

An-Najah National University



Faculty of Engineering and Information
Technology

Computer Engineering Department

Software Graduation Project



Farmable

PREPARED BY:

**Asala Tibi
Hala Barqawi**

SUPERVISED BY:

Dr. Raed Qadi

Presented in practical fulfillment of the requirements of
Bachelor degree in Computer Engineering.

MAY, 2022

Acknowledgements

We would like to thank our supervisor, Dr. Raed Qadi , for his helpful advice and knowledge of this project. His high standards have made us better at what we do.

Also, we would like to thank all our friends and family for helping us survive all the stress from this year and not letting us give up

Abstract

Farm management is considered as an essential procedure for all farmers. However, the majority of farmers lack the ability to manage their farms. Therefore, our application and website play a role in helping the farmers to gain full control over their crops by allowing them to log all their jobs and record the harvest time. In addition, it can gather data easily and precisely. It can also track and organize the various operations that will be performed on the land, such as seed planting , irrigation, fertilization and harvesting.

The collected data can be used to generate reports by a few clicks and this will save time. Moreover, it will help the farmers to communicate with each other.

In short, our application and website will effectively organize and manage farmers' work by utilizing new technology in management.

Contents

1	Introduction	8
1.1	Problem	8
1.2	Objectives	8
1.3	Scope	9
1.4	Importance	9
2	Constraints, Standards/Codes and Earlier course work	11
2.1	Constraints	11
2.2	Standards/Codes	11
2.3	Earlier course work	12
3	Literture Review	13
4	Methodology	15
4.1	Languages and Frameworks	15
4.1.1	Back-End Technologies	15
4.1.2	Back-End-URL Structure	15
4.1.3	Front-End	16
4.2	UML Digrams	17
4.2.1	Class Diagram	17
4.3	Dataset	18
4.4	Development process	18
4.5	Mobile Application	20
4.6	Admin -manager- Side	20
4.7	Worker Side	29
4.8	Web Application	31
4.8.1	Admin Login	31
4.8.2	Admin site details	32

5 Conclusion/ Recommendations and Future Works	40
5.1 Conclusions	40
5.2 Future Work	40

List of Figures

1	FrameWork	15
2	MVC, express js	16
3	Class Diagram	17
4	Starting page	20
5	Starting page	20
6	Starting page	21
7	Login Page	21
8	Sign up	21
9	Add page	21
10	Add Field	22
11	Fields Map	22
12	Fields Map	23
13	Add page	23
14	All Task Display	24
15	Completed Task	24
16	Pending Tasks	24
17	Equipment Display	25
18	Equipment Edit	25
19	Crops Display	26
20	Crops task done on it	26
21	Harvested Fields	26
22	Harvest Details	26
23	Expense Categories	27
24	Add Expense	27
25	Expense Display for specific season	28
26	Add Expense to this open expense category.	28
27	Display Sales.	29
28	Tasks delegated for this worker user.	30

29	Chat between workers.	30
30	Sign in with wrong password.	31
31	Sign in Successfully	31
32	Dashboard	32
33	Dashboard with pie charts	32
34	Display existing crops data	33
35	Display existing worker data	33
36	Adding new Worker.	34
37	Display assigned Tasks data	34
38	Features can be done on Task table	35
39	filtering, sort,...etc	35
40	search , contain...etc	36
41	fields location appear on map	37
42	Expense Table	38
43	Sales Table	38
44	Example of one of Adding screen -Expense -.	39

Chapter 1

1 Introduction

This chapter will shed light on the difficulties that our project solved. It will also explain our main objectives, scope, and the importance of this platform.

1.1 Problem

Farmers must work long hours and be on-site for the majority of the day. They must keep track of things like planting, seeding, harvesting, and weeding to ensure that their crops grow well. These farmers have very little time off because they must constantly monitor their crops with a keen eye in order to avoid problems. To accomplish this, a helper application is required to log and organize all tasks for each worker on the farm, allowing the manager-farmer- to have a complete overview of what happened in his fields. Furthermore, all sales and expenses for all seasons are archived in one location. There is no application in our surrounding region that offers such assistance in farm management .

1.2 Objectives

This research aims to make farm management more effective for farmers who lack management skills or knowledge. It also focuses on farmers who have more fields to

keep track and control on them . This application comes with many features for the farmers, including the ability for the manager to assign a specific task to any of the workers. That task will be attached to a submission date to obligate the worker to do it on time . On the other side, the manager tracks the status of the task .

Moreover, Farmers keep track of the expenses that occur during each planting season, as well as controlling the sales of available harvest and deciding which to sell. The field location which is added by admin will be available on the Map . Every worker can see the Map to know the field location for a specified task .Farmers who work for the same manager can chat together .

1.3 Scope

This platform was built mainly in (Node js in the Back-End, Mongodb as Database, MVC (Model,View ,Controller) architecture, and the Front-End was built with React native for mo- bile and React for Admin-panel website.Express.js framework for building RESTful APIs with Node.js.

At the end, the distributed version control and source code management functionality Git/Github in teamwork is the best choice for communication and making progress in building this platform.

1.4 Importance

The value of our project is that it organizes farm managers' lives and saves their time. Furthermore, it aims to assist them in managing and organizing anything at any time.

It also seeks to reduce the use of paper and the traditional method of calculating revenues and replace them with new management technology.

Chapter 2

2 Constraints, Standards/Codes and Earlier course work

2.1 Constraints

The most significant limitation we experienced was in relation to specifying fields on maps due to a lack of geographical maps that specialize in farming land and accurately detect land boundaries. The best solution was to specify the general location of the field on the map, similar to how Google Maps works. That helped us save time. It was also beneficial in assisting workers in determining the location of the field.

2.2 Standards/Codes

Most companies have standards that they must follow in order to include any feature in their product. Therefore, in Farmable, we continue to add any possible improvement in the development process itself, such as Code Quality, Clean Code, and DesignPatterns. First, we gathered the requirements and created the UML in Draw.io. Then, we assigned tasks to each member of our team and discussed each one. Following that, we selected a framework that corresponded to our requirements and system infrastructure.

Then, we used version control (Git/Github) to make our

work as professional as possible. When it comes to code, there are many recommended standards and best practices which may differ depending on your language of choice.

2.3 Earlier course work

At the beginning of semester, we had no knowledge of the front-end language that we used, which is the react for website and react native for mobile application. Therefore, we took an udemy course in addition to youtube videos and the language documentation [2] ,[3] on the internet . We also attempted to apply our cumulative course work, extra curriculum ,and training.For example , DOS the course project aided us greatly in back-end and in building REST-Ful Api using express js. Advanced Software also assisted us because of our experience in using MVC architecture. Moreover, many others, such as web development and databases, provide us with a solid base for building and working on projects in teams..

Chapter 3

3 Literature Review

In this project the main aim was to reduce the stress of management farmers by implementing a helping functionalities such as track workers task , track expenses and sales , map fields , check available equipment , have a full overview of performance of the farmers or fields ,also providing ability to chat between workers and many other facilities.

From what was mentioned in the paper by students from Purdue University,[4] The United States Department of Agriculture (USDA) recognized that a typical model for farm management does not exist because not all farms are the same size and type (Sindir, 2006). Instead, the agency acknowledged farm management has five basic 7 functions to achieve the goals and objectives of the farming business:

1. Planning: creating daily/seasonal priorities and schedules, recognizing areas needing improvement and finding alternative solutions .
2. Organizing: establishing standard operating procedures and a structure of activities to meet the farm's goals.
3. Managing human resources: recruiting, hiring, training and evaluating workers .

4. Directing: delegation of responsibilities, establishing good communication with workers .
5. Controlling: monitoring expenses and income, maintaining records of operations, comparing rates of productivity, making changes to practices as necessary, and more (Sindir, 2006).

We took those point in consideration in the phase of development.

Another paper emphasizes that Over the past 15 years however, farmers started using computers and software systems to organize their financial data and keep track of their transactions with third parties (Batte, 2005) and also monitor their crops more effectively. [1]

In General , there is a need to integrate the farmer owner to use technology in a software level which will save a lot of efforts for them .

Chapter 4

4 Methodology

This chapter will talk about the process of development that this platform goes on. By Mentioning the languages and frameworks that we use and the process of creating the mobile application and the website.

4.1 Languages and Frameworks

This section will have the main languages and frameworks that we used in our work.

4.1.1 Back-End Technologies



Figure 1: FrameWork

We chose node js because it makes it easy to get started with development. Also, because of its high performance and its language sharing across the stack. Moreover, it is asynchronous and has a huge community support.

4.1.2 Back-End-URL Structure

Building Restful Api with mvc architect using express js

which is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications

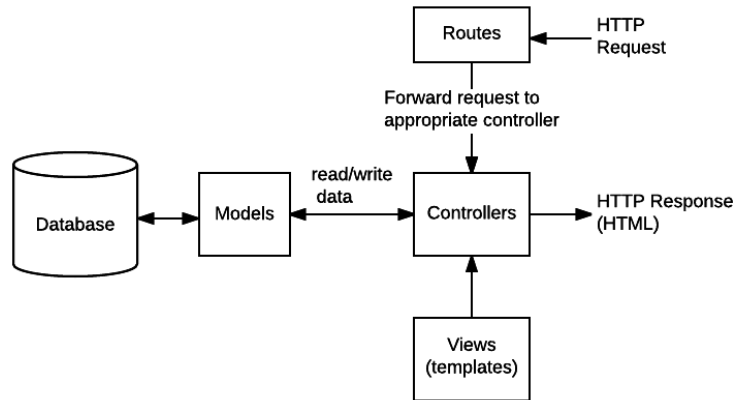


Figure 2: MVC, express js

4.1.3 Front-End

For the front-end we used React for building the website , and React-Native for mobile application .One of the reasons why we chose them is that they are highly used in the market. In addition, it helps us to easily incorporate third-party plug-ins and APIs .Moreover, its rich user interface and a lot more.

4.2 UML Digrams

4.2.1 Class Diagram

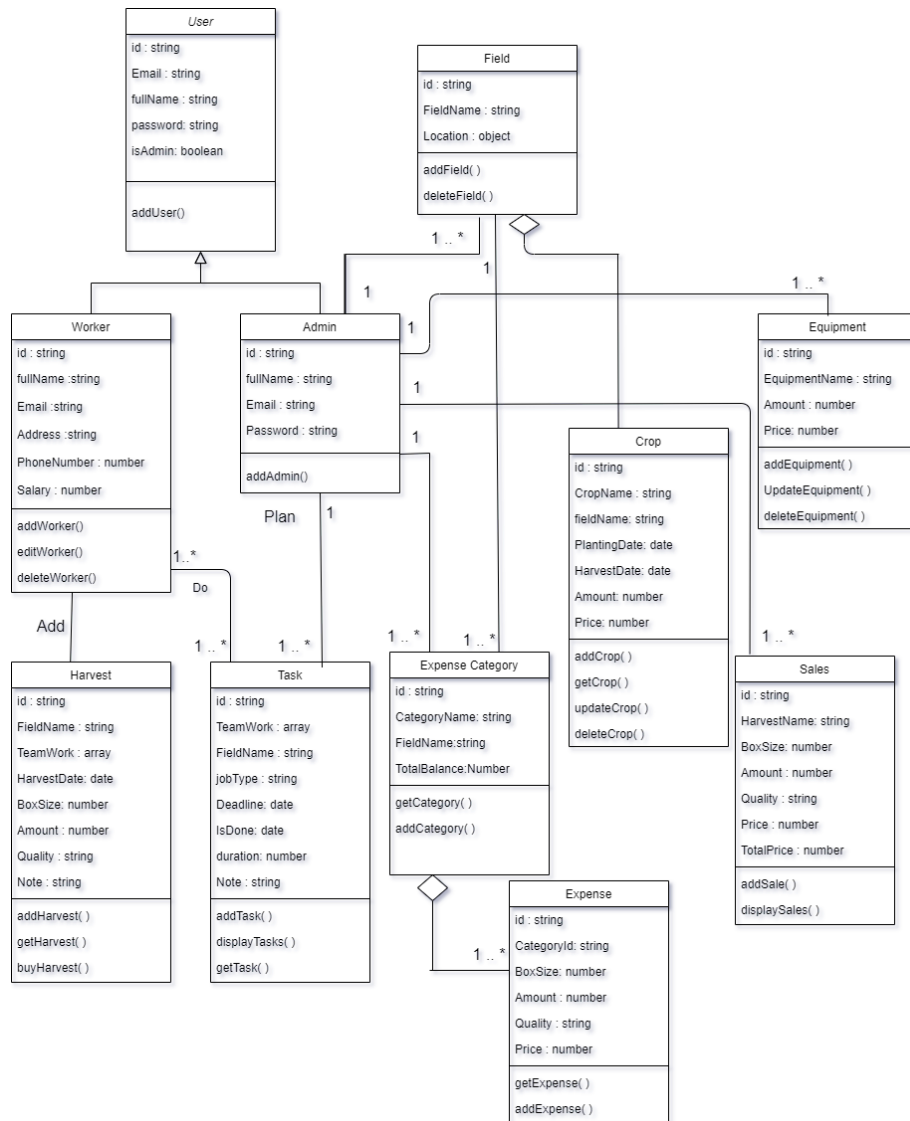


Figure 3: Class Diagram

4.3 Dataset

We chose to work with the MongoDB database, since this kind of database became popular with developers of all kinds for developing scalable applications with evolving data schemas . And has a lot of advantages like the simple installation , high performance and it is a full cloud-based developer data platform. Also , we need to use firebase database to enable chat with worker. Furthermore , we used cloudinary to store images for workers profile.

4.4 Development process

In this section we will shed light on the details of our process in building the project .

As a first step , We brainstormed problems that our society needs a software solution for. Finally, we decided to take up the idea of a farm management application which can be a turning point in farmers' lives . And also it was feasible according to our knowledge and skill set .

Second step was discovering requirements. In order to that , we searched onto the existing work in this related field. Then choosing the technology we want to use .

Software Requirements Specification (SRS):

1. Purpose: To develop a system which primarily is capable of managing and helping farmers in their daily lives.
2. Intended Audience:

- (a) Farmers who have more fields need to be managed.
 - (b) Any farmer with no management experience and wants a professional approach to manage.
3. Product Features:
- (a) Track given tasks , Specified fields locations on Map , display all planted crops and harvest ones which are ready to be sold .
 - (b) Track all expenses and sales operations.
 - (c) Chat between workers
4. User Interface Requirements:A User friendly interface so any user -farmer or worker- can deal easily with it.
5. Software Requirements: Mobile interfaces like (React Native For frontend, Node js and Express js for back-end , mongodb for database). Same for Web Interface but React for Fronted.

After enough practical learning about the software technologies that we will use .We started developing the mobile application .We started developing the front-end using react native in synchrounce with developing the back-end using node js. Integrated front end with Back-End done at the same period.Adding features continues until achieving requirements that we specified before.Web development started after a time mobile development was started.

4.5 Mobile Application

This section will show snippets of our Mobile-Platform.

4.6 Admin -manager- Side

Welcome Screens, login , sign up , Add Page screens where admin can add any of the options .

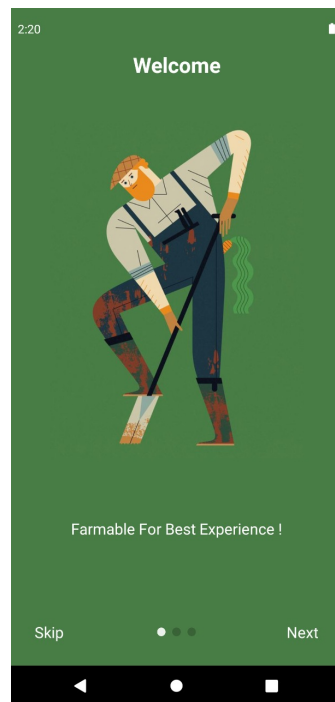


Figure 4: Starting page

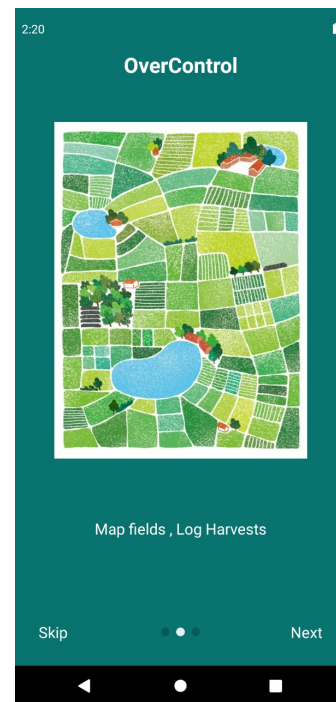


Figure 5: Starting page

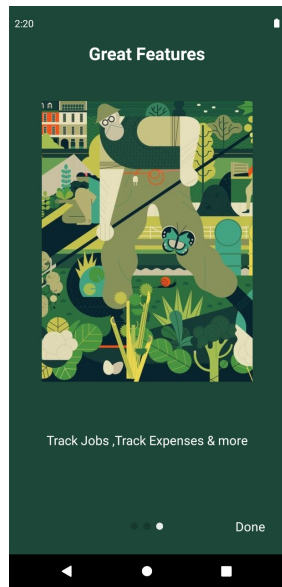


Figure 6: Starting page

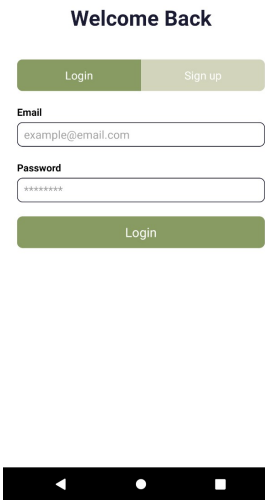


Figure 7: Login Page

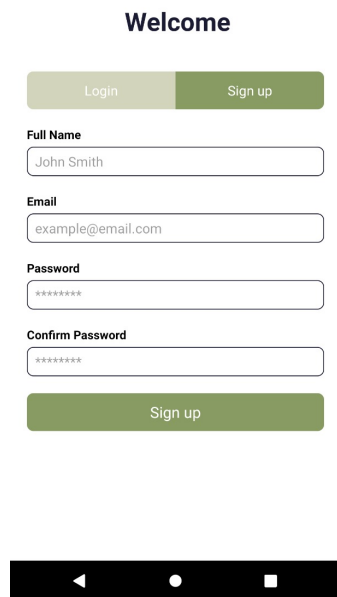


Figure 8: Sign up



Figure 9: Add page

As shown the in figure 11 map specifies the fields locations that is added by Add Field screen.



Figure 10: Add Field

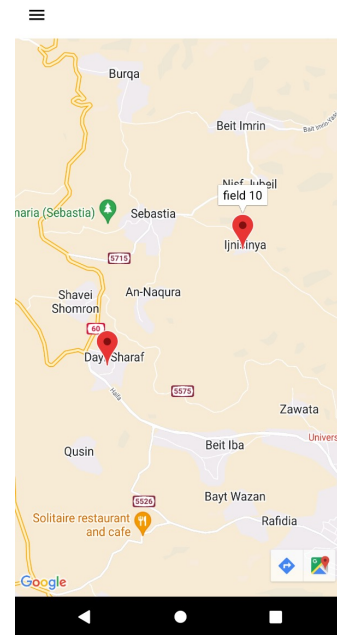


Figure 11: Fields Map

Display Workers information with ability of manager to add from addPage, edit worker information and delete worker from system.

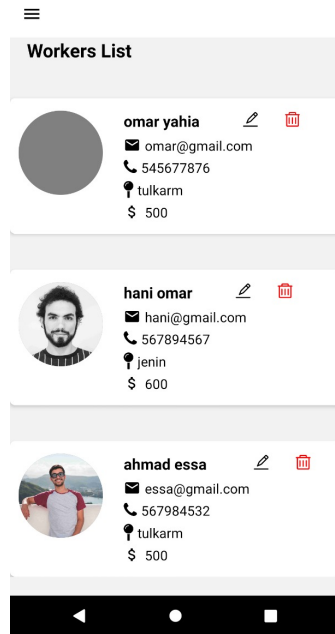


Figure 12: Fields Map

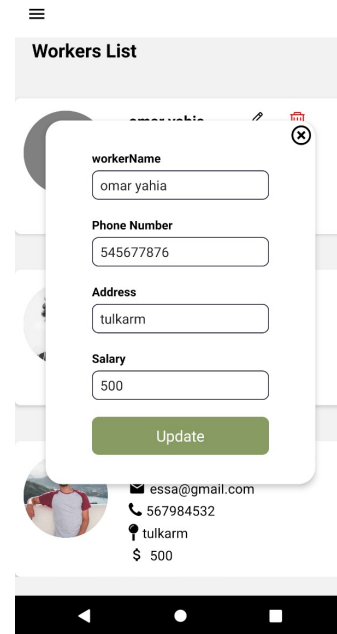


Figure 13: Add page

These figures down below show Task screen with bottom-tab to filter tasks to completed and incompleted tasks. In addition to review task details and status when it was submitted .

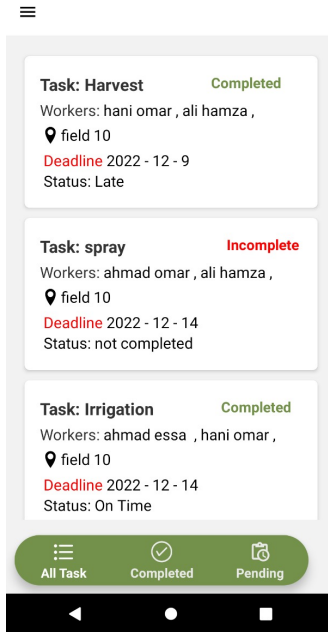


Figure 14: All Task Display

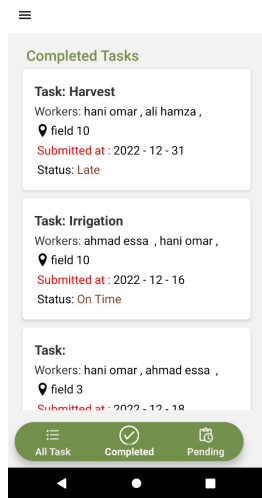


Figure 15: Completed Task

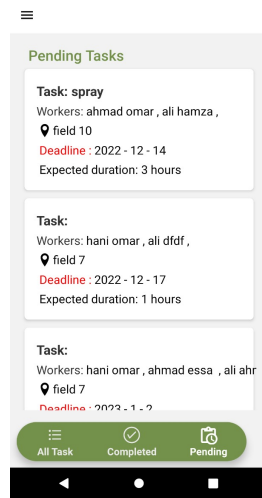


Figure 16: Pending Tasks

Display Equipment with ability to edit , add and delete

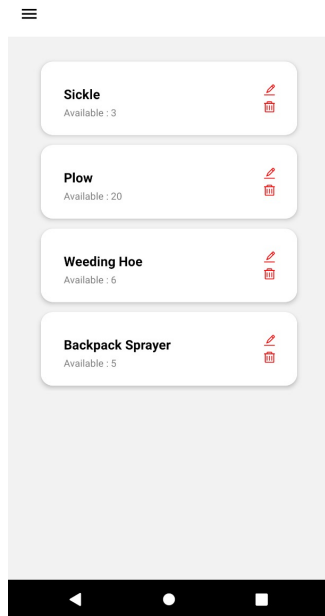


Figure 17: Equipment Display

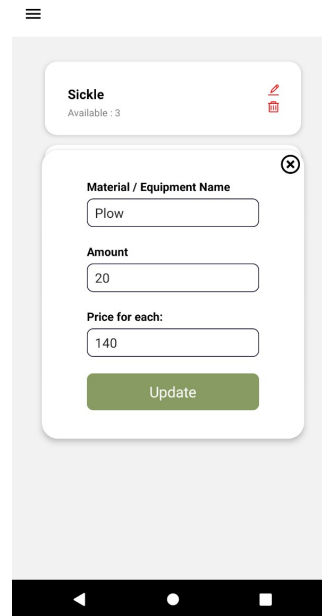


Figure 18: Equipment Edit

Display crops with ability to edit , add and delete . Furthermore, when press on crops name it will display all tasks that done on that crops. Also figures 22, 21 display the harvested field with details about it also there is a Buy button to sell the harvested crops with determined price according to market .when it sold it will be removed from harvest screens and go to sales screen as shown in figure 27 .

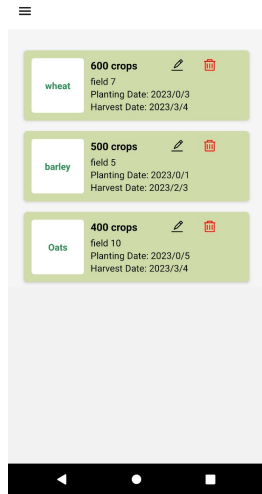


Figure 19: Crops Display

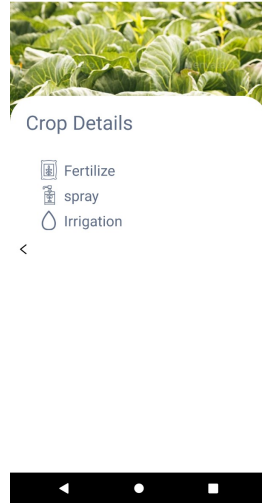


Figure 20: Crops task done on it

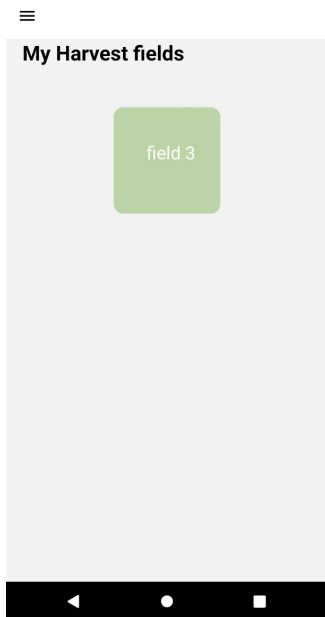


Figure 21: Harvested Fields



Figure 22: Harvest Details

Here the way manager can track expenses , by creating a category of expense for each season , to calculate incomes , expenses and total balance -Revenue- for each farming season . when for example harvested crops being sold the income will store according to the chosen season .

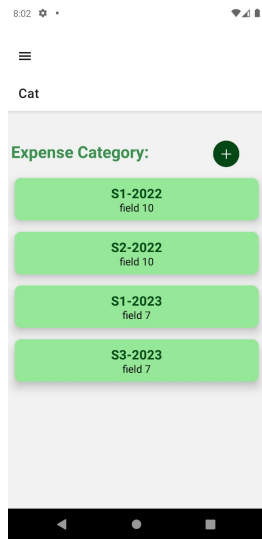


Figure 23: Expense Categories

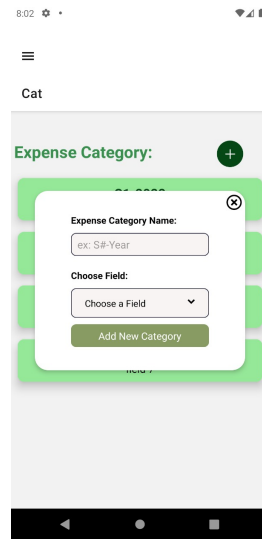


Figure 24: Add Expense

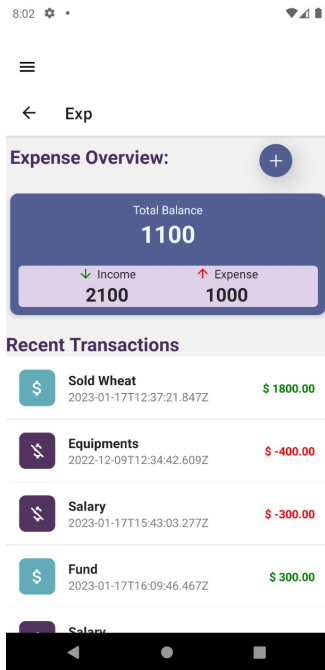


Figure 25: Expense Display for specific season .

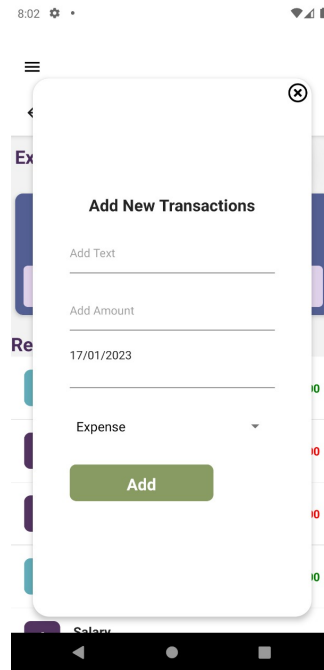


Figure 26: Add Expense to this open expense category.

Here the sales screen where every sold harvest is shown . With details about each one ,like quality when it is sold , the total income from this sales.

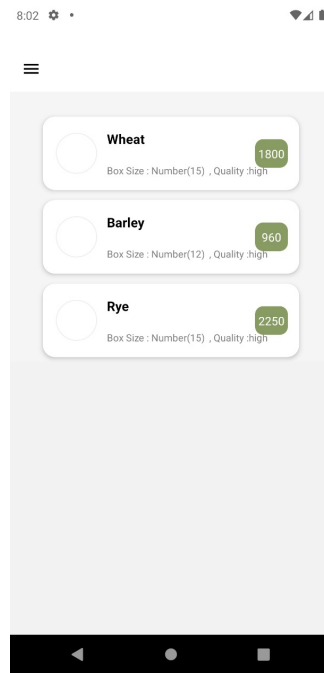


Figure 27: Display Sales.

4.7 Worker Side

After worker login ,the home screen shown in figure 28 will display . All Tasks he have to do will be displayed on the screen . Moreover when the worker press Done that will remove the task and update the status of task to his manager .

In addition to chatting ability between workers.

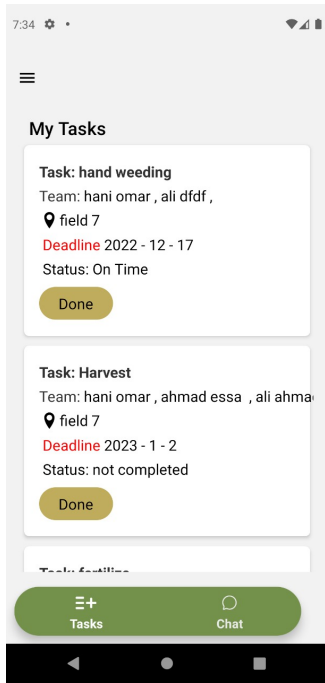


Figure 28: Tasks delegated for this worker user.

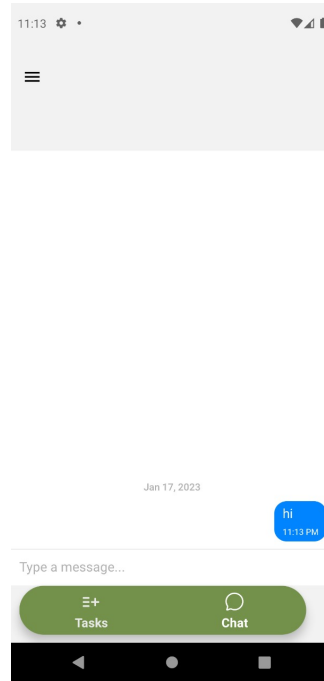


Figure 29: Chat between workers.

4.8 Web Application

This section will show snippets of our Web-Platform.

4.8.1 Admin Login

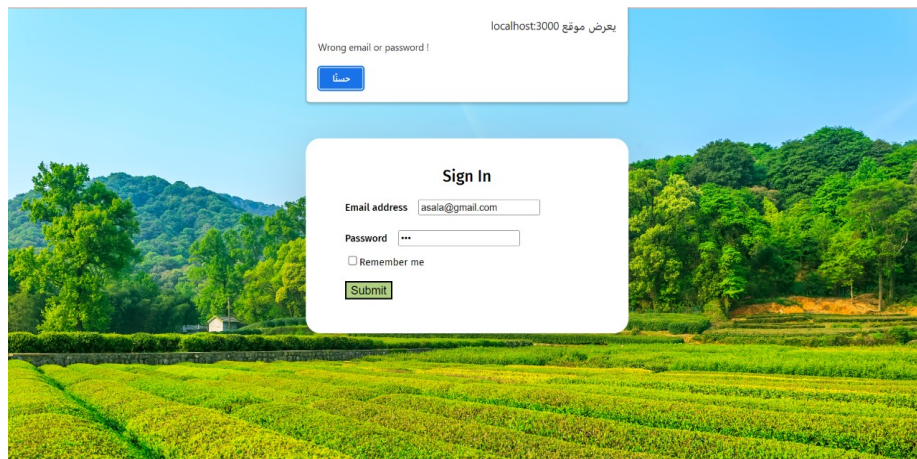


Figure 30: Sign in with wrong password.

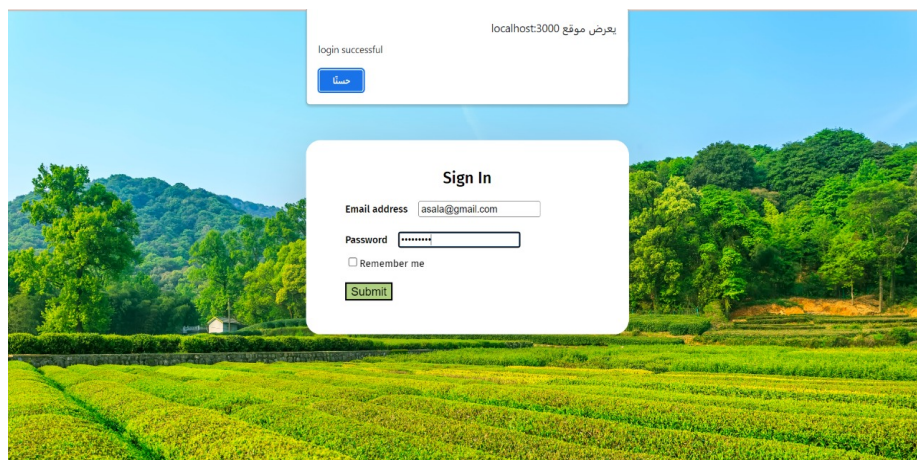


Figure 31: Sign in Successfully

4.8.2 Admin site details

After Admin login successfully ,Dashboard screen with overall view information about farmer fields will appear.

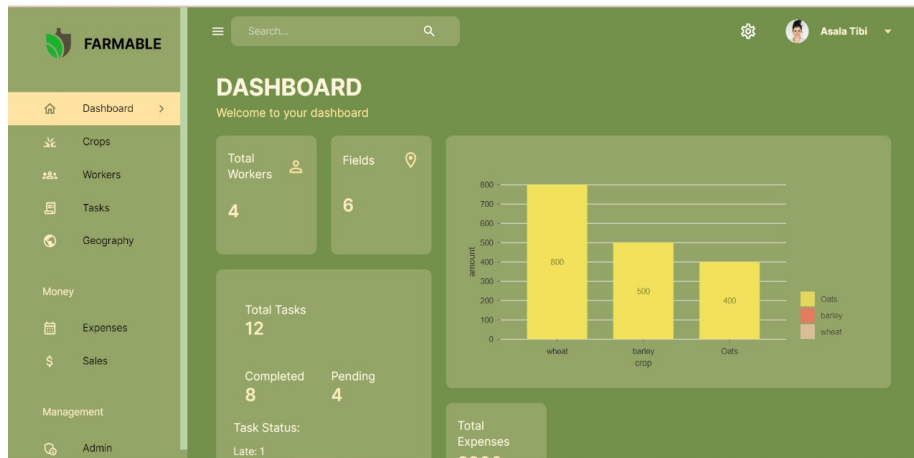


Figure 32: Dashboard



Figure 33: Dashboard with pie charts

Display all existing crops . Admin able to modify crops data , delete specific crop and adding a new crop.

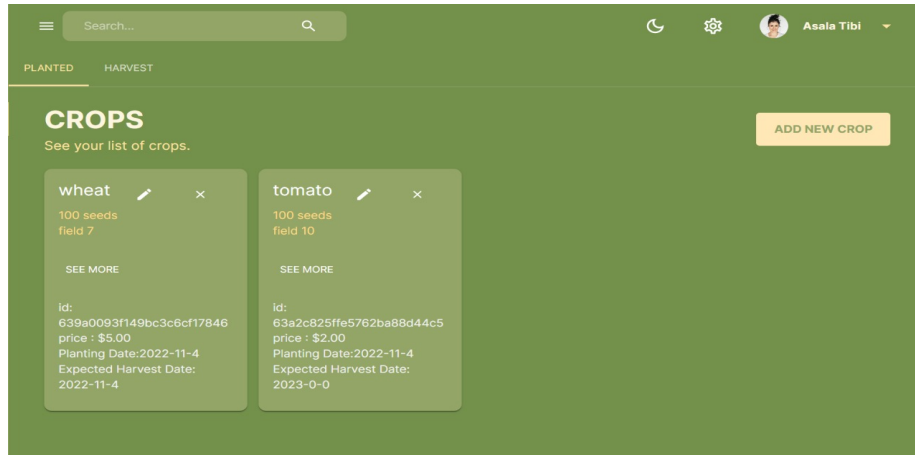


Figure 34: Display existing crops data

Display all workers with ability to delete or update data of any of them, In additional to Add new workers.

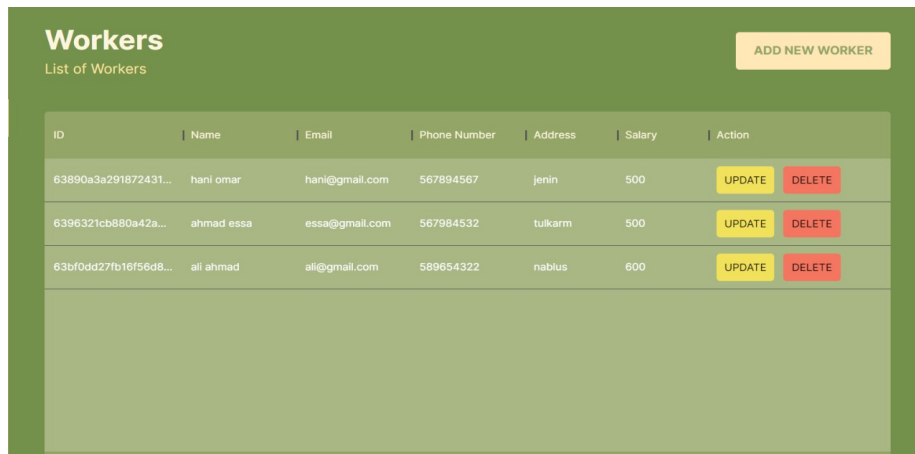


Figure 35: Display existing worker data

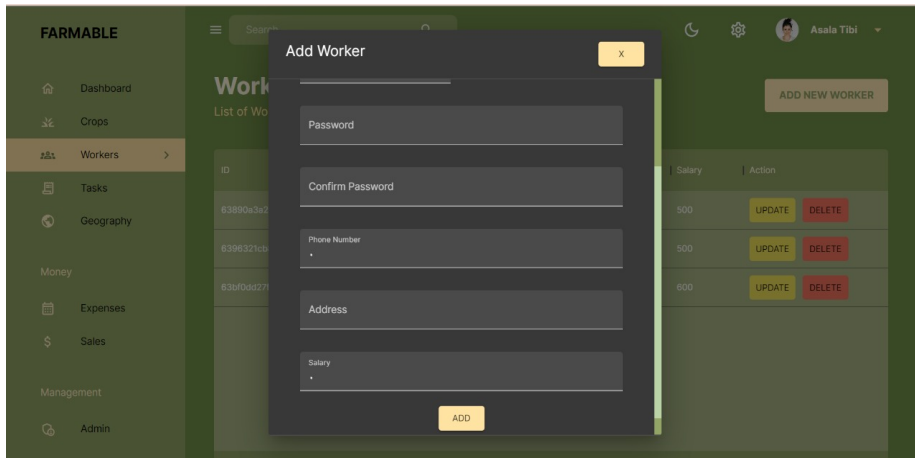


Figure 36: Adding new Worker.

Display all tasks admin assigned to workers.

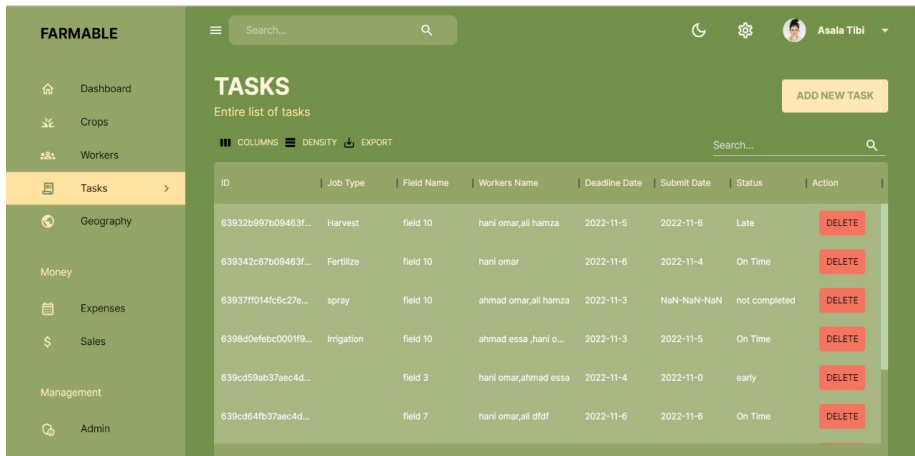


Figure 37: Display assigned Tasks data

TASKS
Entire list of tasks

ADD NEW TASK

COLUMNS DENSITY EXPORT Search...

ID	Job Type	Field Name	Workers Name	Deadline Date	Submit Date	Status	Action
			hani omar,ali hamza	2022-11-5	2022-11-6	Late	DELETE
			hani omar	2022-11-6	2022-11-4	On Time	DELETE
			ahmad omar,ali hamza	2022-11-3	NaN-NaN-NaN	not completed	DELETE
			ahmad essa ,hani o...	2022-11-3	2022-11-5	On Time	DELETE
			hani omar,ahmad essa	2022-11-4	2022-11-0	early	DELETE
			hani omar,ali ddf	2022-11-6	2022-11-6	On Time	DELETE

Find column
Column title

- ID
- Job Type
- Field Name
- Workers Name
- Deadline Date
- Submit Date
- Status
- Action

HIDE ALL SHOW ALL

Figure 38: Features can be done on Task table

COLUMNS DENSITY EXPORT Search...

ID	Job T... ↑	Field Name	Workers Name	Deadline Date	Submit Date	Status	Action
63932b997b09463f		field 10	hani omar,ali hamza	2022-11-5	2022-11-6	Late	DELETE
639342c87b09463f		field 10	hani omar	2022-11-6	2022-11-4	On Time	DELETE
63937ff014fc6c27e		field 10	ahmad omar,ali hamza	2022-11-3	NaN-NaN-NaN	not completed	DELETE
6398d0efebc0001f9		field 10	ahmad essa ,hani o...	2022-11-3	2022-11-5	On Time	DELETE
639cd59ab37aec4d...		field 3	hani omar,ahmad essa	2022-11-4	2022-11-0	early	DELETE

Unsort
Sort by ASC
Sort by DESC
Filter
Hide
Show columns

Figure 39: filtering, sort,...etc

Columns DENSITY EXPORT Search...

ID	Job Type ↑	Field ... ▼	Workers Name	Deadline Date	Submit Date	Status	Action
Columns Operator Value X Field Name ▼ contains ▼ field 10			hani omar	2022-11-5	2022-11-6	Late	DELETE
639342c87b09463f...	Fertilize	field 10	hani omar	2022-11-6	2022-11-4	On Time	DELETE
63937f014fc6c27ec...	spray	field 10	ahmad omar,ali hamza	2022-11-3	NaN-NaN-NaN	not completed	DELETE
6398d0efebc0001f9...	Irrigation	field 10	ahmad essa ,hani om...	2022-11-3	2022-11-5	On Time	DELETE

Figure 40: search , contain...etc

For Fields locations display , we use Google maps Api .

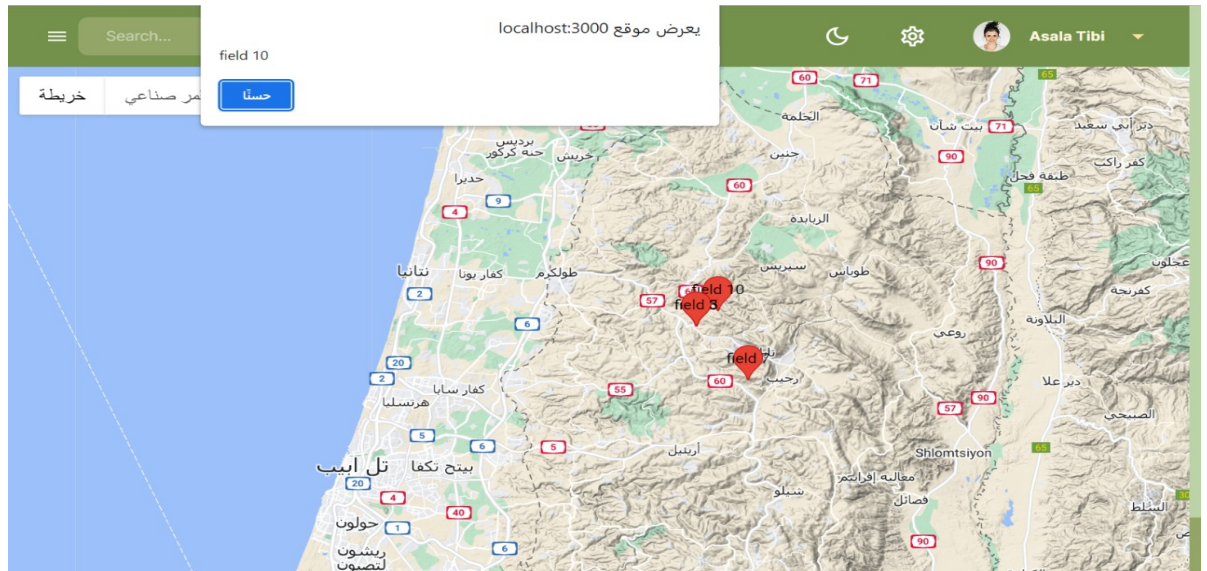


Figure 41: fields location appear on map

Display each of Sales and Expenses data with ability to delete and update to expense table and add new expense to table.

Expenses
Track your farm Expenses Here

ADD NEW TRANSACTION

ID	Expense Type	Cost	Created At	Action
63b977a760033129b0a7aabb	Ahmad Salary	500	2022-11-5	UPDATE DELETE
63b977e260033129b0a7aabc	SALARY2	100	2022-11-5	UPDATE DELETE
63b977ef60033129b0a7aabd	SALARY3	30	2022-11-5	UPDATE DELETE
63c07cc575a5f33a885e12af	money	600	2023-0-4	UPDATE DELETE
63c3246ecc2c16471811cdd4	Sold tomato	6000	2023-0-6	UPDATE DELETE

Figure 42: Expense Table

Sales
List of Sales from harvested crops

ID	Crop Name	Amount	Box Size	Quality	Total Price
63b4876abb8bb5018c25915e	tomato	5	10	low	250
63b8a2ed1cdf476092cff9f1	wheat	10	10	low	300
63c31dfbaf4a1c480444c238	tomato	20	100	High	6000
63c326adcc2c16471811cdd8	tomato	20	100	High	4000

Figure 43: Sales Table

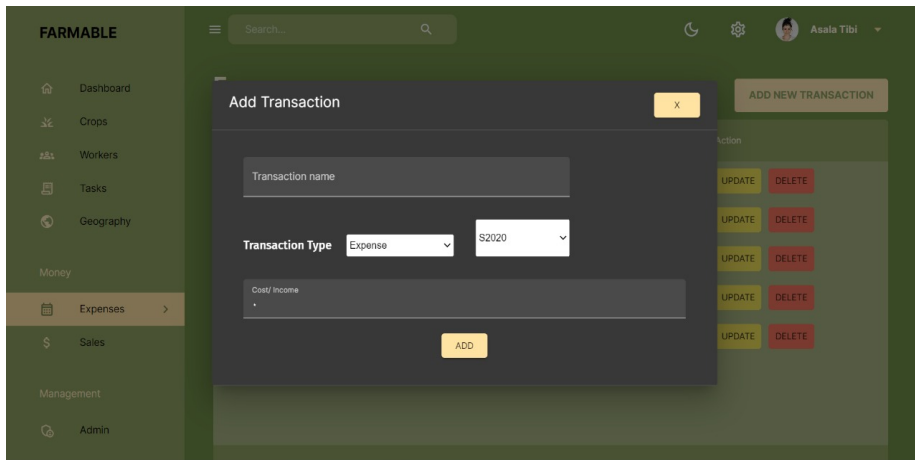


Figure 44: Example of one of Adding screen -Expense -.

Chapter 5

5 Conclusion/ Recommendations and Future Works

5.1 Conclusions

In this research we built a farmer management application and website completely from zero in order to match our requirements and customization to the product we tried to Market.

To sum up everything that has been stated so far , the primary goal of this management Platform is helping farmers plan and document every task , every expense in addition to all their activities for easy access and analysis which will make management easy for them especially to the in-experienced farmers.

Another point is that we test every feature on the farm before rolling it out . We believe in building technology from the ground up and actively involve growers in the feedback process.

5.2 Future Work

Much can be done as future work regarding our version which aims to facilitate and organize farmers' lives . Moreover to provide them more control using technologies .

We will work on merging hardware technologies to our application for more precisely tracking the crops in each field using hardware sensors for example .This could give

us feedback about the status of crops in addition to the use of Internet Of Things. Furthermore , track worker movement on the fields by working on the field maps to be more accurate by using . App will support Arabic language to help farmers who lack English knowledge.

References

- [1] Alexandros Kaloxylos et al. “Farm management systems and the Future Internet era”. In: *Computers and Electronics in Agriculture* 89 (2012), pp. 130–144. ISSN: 0168-1699. DOI: <https://doi.org/10.1016/j.compag.2012.09.002>. URL: <https://www.sciencedirect.com/science/article/pii/S0168169912002219>.
- [2] *React Documenration*. <https://reactjs.org/>. 2022.
- [3] *React Native Documenration*. <https://reactnative.dev/>. 2022.
- [4] Jonathan Tyler Welte. “A farm management information system with task-specific, collaborative mobile apps and cloud storage services”. In: (2014).