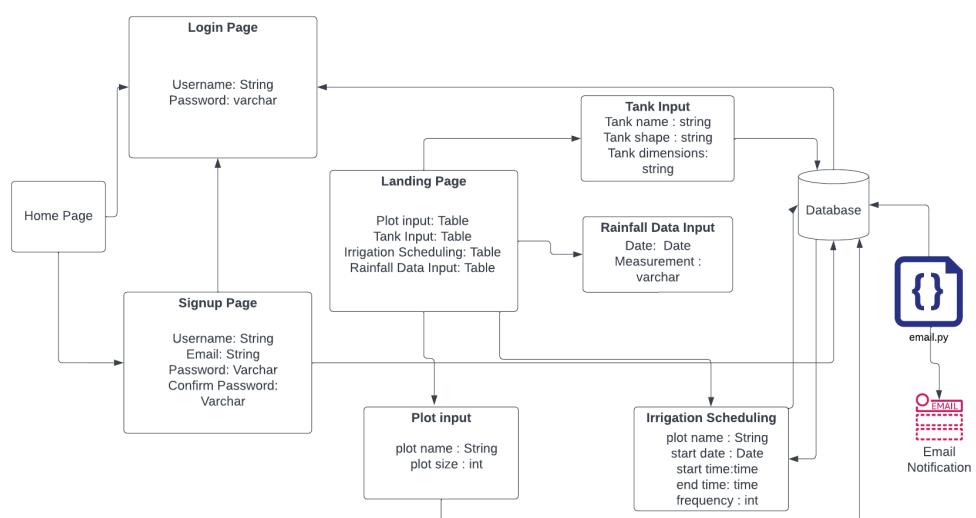
- **Tank Input UI fixes, DB connection and storing data, server-side validations**

Points assigned (*Fully Implemented*)

Points assigned : 3 points

Fixed UI for some pages, made changes to tables in DB and performed server side validations.

Final UML Diagram



3 Team Roles

Scrum Master : Rimsha Maredia

Product Owner : Sundura Saraswathi

Developers : Feras Khemakhem, Siri Pranitha Namburi, Sai Shreyashi Penugonda

While the roles were taken on by the team mates, the roles were also flexibly managed by the other members in each iteration in order to gain experience of such roles through this project.

4 Tasks done in each Iterations

- **Iteration 1** : For the first iteration we created the main landing page, login and signup page and created an AWS database and connected it to the application.
- **Iteration 2** : For the second iteration we created a landing page which can be viewed once the user logs into the account. We created respective navigation pages for every functionality. Lastly, we performed testing for the login and signup page.
- **Iteration 3** : For iteration 3 we created tables for each page and wrote functionality to store and fetch data back and forth. We also customize the UI to make it look more appealing to the users. We also wrote a separate py file to send emails using script to the users. Lastly we added validation to a few pages with forms.
- **Iteration 4** : We wrote functionality to calculate if today is the day of irrigation and automatically send reminder emails to the users on the day of irrigation. Secondly we performed unit testing for the database and all the pages. Lastly we fixed some backend bugs and added validation on remaining pages.

5 Meeting Description

Meeting 1: 9:30 AM CST : March 12 2022 - Discussed project approach and client expectations.

Meeting 2: 9:30 AM CST : March 26, 2022 - Discussed Iteration 1 and Iteration 2 goals

Meeting 3: 11:00 AM CST : April 9, 2022 - Discussed Iteration 2 and Iteration 3 goals

Meeting 4: 11:00 AM CST : April 23, 2022 - Discussed Iteration 3 and Iteration 4 goals and approach

Meeting 5: 11:00 AM CST : April 28, 2022 - Demoed the project

6 BDD/TDD process

We have used both BDD and TDD for testing the code. For functions like email functionality and DB-end-to-end testing, we have used TDD and was done by individual developers. For other pages we used the BDD approach for testing all the pages and forms.

7 Configuration management approach

Spikes: As such no explicit spikes were added due to the elaborate discussion and clarity from the customer. Throughout the team meetings and customer meetings any requirement upon discussion was added as a story in the Icebox and later moved to the iteration once it started.

Branches: Main, develop and few test branches when some individual changes for experimentation were carried out.

Releases: We had releases every iteration(4 major) after meeting up with the team prior to deployment in order to verify the status of development tasks' completion followed by merging the develop branch to main branch in order to deploy from it.

8 Heroku Issues

Database Environment Variables: The environment variables such as the port number and hostname were clashing with the system/heroku internal environment variables and causing errors. Once the variable names were modified the issue was fixed.

Cron Job Setup: Scheduling emails is designed to be done at certain parts of the day on certain days, and figuring out how to set up another dyno to run at certain times caused some issues. We resolved this by setting up a cron job dyno on a clock process.

9 CodeClimate

We used CodeClimate in order to receive a code quality report of our project and it helped us gain more insight on the coding practices and standards.

10 Important Links and Details

Pivotal Tracker: <https://www.pivotaltracker.com/projects/2555651>

Github: <https://github.com/feraskhemakhem/ArecanutFarmInformatics>

Heroku deployed app: <http://arecanutfarminformatics.herokuapp.com/>

Official Email Id: agrihelp2022@gmail.com

Poster Demo Video: <https://youtu.be/bww8pZqY1W4>