Faculty of Engineering and Information Technology

Computer Engineering Department

Contracting Company

Prepared by

Ehab Dwikat

Yazan Alsade

Supervised by

Dr. Anas Toma

Presented in partial fulfillment of the requirements for Bachelor degree in Computer Engineering.

Acknowledgment

We would like to express our deepest appreciation to our families and friends for their huge support during the project. Our success would not have been possible without the support of Dr.Anas Toma Thanks will not be enough for their valuable advice and helpful contributions .Your sincerely Yazan and Ehab.

Disclaimer Statement

This report was written by Yazan and Ehab at the Computer Engineering Department, Faculty of Engineering, An-Najah National University. It has not been altered or corrected, other than editorial corrections, as a result of assessment and it may contain language as well as content errors. The views expressed in it along with any outcomes and recommendations are solely those of the students. An-Najah National University accepts no responsibility or liability for the consequences of this report being used for a purpose other than the purpose for which it was commissioned..

.

Abstract:

This application is important for contracting companies because it helps to manage the business of these companies, monitor their employees, follow up the speed and completion of projects within that company, in addition to having records of the attendance and absence of workers and the total number of working days, in addition to the existence of a special part for the equipment and materials available to the company, which the supervisor can use to request the necessary materials To complete and evaluate projects, there is also the ability for the worker when he is added to the company to know his current location on the maps as well as knowing the location of the special task for easy access.

Table of Contents:

Chapter 1: Introduction

1.1 Statement of the problem

Nowadays, the trend towards real estate and contracting has increased, and therefore those responsible for these businesses will certainly face great problems in organizing their work. In addition, the presence of such applications for them saves them time, effort and money. Instead of having a large number of employees responsible for the attendance and absence of workers and recording what enters and exits for these Companies from materials and equipment such an application came to help them in carrying out all these tasks that used to take a lot of effort. All the data they need will be stored electronically within huge databases so that they can be referred to at any time and this makes it easier for them to organize and distribute everything within the company as well as reduce Absenteeism and material theft In addition, some patients require a home visit from the lab for checks but are unable to reach the labs that provide this service.

1.2 Objectives

The main goal of the application is to organize the work of companies and not to lose any information, whatever it is, because it is stored in databases, in addition to saving time and effort for the owners of these companies, because they can reduce the number of employees responsible for administrative matters, and thus exploit their salaries with other things useful for the development of the company

1.3 Scope Of work

We have built this application so that it allows officials in the contracting company, such as engineers, to register an account in this application, then they can make a login to access the main page in this application, after that different pages will appear for him, such as the page for projects and also the page for materials and heavy equipment in order for him to review it in addition There is a page for the workers inside the company, where the engineer has the ability to add several tasks for each worker, in addition to recording the attendance and absence of this worker.

The engineer is also able to access the maps page to track the locations of workers, tasks and projects on this map, and he also adds a worker and his task to this worker. This page also makes it easier for the worker himself to know the location of his task that he must accomplish.

Chapter 2: Constraints and Technologies

2.1 constraints :

- The biggest issue we had was the heavy load our laptops were under from extended workdays, which combined with their limited storage capacity resulted in poor performance.

- We spent a lot of time learning new programming languages like Dart and mobile frameworks like flutter because this was our first time developing a mobile application and we had no prior knowledge of its technologies. We also had to learn nodeJs and the express framework in order to work with the Mongo DB database management system.
-

2.2 Technologies:

In this project many technologies are used :

- For mobile front-end development: flutter framework was used.
- For website front-end development: html+css+js+media query
- For the APIs building (back-end): Node.js and Express were used.
- For the Database: we used MongoDB database.
- Version Control with Git and Github were used.
- For maps Google APIs were used.

Chapter 3 : Literature Review

We conducted extensive research before trying to put the project idea into action.
We discovered apps and concepts that are similar to the one we wish to use.
Although there are other apps that are somewhat comparable to the concept we want to build These applications were not things that comprise all that the contracting companies require to arrange their demands, but rather things that were highly unique and personalized. One of these applications merely shows the materials and their categories, without any other information.

In our Project , we have built this application so that it allows officials in the contracting company, such as engineers, to register an account in this application, then they can make a login to access the main page in this application, after that different pages will appear for him, such as the page for projects and also the page for materials and heavy equipment in order for him to review it in addition there is a page for the workers inside the company, where the engineer has the ability to add several tasks for each worker, in addition to recording the attendance and absence of this worker.

The engineer is also able to access the maps page to track the locations of workers, tasks and projects on this map, and he also adds a worker and his task to this worker. This page also makes it easier for the worker himself to know the location of his task that he must accomplish.

In addition, a function of machine learning called collaborative filtering was used to create a recommendation system so that when an engineer, for example, searches for a material or heavy equipment that he needs for a specific project, he searches for it, then he brings all the materials similar to this material that we searched for in terms of its description and price.

Chapter 4 : Methodology

4.1. Tools, Programming Languages, and Technologies

4.1.1. Mobile application

Flutter provides developers with a full toolbox for creating cross-platform mobile apps that have outstanding performance, gorgeous UIs, and shorter development times. It is the perfect framework for building reliable and aesthetically pleasing mobile applications thanks to its single codebase approach, large widget library, hot reload features, and active community support.

4.1.2 Back-End Languages

Node JS, the programming language we employ, is a back-end JavaScript runtime environment that is open-source, cross-platform, and runs on a JavaScript Engine. Node JS is simple to comprehend and has extensive documentation, making it simple to correct and troubleshoot any mistakes that could arise.

4.1.2 Website Dashboard

For website administrators, we have developed a dashboard.
The front-end was built using HTML and CSS, and the back-end with the nodeJs programming language.

4.1.3 Database

Popular NoSQL database management system MongoDB has several advantages for contemporary applications. Due to the fact that it is built to manage both structured and unstructured data, it is flexible and responsive to shifting data models.

A flexible data model is also provided by MongoDB. Data is saved in adaptable JSON-like documents called BSONs (Binary JSONs) using a document-oriented approach. This eliminates the requirement for intricate mapping or object-relational mapping (ORM) layers by enabling developers to work with data in a manner that closely mimics the data structures used in their applications. Faster and more effective development is made possible by the flexible data model, which also makes data management and querying simpler.

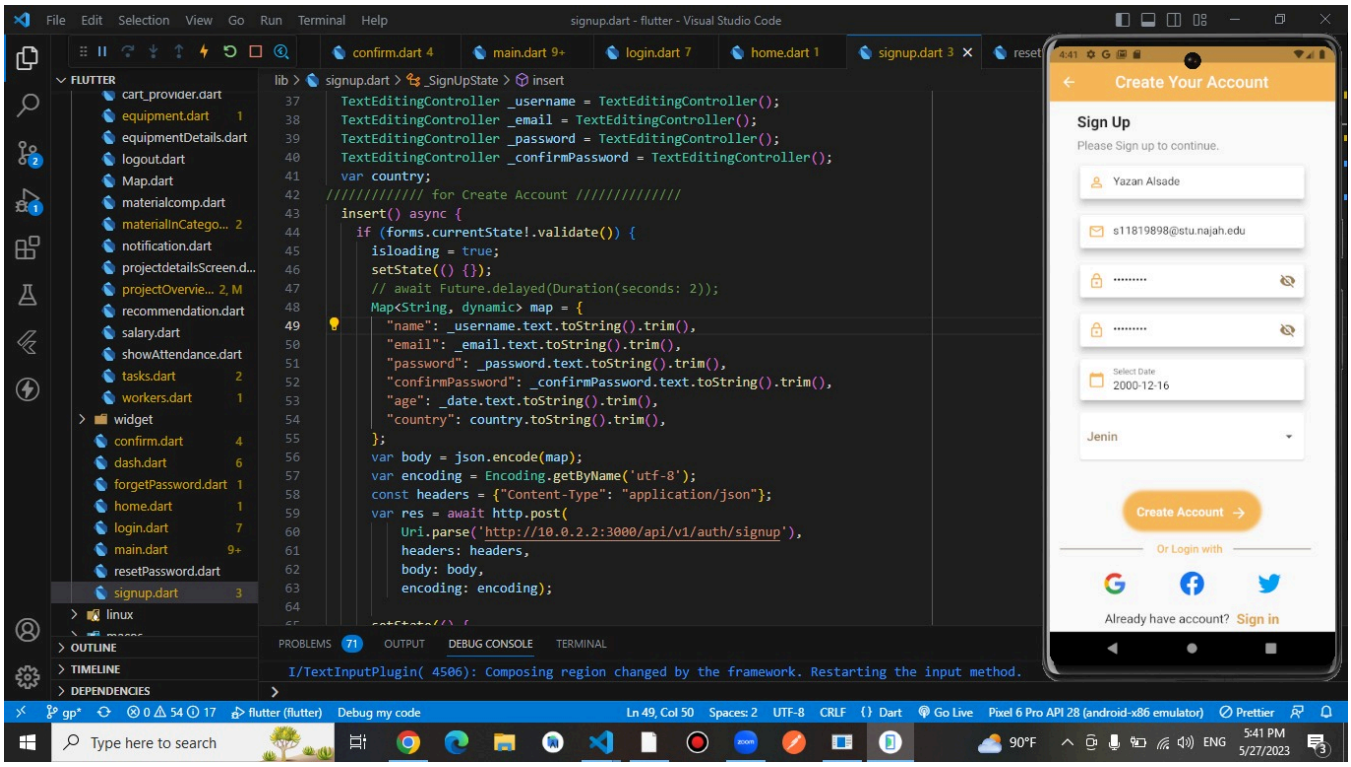
Scalability, a flexible data model, effective read and write operations, high availability and fault tolerance, and flexible deployment choices are all advantages of MongoDB. Due to these characteristics, MongoDB is a well-liked option for contemporary applications that need flexibility, performance, and scalability to handle a variety of changing data needs.

4.4 Implementation and Features

Mobile Application

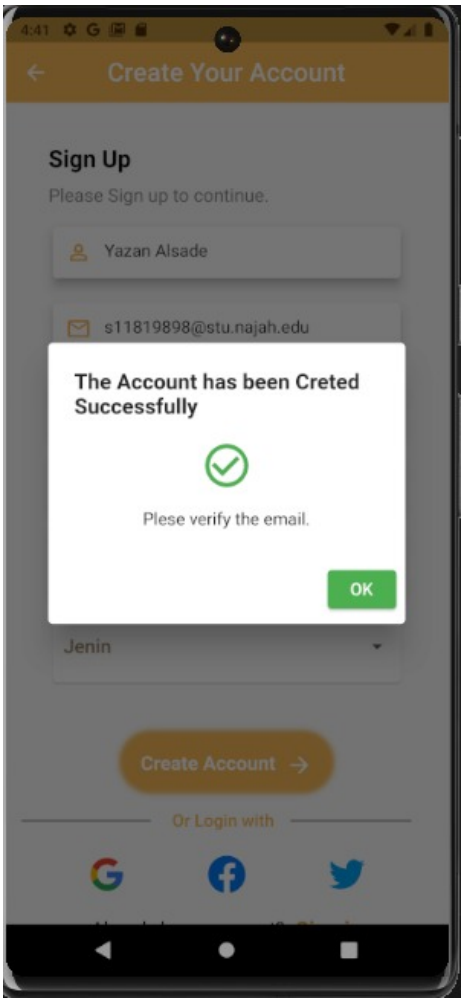
- **Sign-up screen**

At first, when you open the application, the sign up page appears, so that the engineer or worker creates his own account, where he enters his information in terms of name, email, password, date of birth, and place of residence..



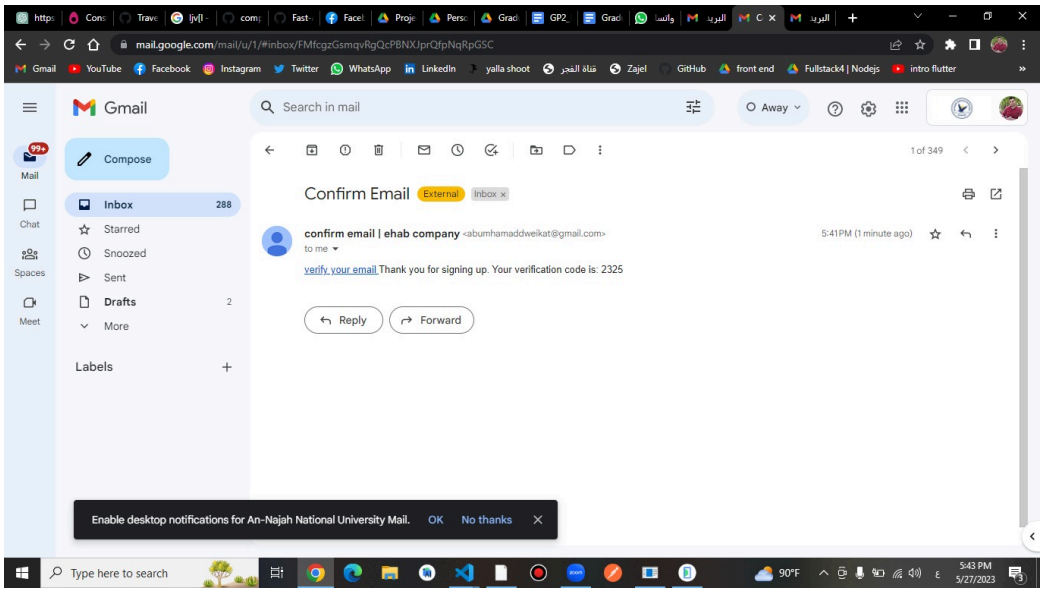
- **After SignUP :**

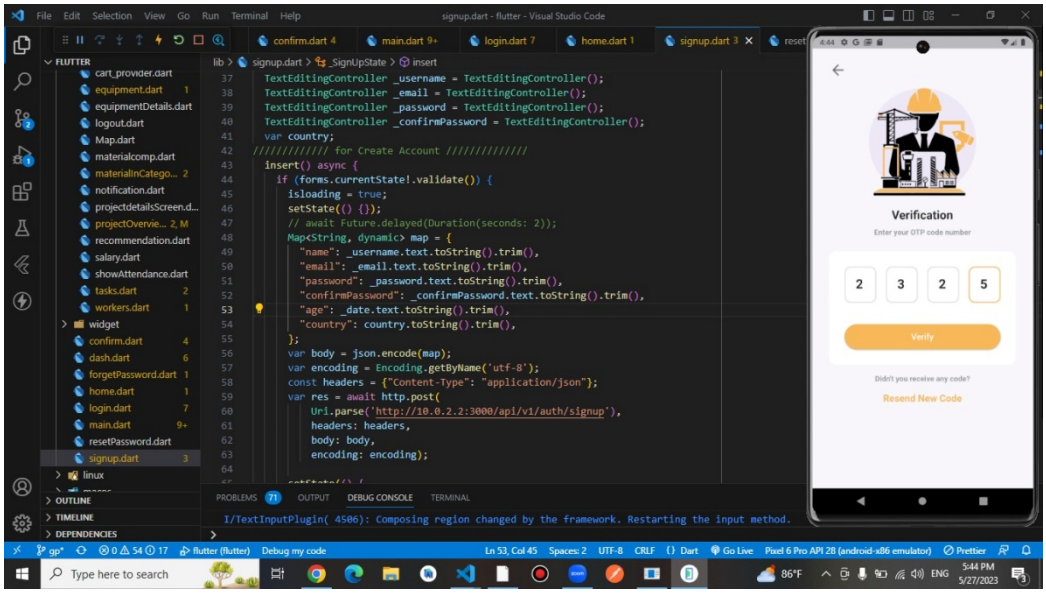
→The account will be created successfully and after that will send message to the user email with verification code .



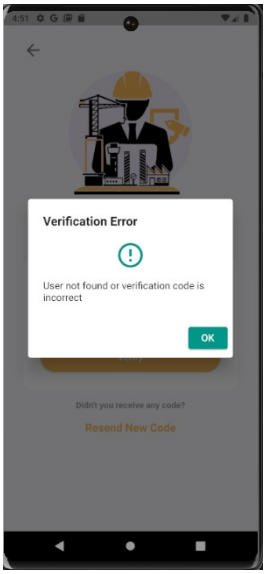
.

● Verification Code Page :

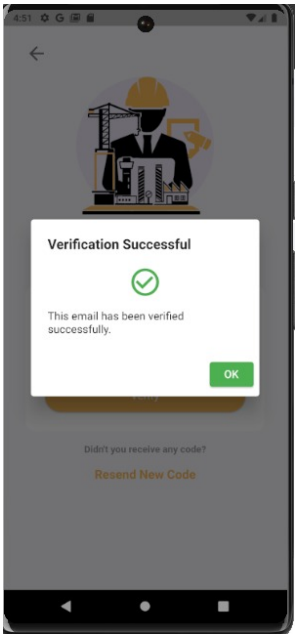




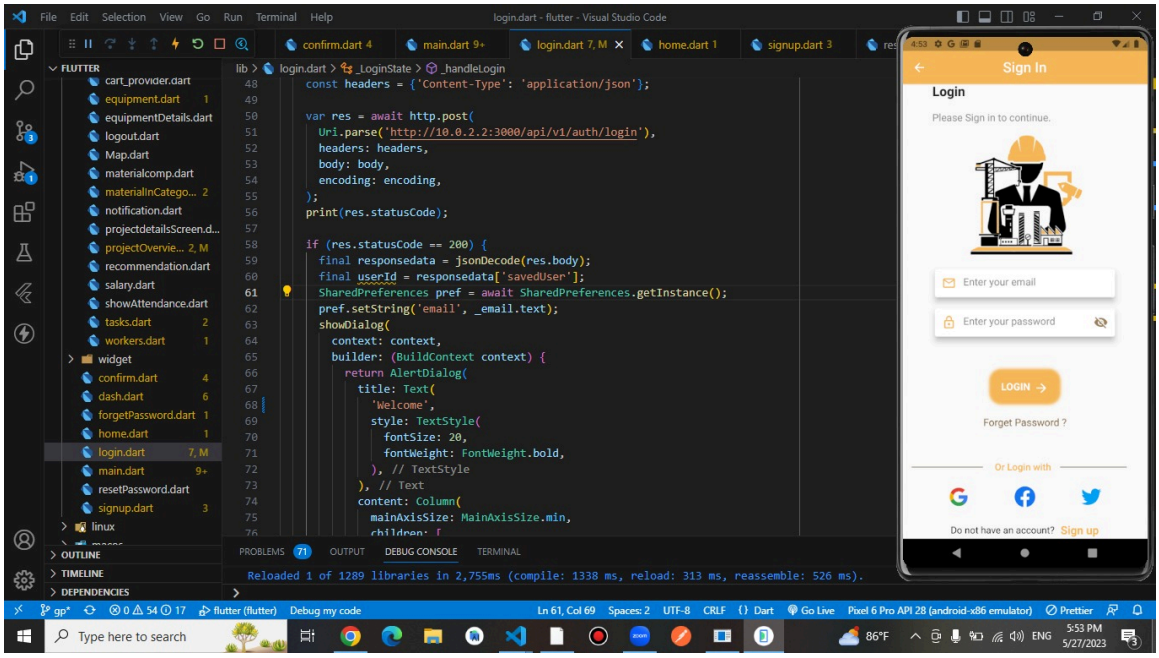
➔ **Now if the verification code that user enter is incorrect will have like this:**

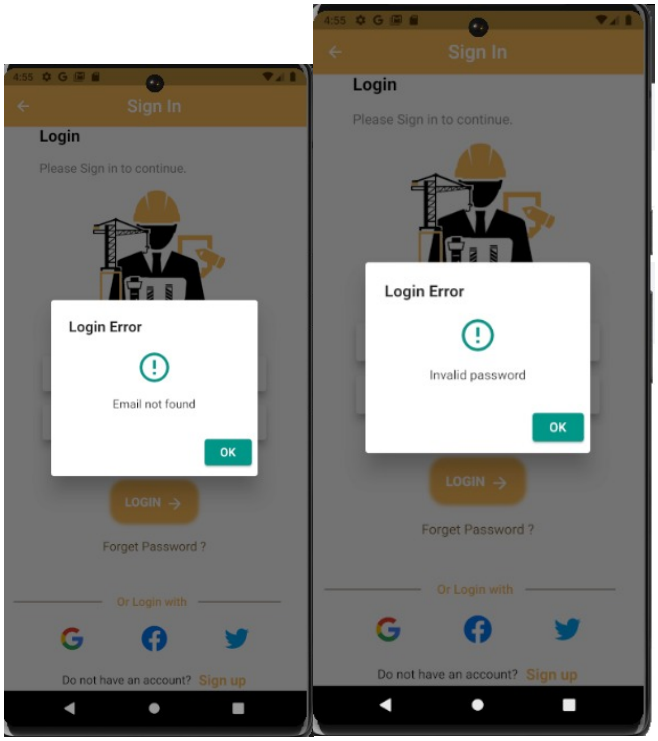


→If its correct will have this:

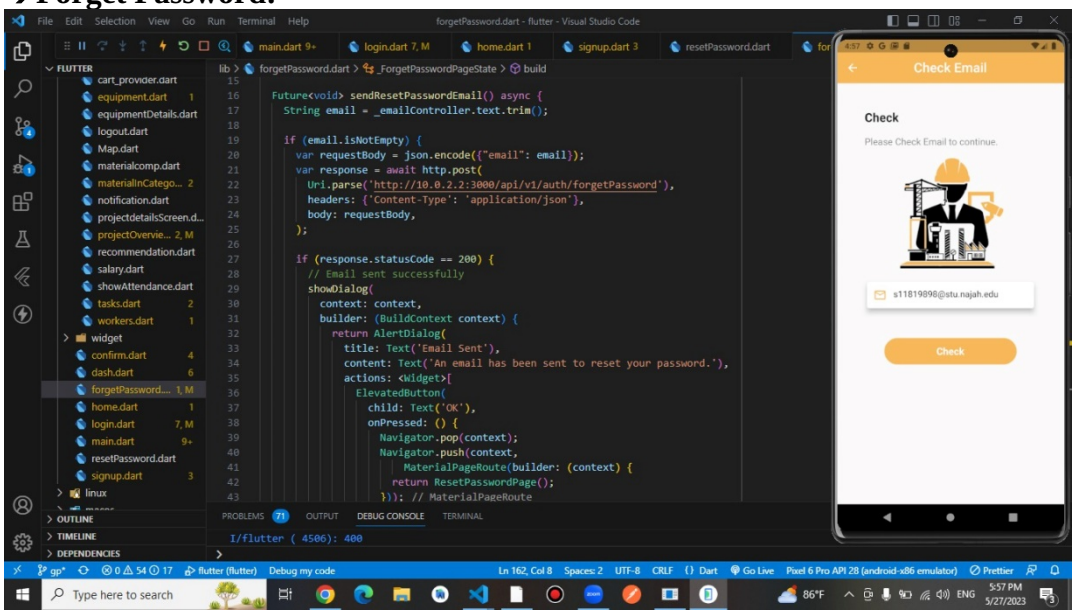


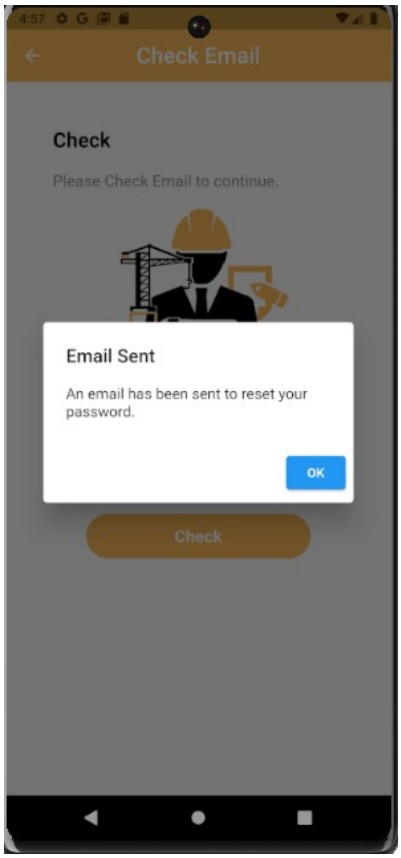
- **Login Screen:**

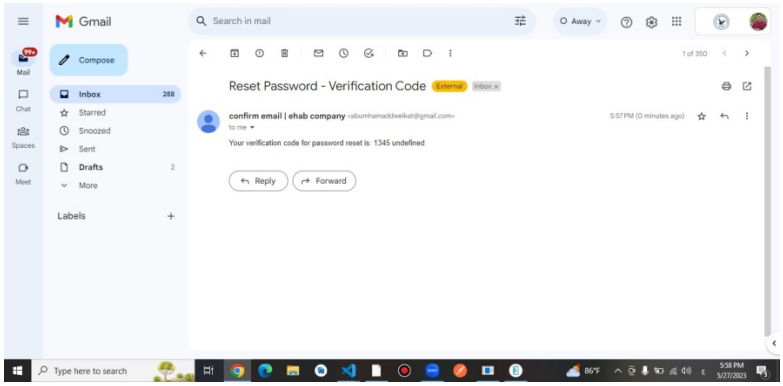


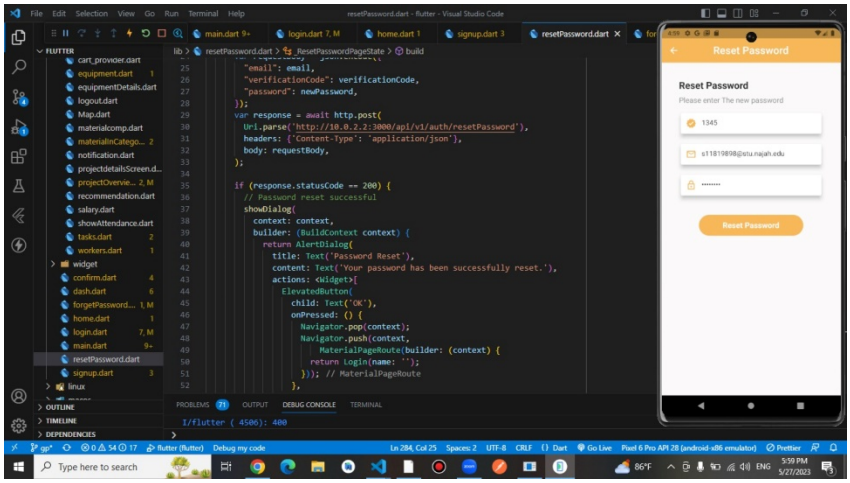


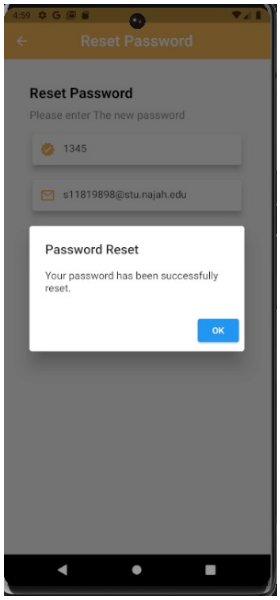
→Forget Password:



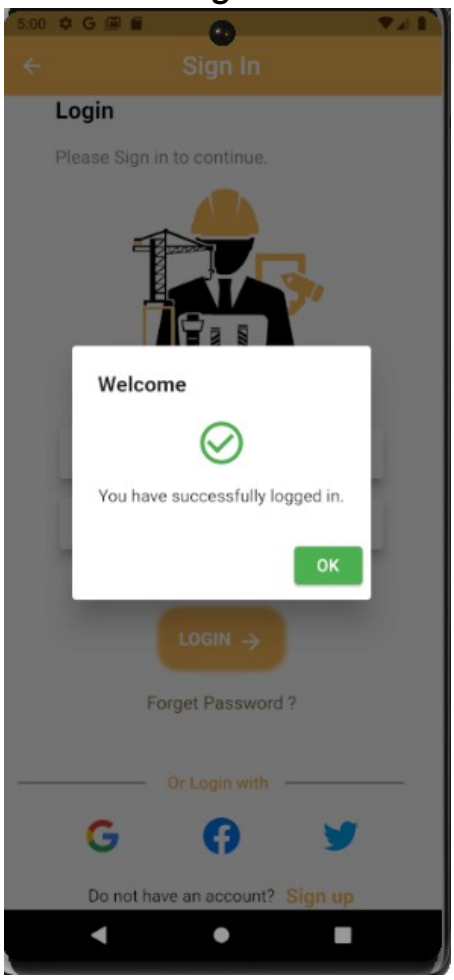



Gmail
Search in mail
Compose
Inbox 288
Starred
Snoozed
Sent
Drafts 2
More
Labels
Reset Password - Verification Code
confirm email | ehab company
Your verification code for password reset is: 1345 undefined
Reply
Forward


resetPassword.dart - flutter - Visual Studio Code
"email": email,
"verificationCode": verificationCode,
"password": newPassword,
var response = await http.post(
Uri.parse('http://10.0.2.2:3000/api/v1/auth/resetPassword'),
headers: {'Content-Type': 'application/json'},
body: requestBody,
if (response.statusCode == 200) {
// Password reset successful
showDialog(
context: context,
builder: (BuildContext context) {
return AlertDialog(
title: Text('Password Reset'),
content: Text('Your password has been successfully reset.'),
actions: <Widget>[
ElevatedButton(
child: Text('OK'),
onPressed: () {
Navigator.pop(context);
Navigator.push(context,
MaterialPageRoute(builder: (context) {
return Login(name: '');
})); // MaterialPageRoute
},
I/flutter (4506): 400
Reset Password
Reset Password
Please enter The new password
1345
s11819898@stu.najah.edu
Reset Password

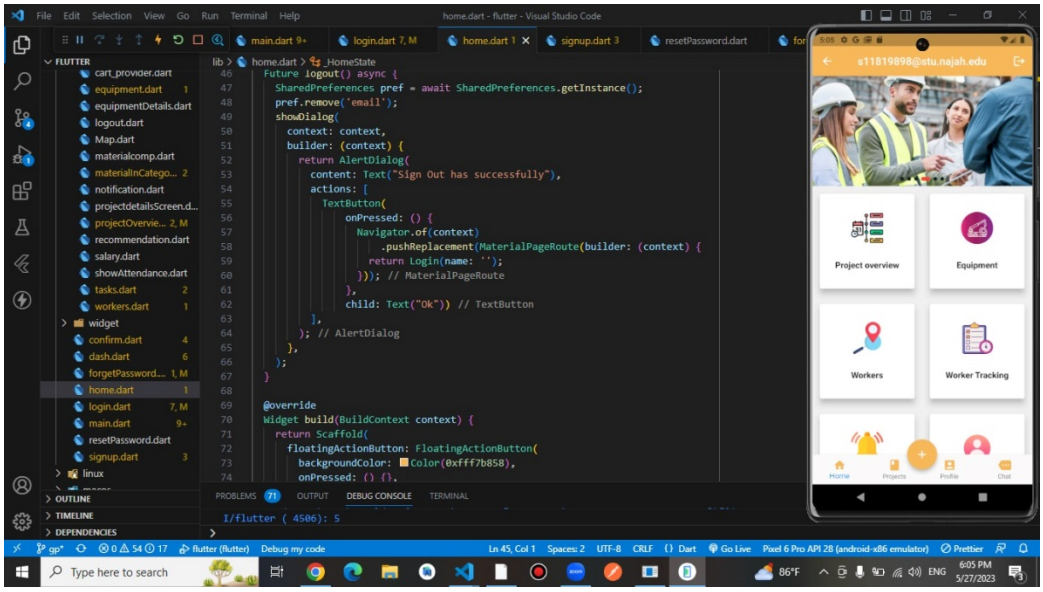

Reset Password
Reset Password
Please enter The new password
1345
s11819898@stu.najah.edu
Password Reset
Your password has been successfully reset.
OK

→Now the login done succssfully:



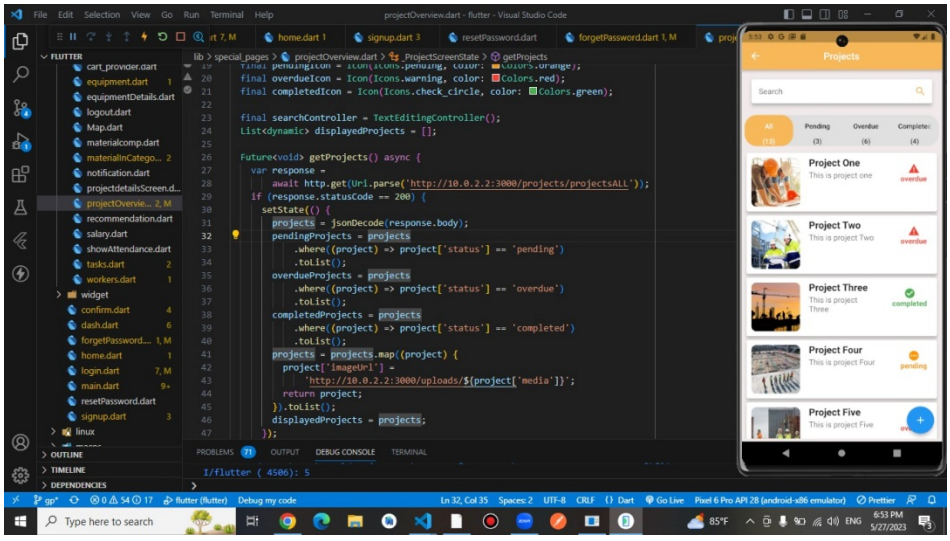
- **Home page:**

On this page, the engineer will choose one of the pages to access, and it contains several pages



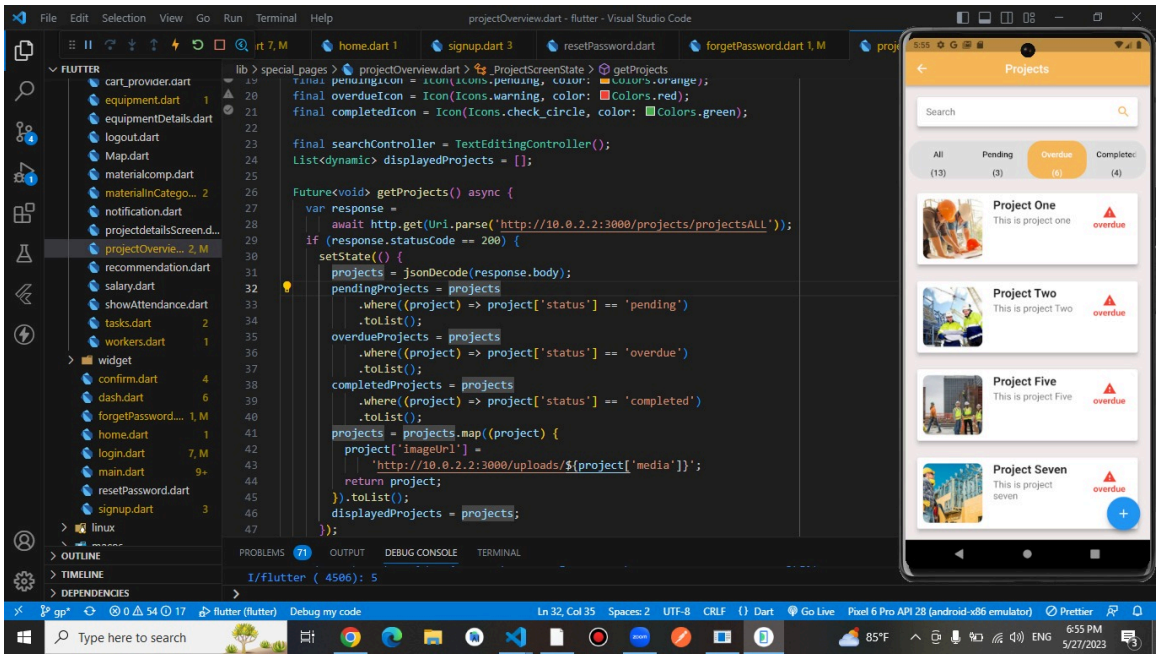
- **Project Page :**

→ On this page, the engineer and the administrator can see all the projects that the company is working on and supervise with the status of each project:

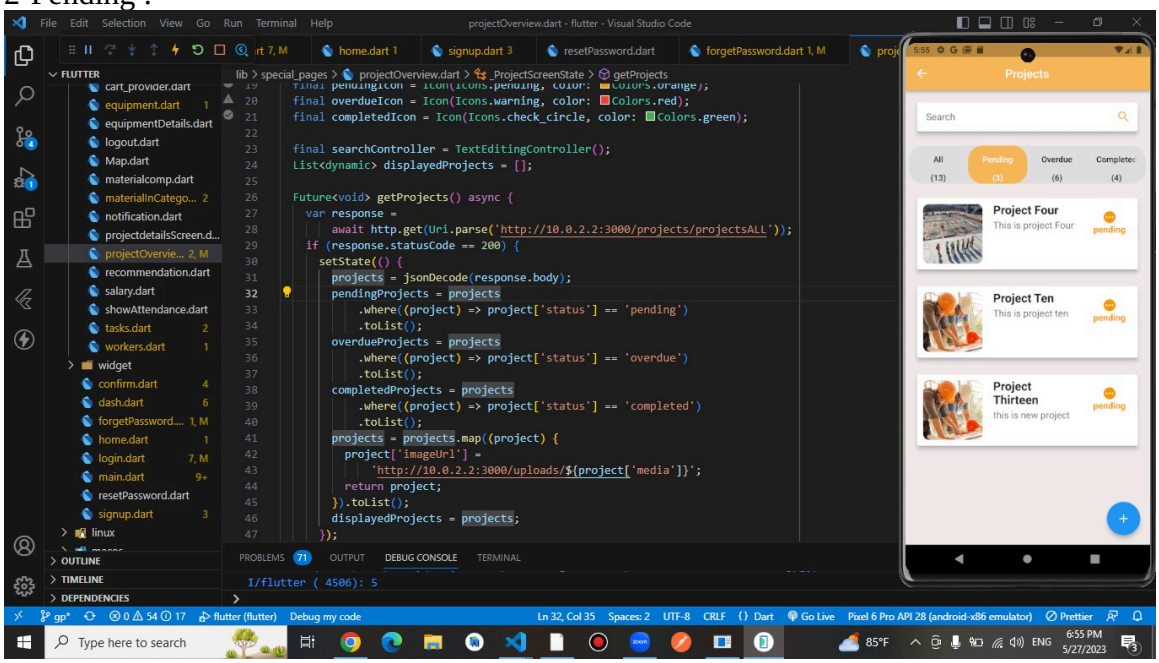


→ Also, the engineer can easily move between the three cases of the project:

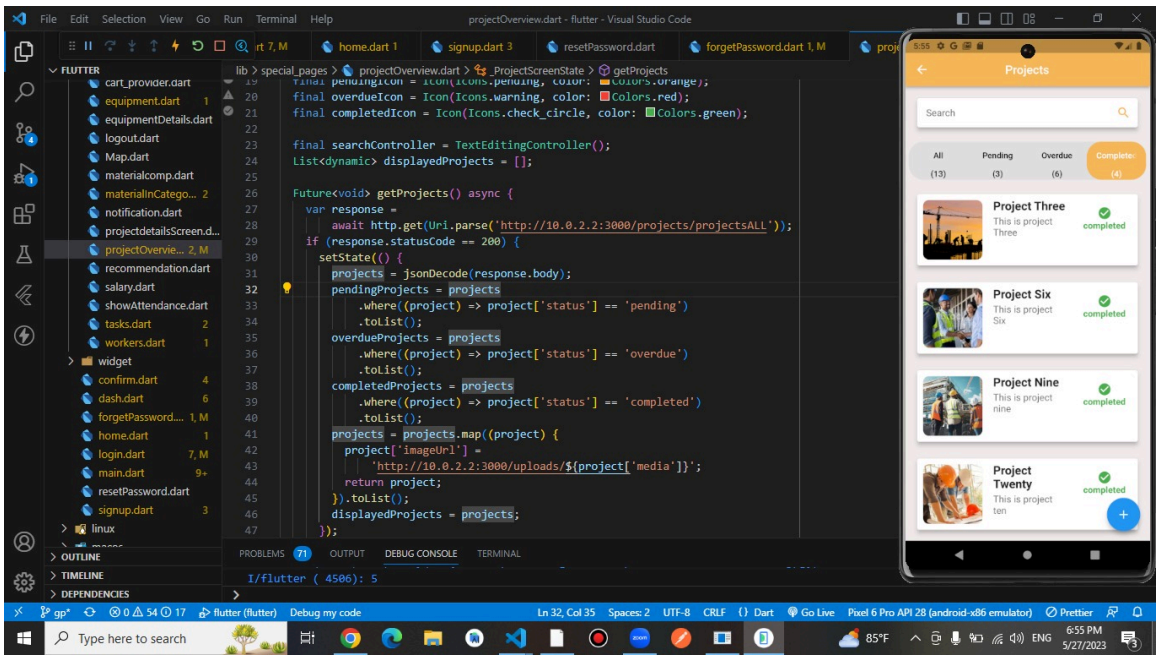
1-Overdue:



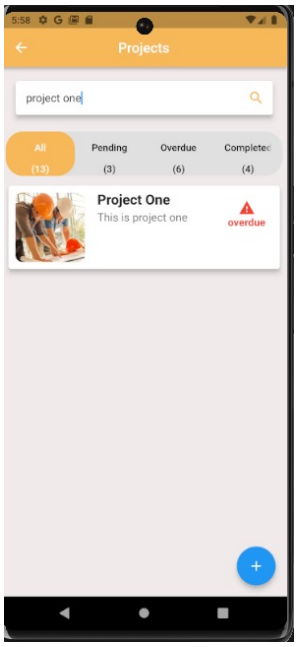
2-Pending :



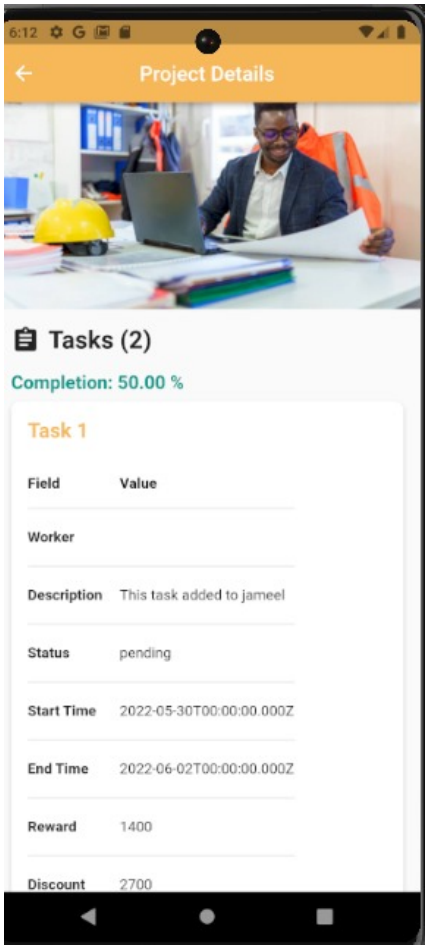
3-Completed:



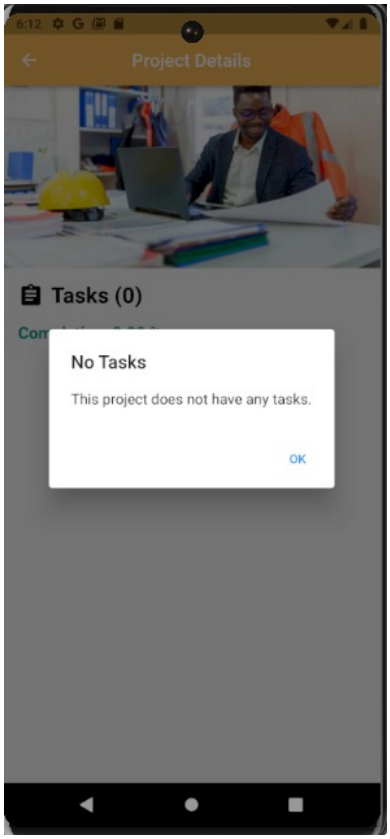
→The Engineer can search for specific Project :



→ Now, when the engineer clicks on a project, he will see all the tasks related to this project and the percentage of all tasks completed:



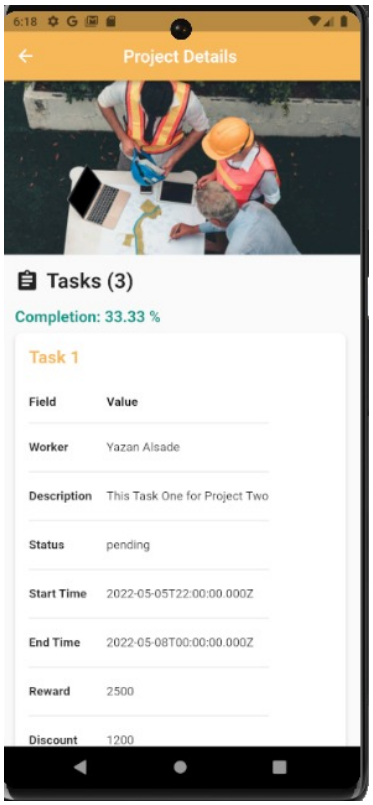
→Now If the Project Not Contain Any Taske:



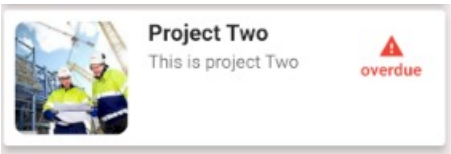
→ Now we will notice the following:

If we go back to the pictures above, we will notice that the status of the second project was Overdue. Now the process is done. If the percentage of completed tasks in each project is less than 40, the project status will remain Overdue. When the percentage of completed tasks becomes more than 40 and less than 80, the project status will change to Pending. Either if the percentage of completed tasks is greater than or equal to 80, the project status will become Completed:

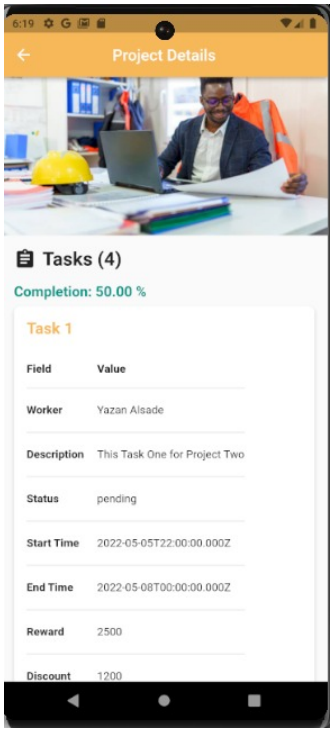
1- Overdue:



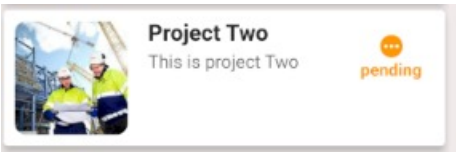
→The Status Of Project is:



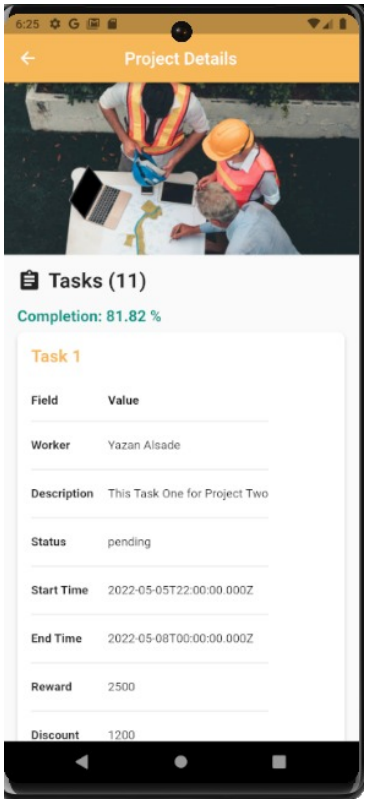
2-Pending :



→The Status Of Project is:



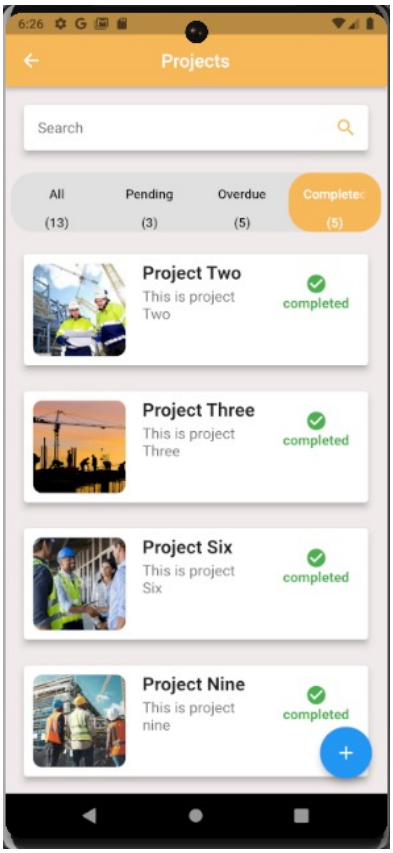
→Completed :



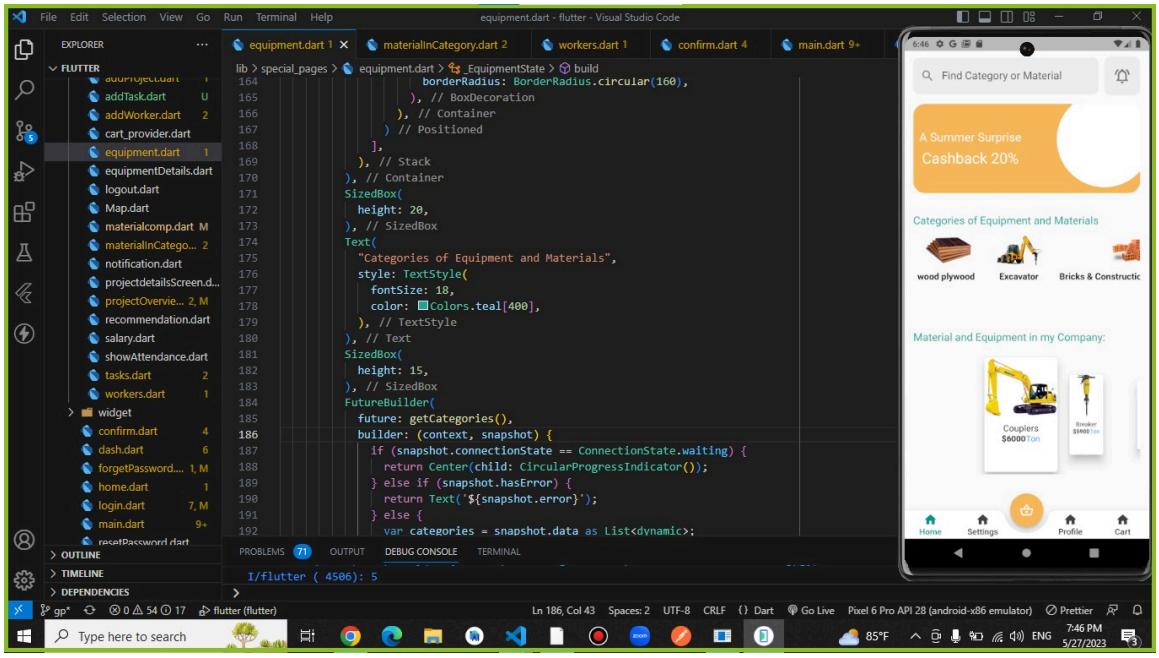
→The Status Of Project is:

Here, in the second project, the number of tasks contained in the project appears 11 task

→When the status of the second project became complete, it was added to the completed projects, and the number of completed projects increased to five projects instead of four projects

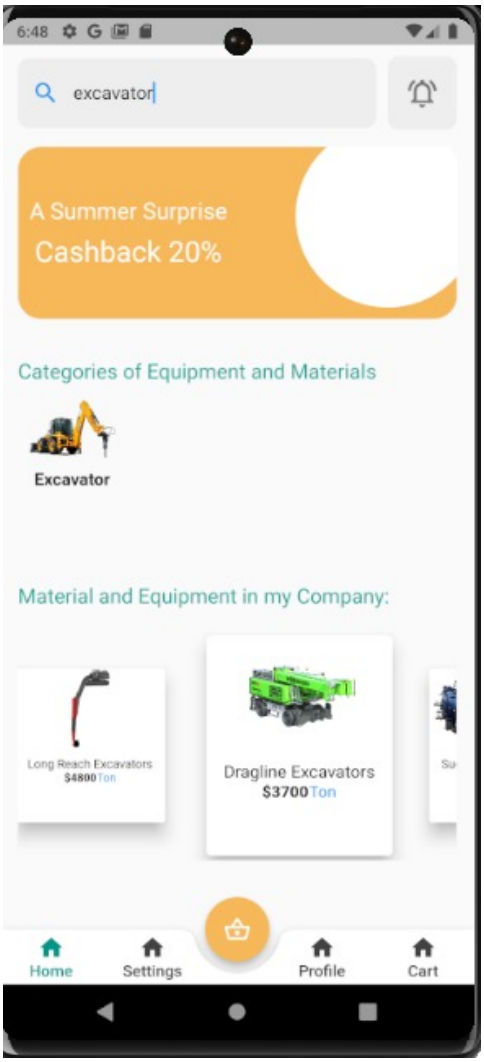


→Now, when the engineer is on the home page and selects the equipment page, this page will appear to him:

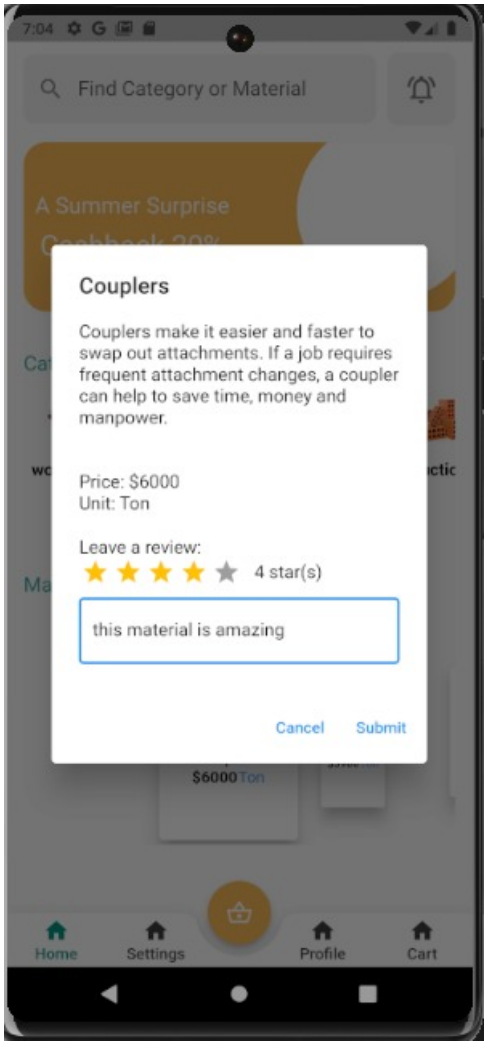


➔ This Is the categories on this page:

➔When the user searches for a specific material, it shows him all the materials with the same name, in addition to the category that contains these materials.

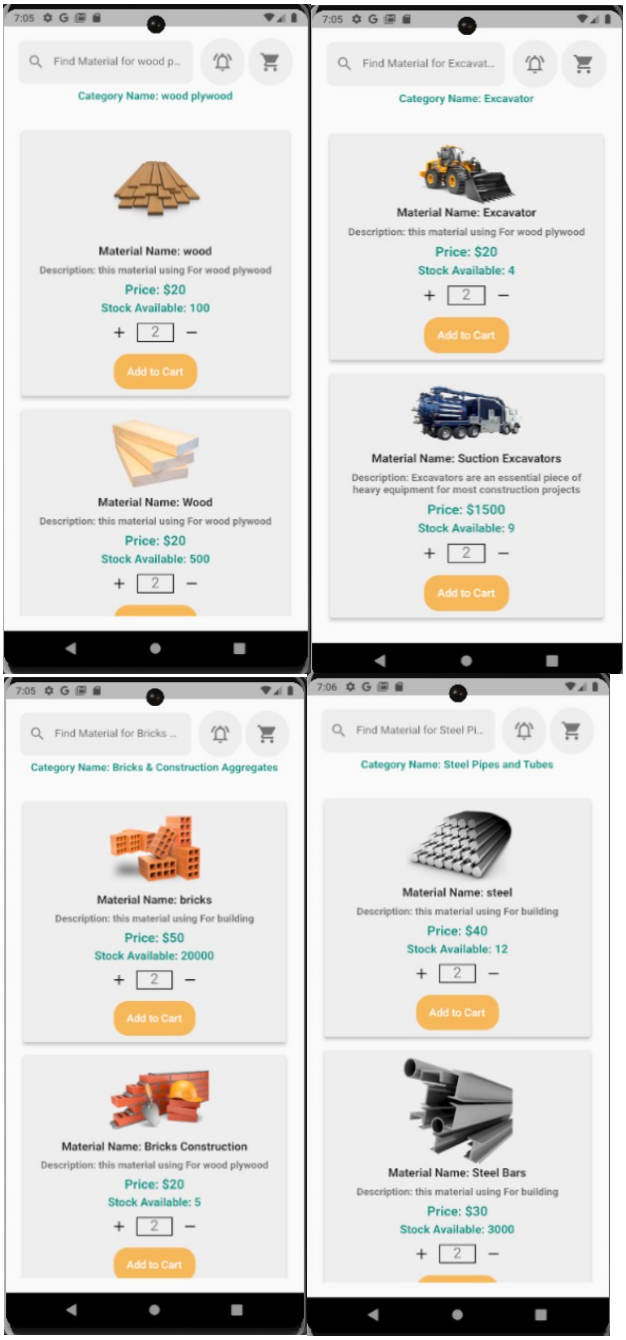


→When the user clicks on a specific material, a form will appear for him to evaluate and see the evaluations for this material:

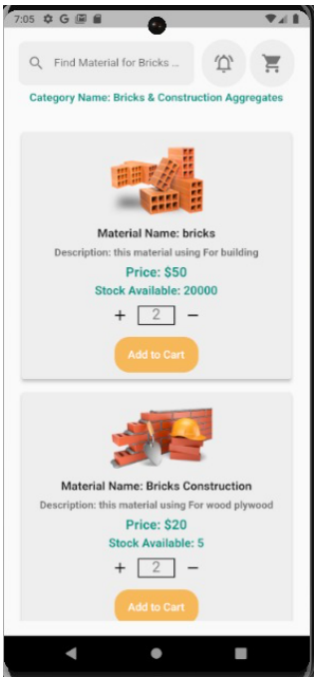


➔ And this message will appear:

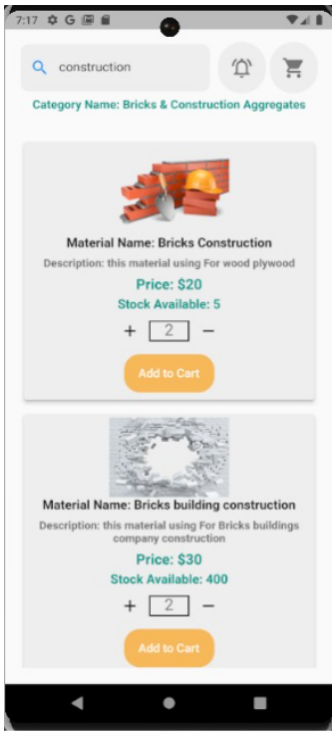
➔ When clicking on a specific category, it will show him all the materials in this category:



→When searching for a specific material within a specific category:

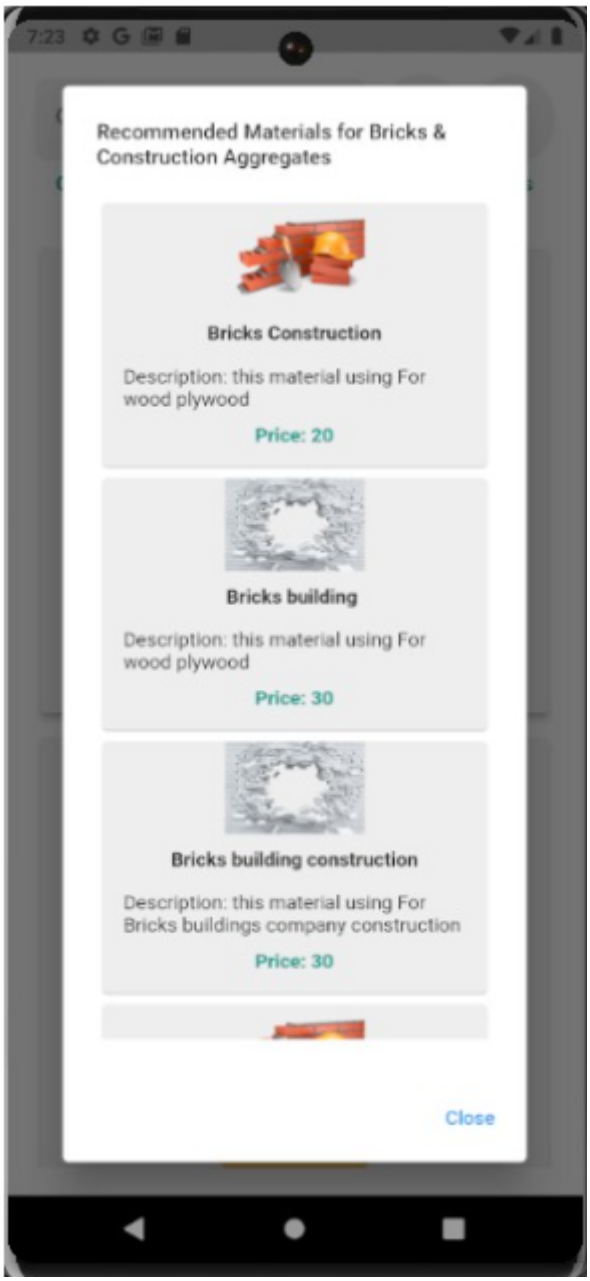


➔ In this category called bricks & construction aggregates all the materials we searched for have the same name



→Recommandation System:

When you click on a specific material, it shows the five most similar materials that we clicked on in terms of description and price. This is called the recommendation system and it is called collaboritve filltering, which is one of the types of machine learning.



→This is the collaborative Filttering That Written in node js :

components > Recomandation > JS Recomandation.services.js > cosineSimilarity

```
const { Material, Review,Category }= require('../materialsAndReview/materilas.module');
const { TfIdf } = require('natural');

// Get recommended materials for a given material id
exports.getRecommendedMaterials = async (req, res) => {
  const { id } = req.params;

  try {
    // Get the material for the given id
    const material = await Material.findById(id);

    // Extract material attributes
    const materialCategory = material.category;
    const materialPrice = material.price;
    const materialDescription = material.description;

    // Get all materials in the same category as the material
    const materials = await Material.find({ category: materialCategory });

    // Extract attributes for all materials
    const materialAttributes = materials.map(m => {
      const tfidf = new TfIdf();
      tfidf.addDocument(m.description);
      const tfidfVector = tfidf.tfidf;
      return {
        id: m._id,
        price: m.price,
        description: m.description,
        vector: tfidfVector
      }
    });

    // Calculate similarity between the material and all other materials
    const tfidf = new TfIdf();
    tfidf.addDocument(materialDescription);
    const materialVector = tfidf.tfidf;
    const similarMaterials = materialAttributes
      .map(m => ({
        id: m.id,
        similarity: cosineSimilarity(m.vector, materialVector)
      }))
      .filter(m => m.id != id)
      .sort((a, b) => b.similarity - a.similarity)
      .slice(0, 5);

    // Get details for recommended materials
    const recommendedMaterials = await Promise.all(similarMaterials.map(async m => {
      const material = await Material.findById(m.id);
      return {
        id: material._id,
        name: material.name,
        price: material.price,
        description: material.description,
        imageUrl: `${req.protocol}://${req.get('host')}/uploads/${material.media}`
      }
    }));

    return res.status(200).json({ message: 'Recommended materials found', recommendedMaterials });
  } catch (error) {
    console.error(error);
```

→This is explained for this recomandition system :

1-The code receives a request for recommended materials based on a given material ID.

2-It retrieves the material associated with the ID from the database.

3-The material's category, price, and description are extracted.

4-All materials in the same category as the queried material are fetched from the database.

5-For each material, a TF-IDF vector is calculated based on its description.

6-The cosine similarity is computed between the TF-IDF vector of the queried material and all other material vectors.

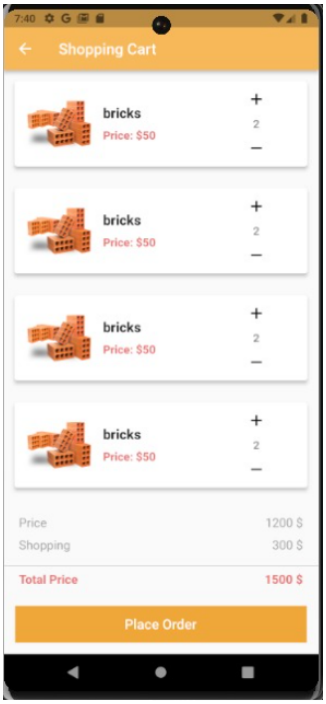
7-The similar materials are filtered to exclude the queried material, sorted based on similarity scores, and limited to the top 5.

8-Details of the recommended materials (name, price, description, media URL) are retrieved from the database.

9-The recommended materials are returned as a JSON response.

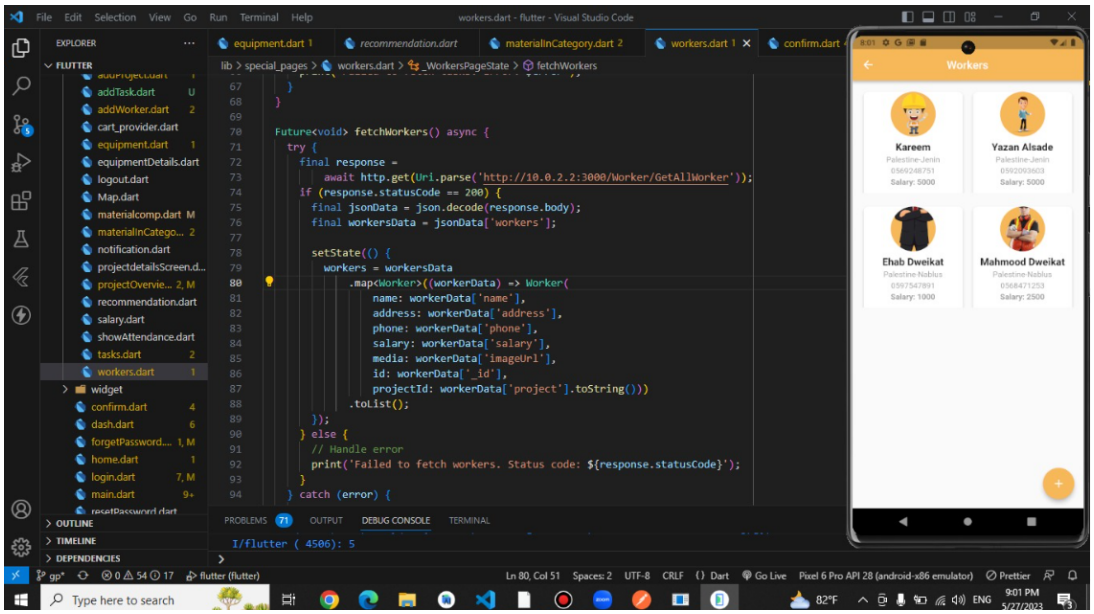
10-The logic revolves around using TF-IDF and cosine similarity to determine the similarity between the queried material and other materials, and then retrieving and returning the most similar materials as recommendations.

→When the user wants to buy a specific material, he adds it to the chart:

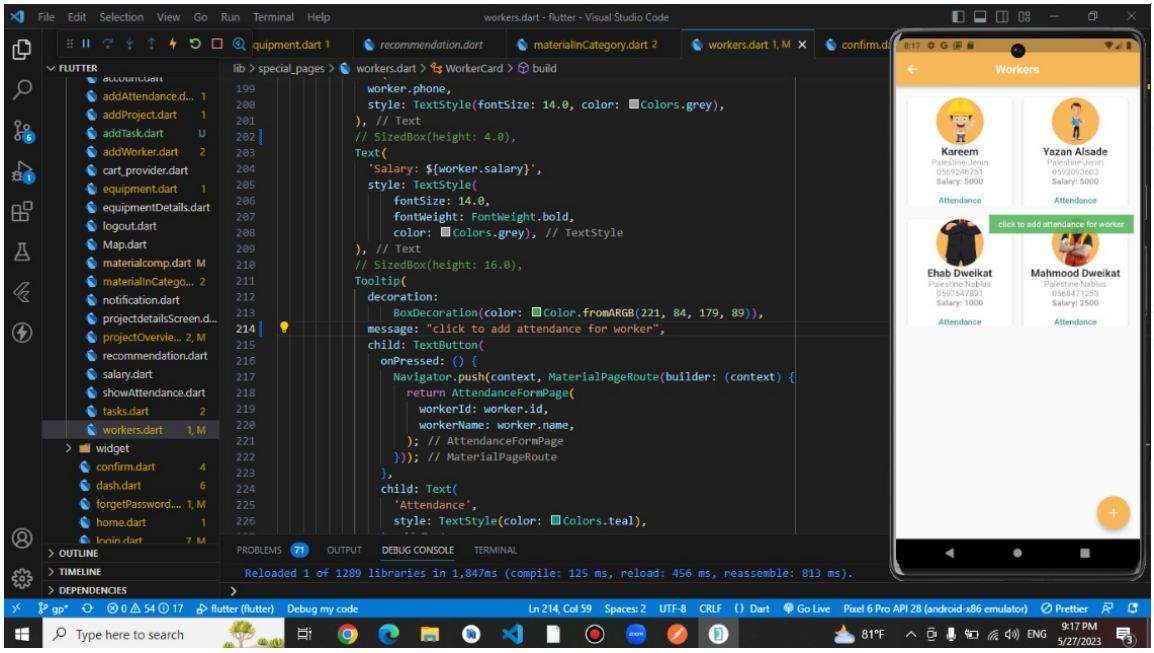


→Worker Page :

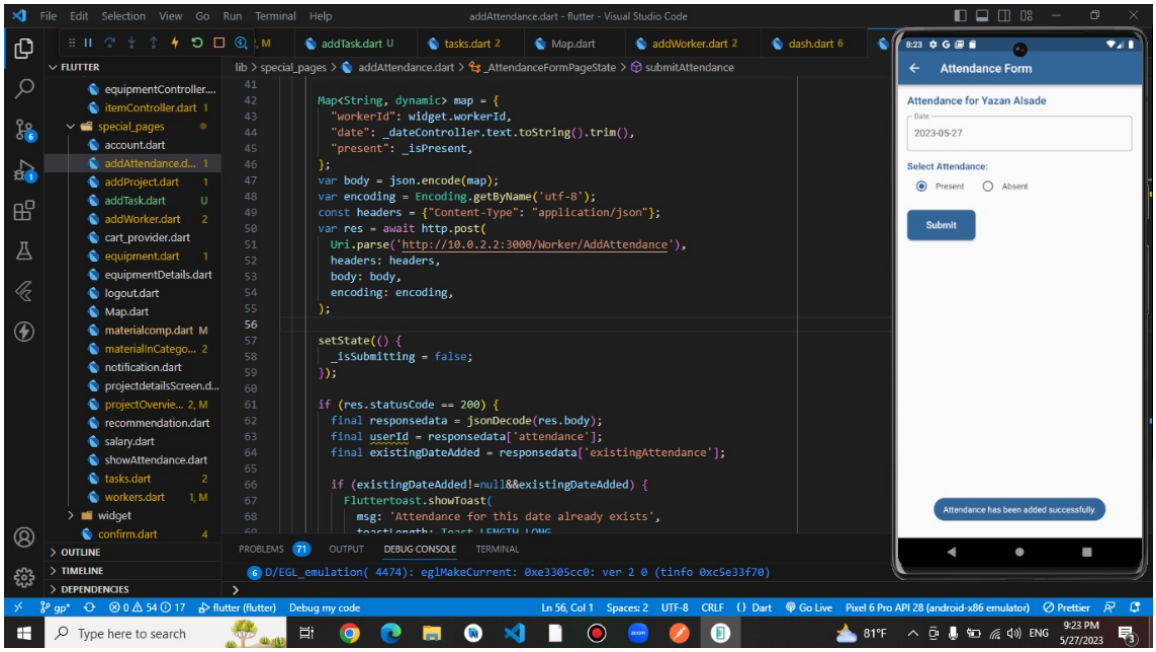
When the engineer clicks on the workers page, it shows him all the workers working in the company and the information of each worker :



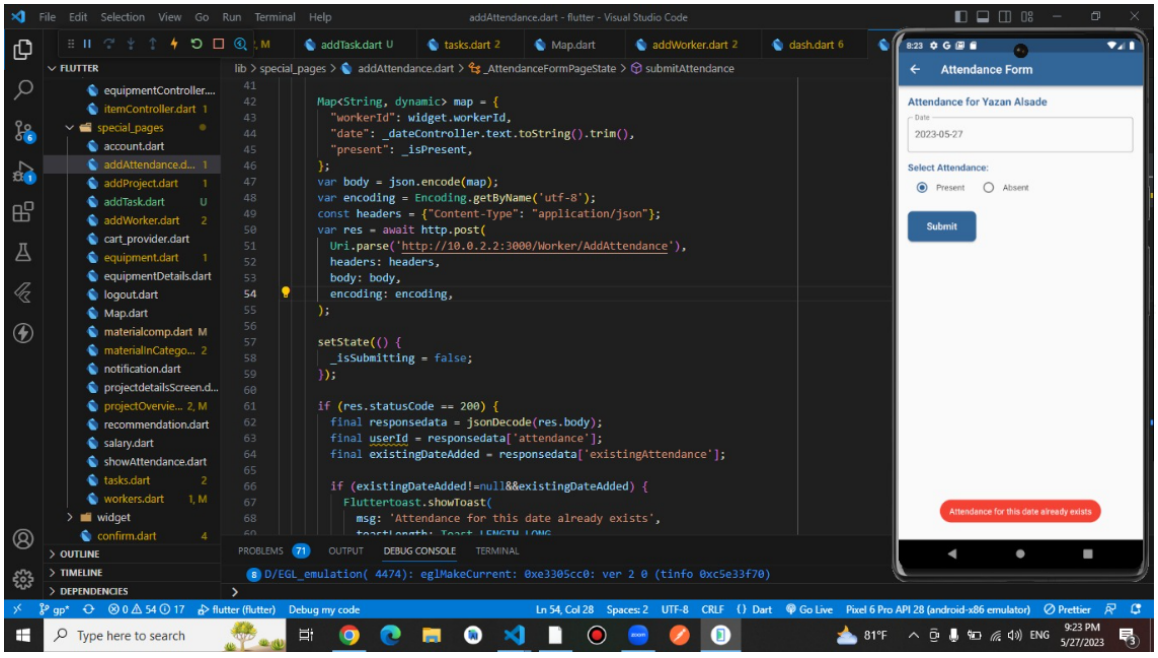
➔ The engineer adds presence and absence to each worker when clicking on Attendance:



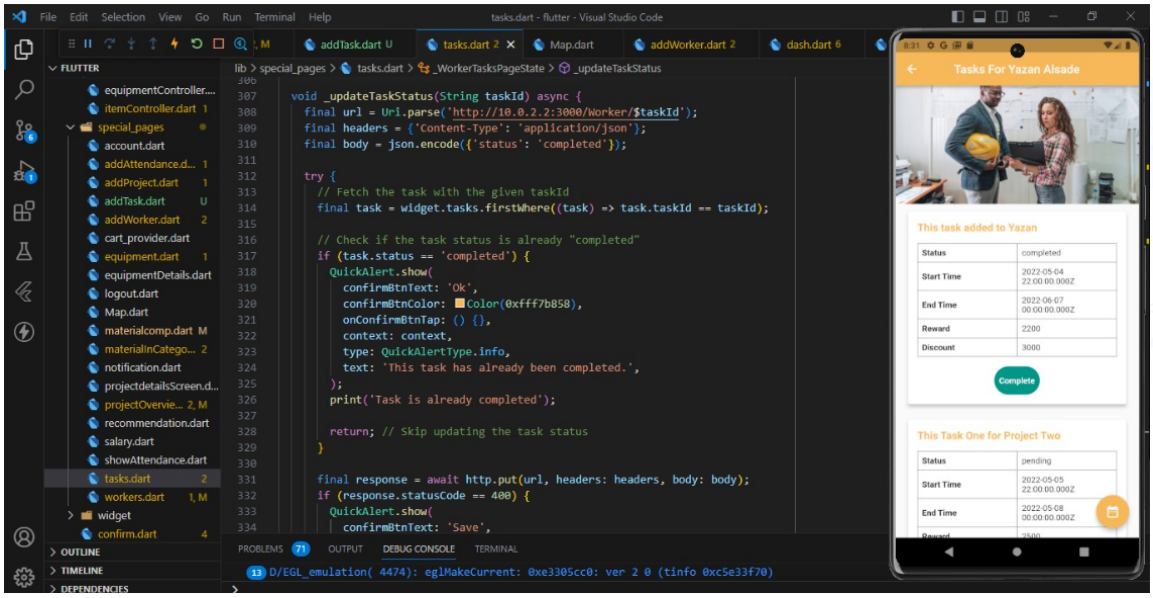
→ Here the engineer enters the attendance and absence of the worker who chose him, depending on whether this worker came to work that day or did not come:



→ Now, if the engineer mistakenly adds the same day of attendance and absence to the same worker, he shows an error message that this day and date already exist.:

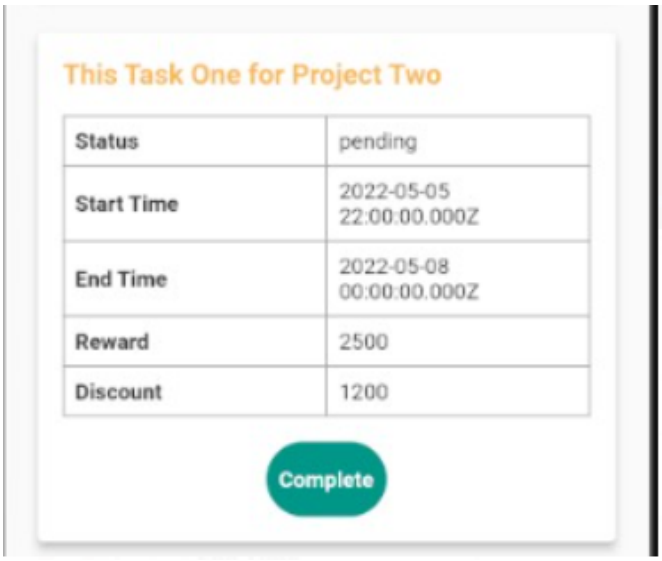


→ When the engineer or worker clicks on his own card, he will see all the tasks that the engineers attached to them so that they can see the details and complete them :

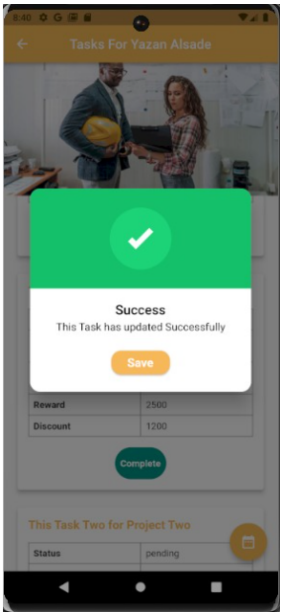


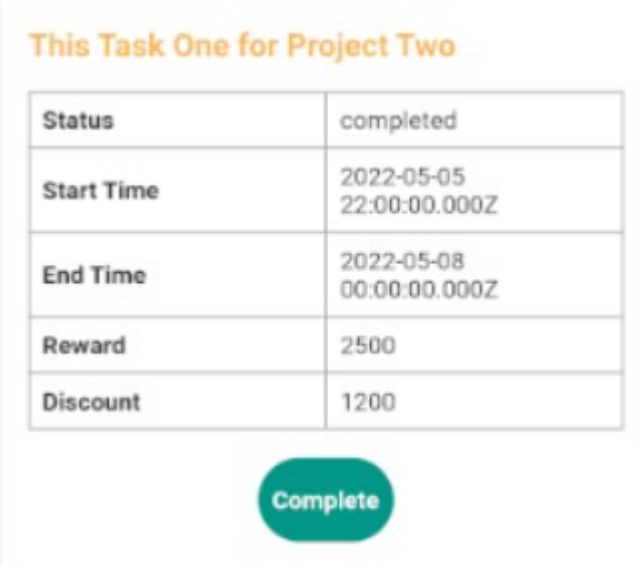
When the worker presses the complete button, it will change its task state to become task complete:

→ Here is an example showing that the Task One state is in Project to Binding:

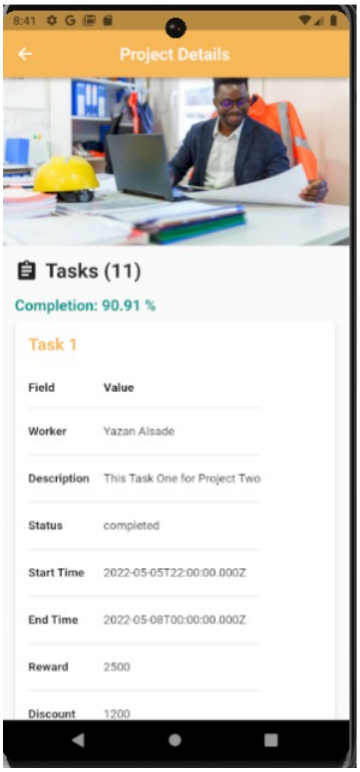


→When pressing the Complete button, the task state changes to Complete:

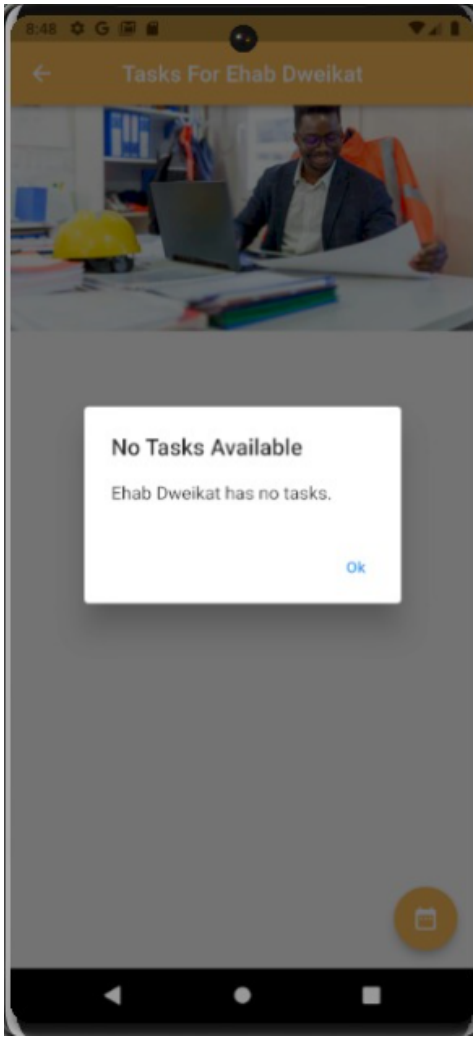




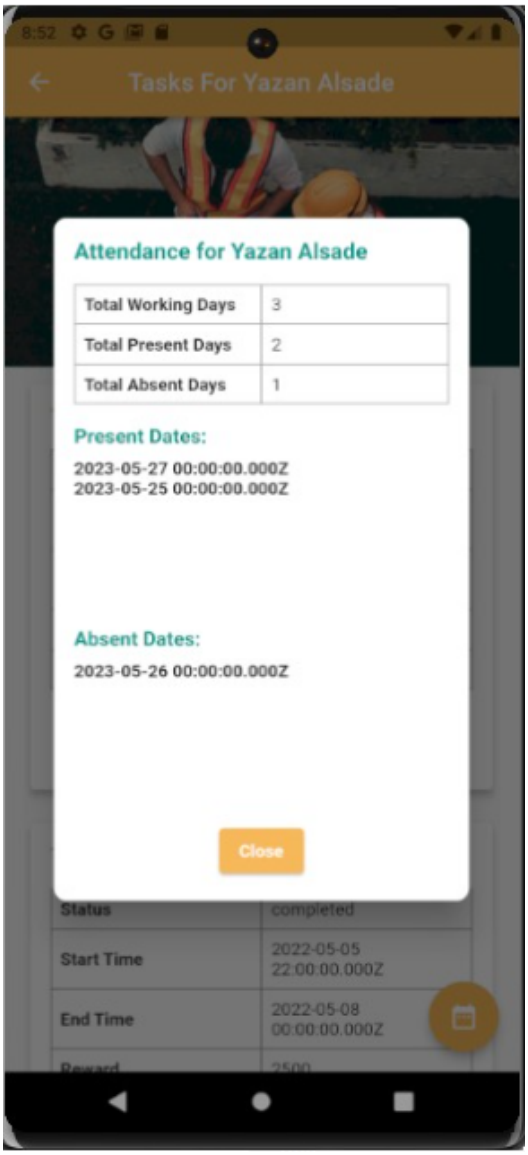
➔ Now, when you return to the projects page and click on Project Two, the percentage of completed tasks in it increases:



→ Now, when you click on a specific worker and there are no tasks for him, it will show him a message for which there are no tasks available for this worker:

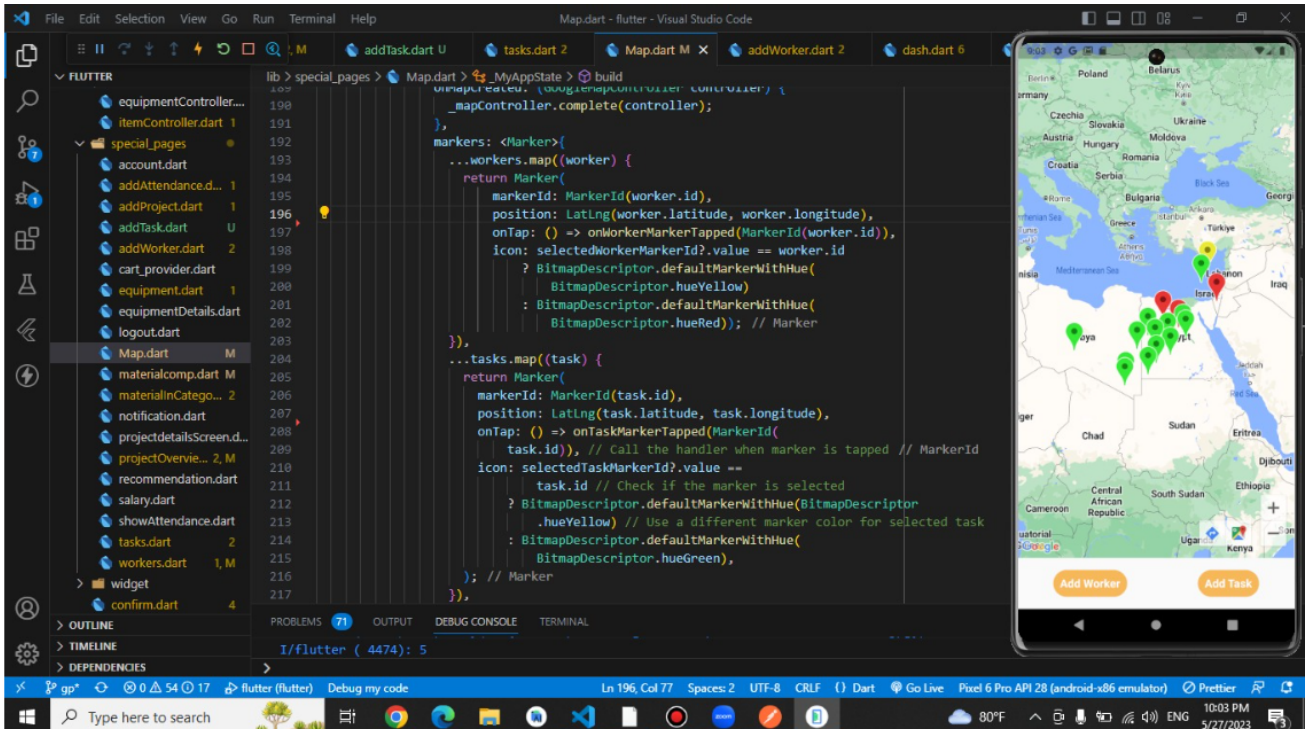


→ When the engineer clicks on the button to see attendance and absence, this window will appear for him to see the total working days for this worker:

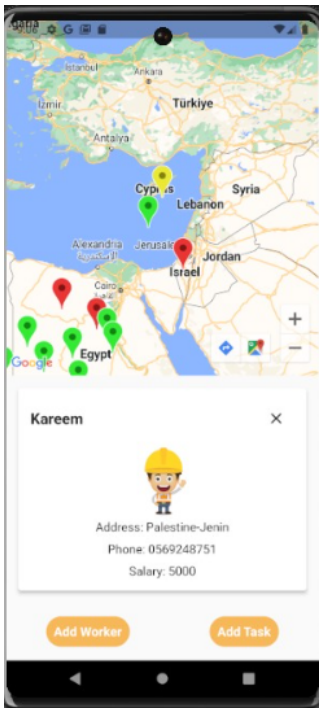


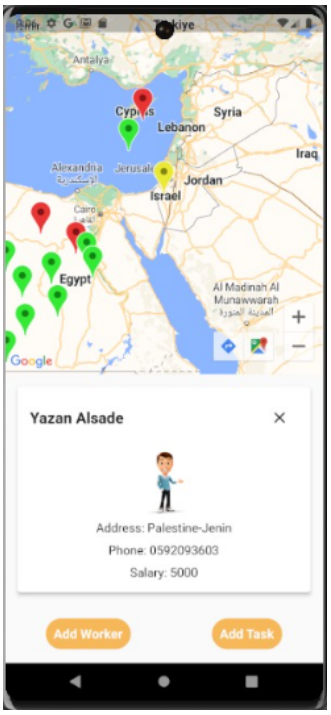
→Google Maps :

- We used Google Maps in our application in order to locate the worker and also to locate the task of this worker, and thus the worker will be able to know the location of his task and go to it to complete it without referring to the administrator to inquire about the location of the task that he will carry out:

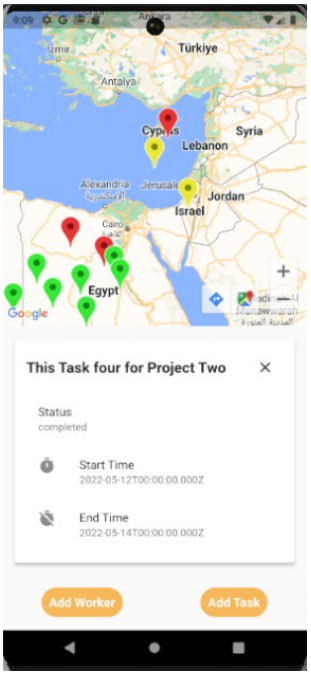


→ When clicking on the red sign that represents the worker, it will show the information of this worker and his location, and then the color of the sign will change to yellow:

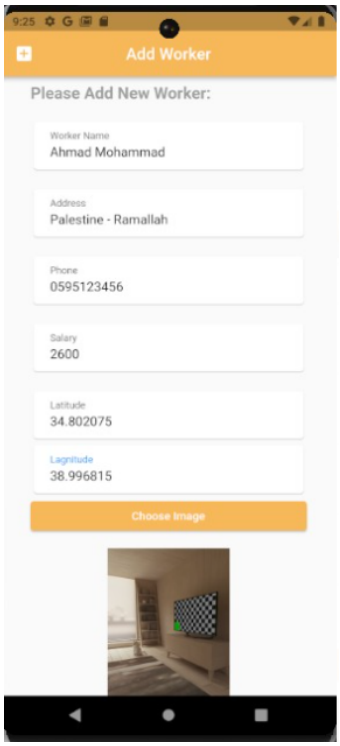


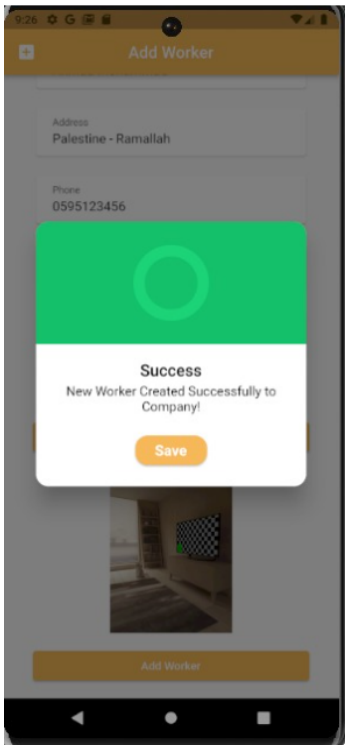


➔ And this is for tasks :

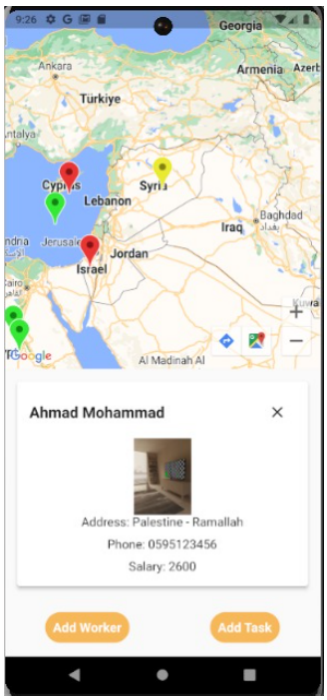


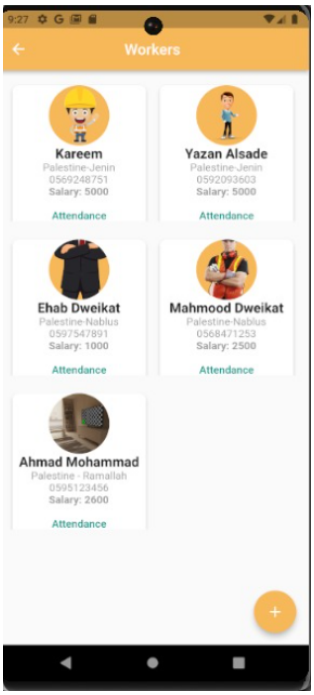
→ Now the engineer can add a worker , and this worker will appear in its location on the map the latitude and longitude for Syria :





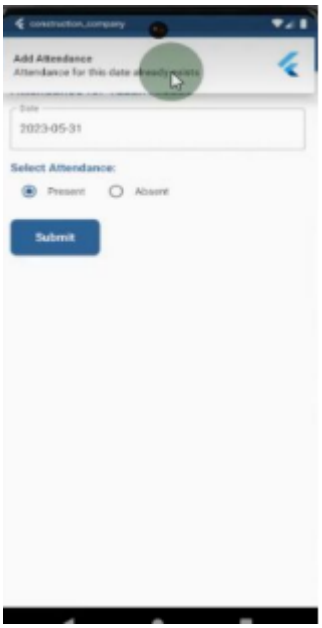
→And Now this worker will appear on the map and worker page :



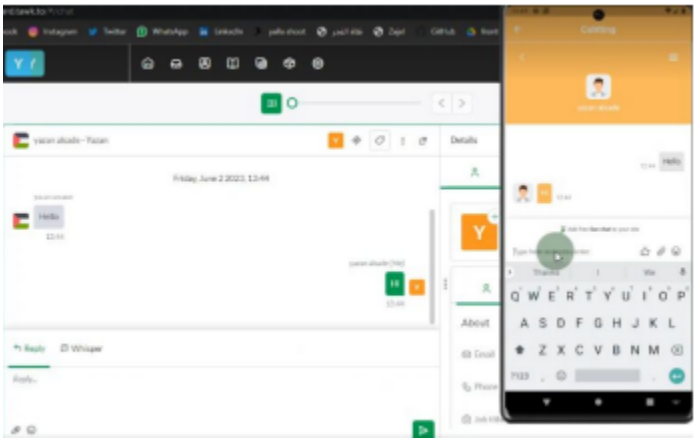


➔

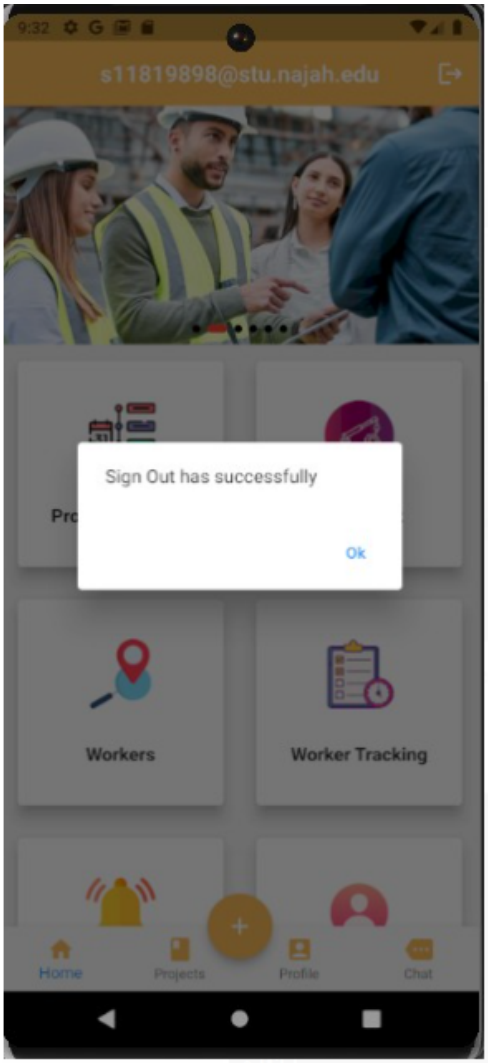
→**Notification System** : Notification systems are designed to let users know when something related to them occurs even if they don't use the application right then. The majority of users do not find it convenient to constantly check what is happening in an application. This makes it necessary to create a notification system to keep them connected to the outside world without having to exert so much effort. In our application and using flutter, Local Notification System was used to make notifications happen



→Live Chat for Technical Support: We have used the idea of live chat in order to provide an opportunity for technical support for engineers in the event that they encounter technical problems, or even to transfer the problems of workers that they face. We have used the Tawk.To library in applying filters



➔ Now The User Can Make Logout From the App:



Chapter 5 : Results and Discussion

We have completed an application aimed at facilitating companies in order to organize their work and save time and effort on them.

5.1 Learning

It takes time, effort, and a lot of research to learn something new. This project taught us a lot, and we employed a lot of new technologies that we weren't familiar with before. For the front end, Node JS, the back end, and the database, we decided to learn and use Flutter. Websites were very helpful to us as we learned these new languages; before to this project, we had no idea how to create a mobile application, so programming tutorials were very beneficial..

5.2 Challenges

It takes a lot of time and effort to look for a dependable source to learn a new programming language. We don't always solve the problems we encounter. The dependency versions of the flutter packages we wanted to use also had a problem.

Chapter 6 : Conclusions and Recommendation

6.1 Conclusion

We gained and were able to have experience in designing mobile pages using flutter and the Dart language, as well as we were able to deal with a new backend language and a new type of database represented in Node.js and MongoDB, and we dealt for the first time with Google Maps as well as with Git and GitHub.

6.2 Recommendations and Future work

There are many features that we want to improve upon and some that we want to add to our software, so we don't want to just stop there

- Notification is done when the engineer adds a task to the worker
- Create a chat between the worker and the engineer, between the worker and the owner of the company, and also between the workers
- Make the admin able to access the application through the mobile phone, not just the web

References

- *Node.js documentation.* Retrieved Jan, 2023 from https://nodejs.org/en/docs/
- https://www.pngegg.com/en/search?q=construction+company+logo
- https://www.flaticon.com/
- https://console.cloud.google.com/google/maps-apis/overview
- https://docs.flutter.dev/get-started/install
- https://www.mongodb.com/docs/manual/core/document/
- https://www.youtube.com/@WaelabohamzaFlutter