AN-NAJAH NATIONAL UNIVERSITY

FACULTY OF ENGINEERING AND INFORMATION TECHNOLOGY

Computer Engineering Department

Software Graduation Project

Fana Abdul Rahim Yamk & Shahd Bilal Hamza

Supervisor: Dr.Haya Samaana

Presented in partial fulfillment of the requirements for a Bachelor's degree in Computer Engineering

May 31, 2023

Acknowledgment

We would like to express our heartfelt gratitude to all those who have contributed to our journey and aided us in achieving the successful completion of this integrated project.

First and foremost, we extend our sincerest appreciation to our beloved families and mentors for their unwavering support, valuable guidance, and continuous encouragement throughout this endeavor.

We are especially grateful to our project supervisor, Dr. Haya Samana, for her invaluable expertise, insightful feedback, and dedicated supervision, which played a crucial role in shaping the project's direction and ensuring its success.

Lastly, we extend our thanks to our friends who provided us with encouragement, constructive suggestions, and helpful tips, contributing to the refinement of our work.

We are truly fortunate to have professors in the Department of Computer Engineering of individuals who believed in our abilities and invested their time and efforts in our personal and academic growth. Their contributions have been instrumental in our achievements, and we are sincerely grateful for their presence in our lives.

Contents

Abstract

We often need help finding maintenance technicians, which can be due to a scarcity of technicians in certain areas, their unreliable attendance for crucial appointments, or uncertainty about their skills. These problems are even more significant for women. An application that addresses all these issues and offers several benefits for both users and technicians is the solution. The application streamlines communication between regular users and registered technicians, enabling users to choose the maintenance they need, view available technicians, and pick their preferred technician based on factors like chat, notifications, and feedback. the user can sign up as a regular user with basic information or as a technician with additional information such as profession, age, experience, and work location. The program features a maintenance services page that covers cleaning, electrical, electronics, and other services, and can be detailed based on the type of service. The admin controls this page through a web page that lists all registered users and technicians.This type of application has begun to emerge as technology infiltrates all aspects of life. Still, there are not enough services available yet, and it has not been implemented in Palestine.

We choose to use the Flutter platform for the front end because it has many libraries and is compatible with various operating systems, and for the back end, we will use Node.js it supports real-time communication and we are already familiar with it.

1. Introduction

Despite the remarkable technological advancements witnessed in our modern era, with artificial intelligence and technology often replacing human capabilities, there remains a significant technological blind spot when it comes to artisans and skilled professionals. Unfortunately, it is the ordinary citizen who suffers the most from this oversight. The distribution of technicians is often inadequate in many areas, despite the pressing need for their services everywhere. In critical situations, individuals are sometimes left with no choice but to rely on professionals without any knowledge of their skills, behavior, or trustworthiness. Furthermore, they lack any information that would enable them to hold these professionals accountable later on. These are all issues that could be avoided if we harness the power of technology.

So, With a firm belief that technology should be primarily utilized to address people's daily and urgent needs, we embarked on developing this application. It serves as a user-friendly interface, seamlessly connecting professionals with those seeking their services, thereby benefiting both parties. Given that approximately 54% of the Palestinian workforce is engaged in skilled trades, the primary objective of this application is to uplift and advance the Palestinian economy. It empowers professionals by granting them greater visibility and opportunities beyond what they could achieve independently. Simultaneously, it enables citizens to choose professionals based on their preferred criteria, such as pricing, good conduct, skills, and more. Furthermore, the application encourages constructive feedback, serving the interests of both the citizens and professionals, fostering excellence and responsible behavior, In addition to that, it allows users to explore new services they were not aware of, increases competition among professionals, which benefits the country's economy, and ensures complete transparency between the user and the professional.

The application offers a unique advantage for Palestinian women who are typically at the forefront of tackling technical issues at home. However, they often face numerous challenges when it comes to accessing technicians, lacking the experience to select reliable ones and placing trust in their services. Conversely, the application plays a role in creating employment opportunities for women by enabling them to showcase and provide suitable services.

Within this report, we will showcase our implementation for this project. It begins with our ideation process, drawing inspiration for the concept. We then infuse it with our own touch, building upon previous attempts while incorporating various methods and tools. As a result, we obtain valuable outcomes through this application. Subsequently, we analyze these results and provide a succinct overview of our experience with the project.

2. Constraints, Standards and Earlier Coursework

2.1. Constraints Limitations

Embarking on a comprehensive and ambitious project inevitably comes with initial difficulties. Our journey commenced with the lofty objective of developing an integrated project encompassing mobile and web front-end as well as back-end components. One of the challenges we faced was the utilization of the Mongo database, which required a fast and stable internet connection. To address this, we implemented various optimization techniques in the backend code to improve data retrieval speed and ensure smooth communication between users and technicians.

Another constraint we encountered was the seamless transfer of work between team members. To overcome this challenge, we leveraged collaborative tools such as GitHub. This platform facilitated simultaneous work on different project components, enabling us to track progress and coordinate effectively.

2.2. Standards

The development of both the web and mobile applications utilized the Flutter platform as the chosen frontend framework for all user categories, including admin, workers, and regular users. This allowed for a consistent and unified user experience across different platforms. As for the backend, we opted for Node.js as the backend technology, ensuring efficient and seamless communication between the front end and the database.

2.3. Earlier coursework

The knowledge and skills acquired during our previous coursework proved invaluable in the successful execution of our project. Concepts such as software development methodologies and database management played a crucial role in guiding our approach and informing our decision-making process. Despite the initial constraints and challenges we encountered, we take great pride in the fact that we were able to develop the mobile application and web pages through self-learning, utilizing freely available online resources.

By overcoming these obstacles, we not only achieved our project goals but also experienced substantial professional and personal growth, gaining valuable insights and expertise along the way.

3. Literature Review

Extensive literature has addressed the pressing need for efficient and dependable maintenance services, particularly within the realm of mobile applications. Numerous studies have highlighted the challenges users face in finding suitable technicians, including the scarcity of technicians in specific areas, their unreliability in attending appointments, and uncertainties regarding their skills. Gender-related barriers, especially for women, have also been identified in accessing reliable maintenance services. In response to these challenges, mobile applications have emerged as promising solutions.

TaskRabbit[1] and Handy[2] are noteworthy examples of successful home service apps that have revolutionized the process of finding and hiring maintenance technicians. These platforms have introduced user-friendly interfaces that streamline communication between users and technicians, empower users to select services, and provide mechanisms for feedback. These applications have proven to enhance convenience for users and improve service delivery efficiency.

However, it is crucial to note that the implementation of such applications in Palestine is limited. Therefore, this study aims to contribute to the existing body of knowledge by developing a mobile application specifically tailored to the Palestinian context, taking into account the unique challenges and requirements of the local maintenance service industry. A distinctive feature of our app is the provision for workers to select multiple jobs, creating more opportunities for them while simultaneously offering users a broader range of options to choose the most suitable worker based on various criteria.

4. Methodology

4.1. Tools, Methods and Programming Languages

4.1.1. Tools

Visual Studio Code was the primary IDE used for writing and executing the code. It provided essential features and tools for code editing, debugging, and project management.

Android Studio was utilized as an emulator to test the mobile application on different virtual devices, ensuring compatibility and smooth functionality.

We utilized GitHub for collaborative development, code management, and tracking changes.

Firebase was integrated into our project to leverage its real-time database and cloud storage capabilities. This allowed us to securely and efficiently store messages and images. We also utilized Firebase's messaging service to enable notification functionality, enabling communication from the web to mobile and between mobile devices.

Render was employed as a hosting platform for our APIs. It provided a reliable infrastructure for deploying and serving our APIs with fixed endpoints, eliminating the need to rely solely on the local host environment. This ensured consistent access and improved the overall stability of our application.

Thunder Client plugin as our API testing tool, enabling us to test and validate our APIs. With its user-friendly interface, we could send HTTP requests and receive responses.

4.1.2. programming language

The mobile application and web pages were developed using a combination of programming languages:

For the frontend, we utilized Dart and the Flutter framework. Dart is a language specifically designed for building user interfaces, while Flutter provided a rich set of tools and widgets for creating visually appealing and interactive mobile applications.

Node.js was chosen as the backend programming language. It is a popular runtime environment that enables server-side development and provides efficient handling of asynchronous operations, making it suitable for building robust and scalable web applications.

4.1.3. Database

For our project, we utilized MongoDB Atlas as the chosen database solution. MongoDB Atlas is a fully managed cloud database service that offers scalability, reliability, and security. It provided the necessary infrastructure to store and manage the project's data effectively.

Several tables were created in the MongoDB Atlas database to accommodate different aspects of the application, including

Admin Table:

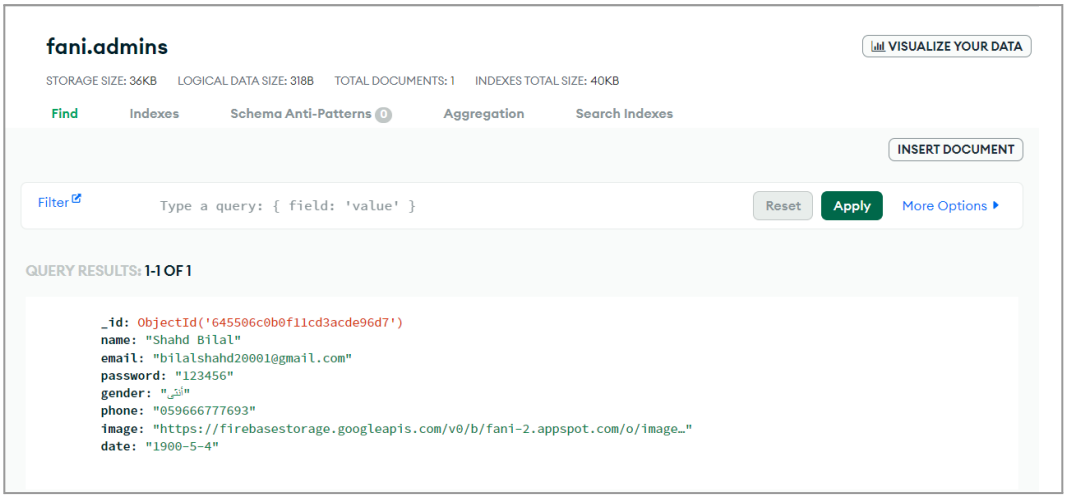


Figure 1: Admin Table

Worker Table:

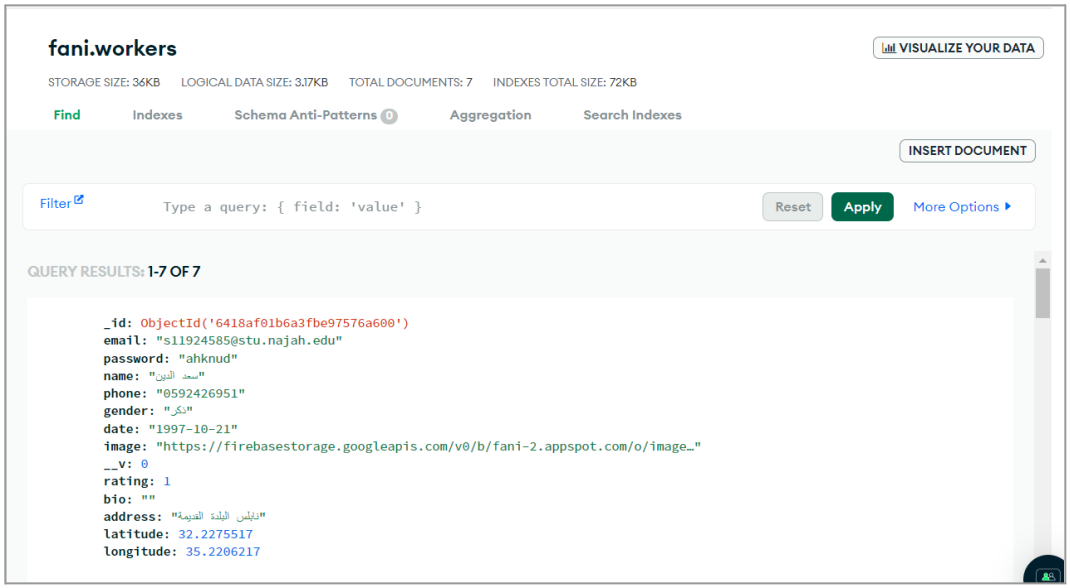


Figure 2: Worker Table

User Table: In addition to storing basic information related to users, an additional attribute called "pref" was added to the user table to represent user preferences. This attribute captures the preferences that users have when selecting a worker for their desired services.

For example, the "pref" attribute can include criteria such as the worker's rating, price, and punctuality. Users can indicate their preference for a worker who has the highest rating, offers a reasonable price, and is known for being punctual.

By incorporating this preference attribute, the application can better match users with workers who align with their specific requirements and preferences. This enhances the user experience by ensuring that the selected worker meets the user's desired criteria, resulting in increased satisfaction and a more tailored service.

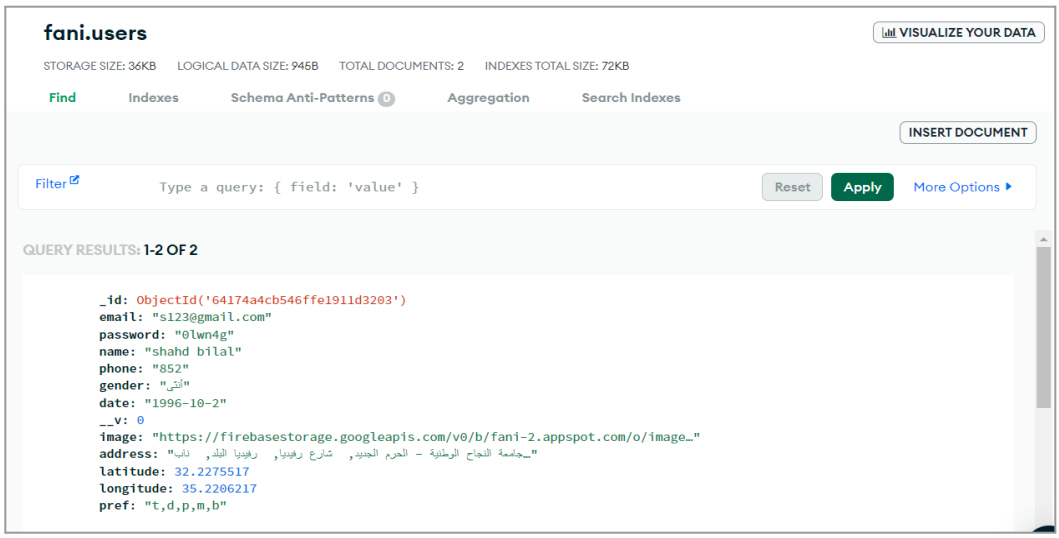


Figure 3: User Table

Type Tabel: The "hm" attribute captures whether the service is priced based on hours or meters. This allows for flexibility in pricing based on the nature of the service. For example, some services may be charged per hour, while others may be priced based on the area covered (in meters).

Figure 4: Type Table

Servs Tabel: contains information about the main maintenance services offered through the application

Figure 5: Servs Tabel

Servworks Tabel: The service-worker table serves as a linkage between workers and the specific services they provide. It enables the association of multiple services with each worker, along with relevant details such as service price and working hours. Furthermore, the table incorporates variables that allow users to evaluate and provide feedback on the worker's performance. These variables include ratings and assessments related to the worker's proficiency, reliability, and adherence to scheduled appointments.

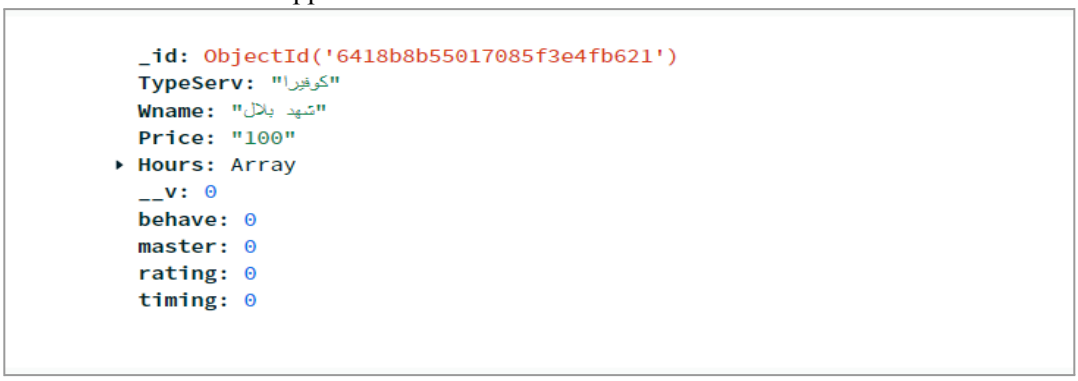


Figure 6: Servworks Table

Ords Table: This table was responsible for storing information regarding user requests and orders

Figure 7: Ords Table

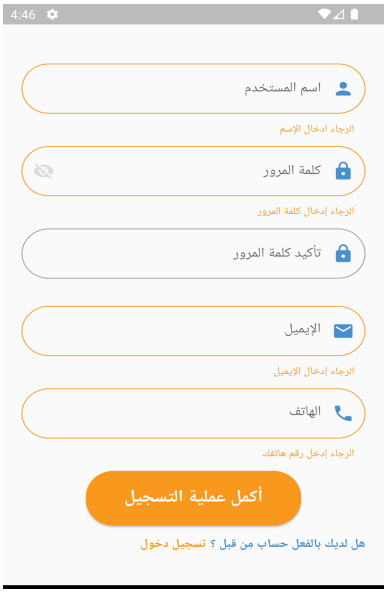
4.2. System Features Implementation

4.2.1. Mobile Application

The mobile application, named "Fani," was developed to facilitate seamless interaction between users and workers, ensuring ease of communication and clarity in their interactions.

Figure 8: Main App Page

- **sign-up page** the user can register to the application with the required information, and determine if he wants to register as a technician or stay as a client.



sign-up page has been implemented for both workers and users, ensuring that the required information is entered accurately. The page is designed to match the provided visual representation, with additional error handling to address various scenarios. This includes detecting and preventing cases where important fields are left blank, weak passwords are used, or incorrect email addresses are entered, etc.

Figure 10: Sign up Page

Figure 10.1: empty field check

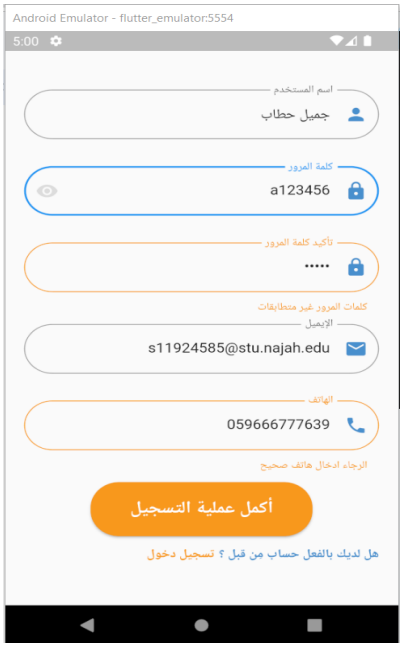


Figure 10.2: password checks

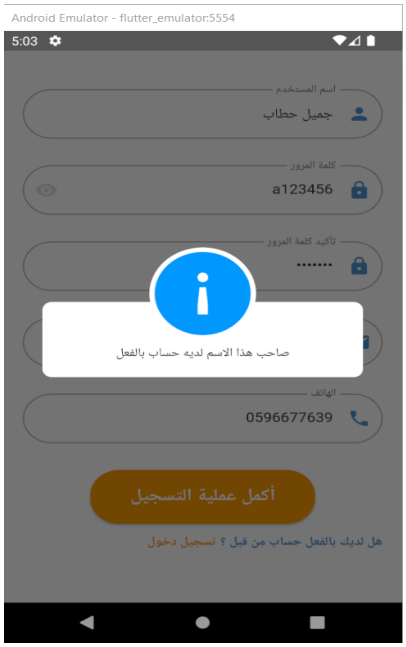


Figure 10.3: name check

after entering the correct information the user can select if he needs to register as a worker or as a client:

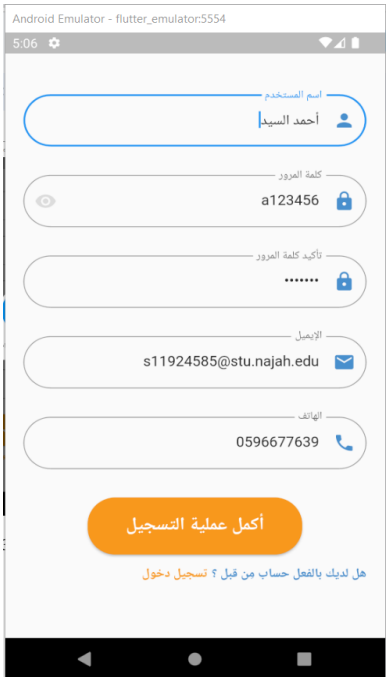


Figure 10.4: correct field

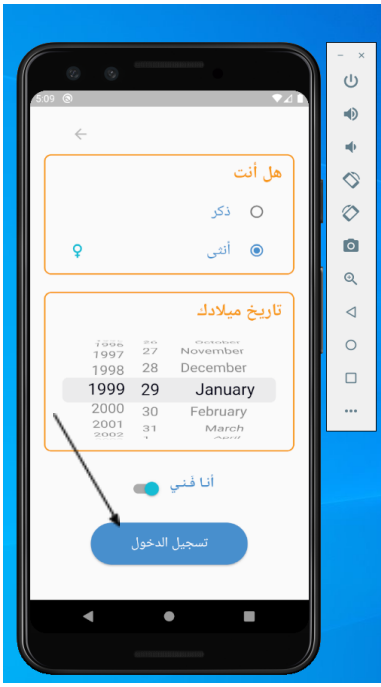


Figure 10.5: as user

Figure 10.6: as worker

after signup correctly from user and worker the system send to their notification and convert them to log in pages: for the login page, the user can only log in if he was previously registered either as a technician or client

Figure 11: log-in page

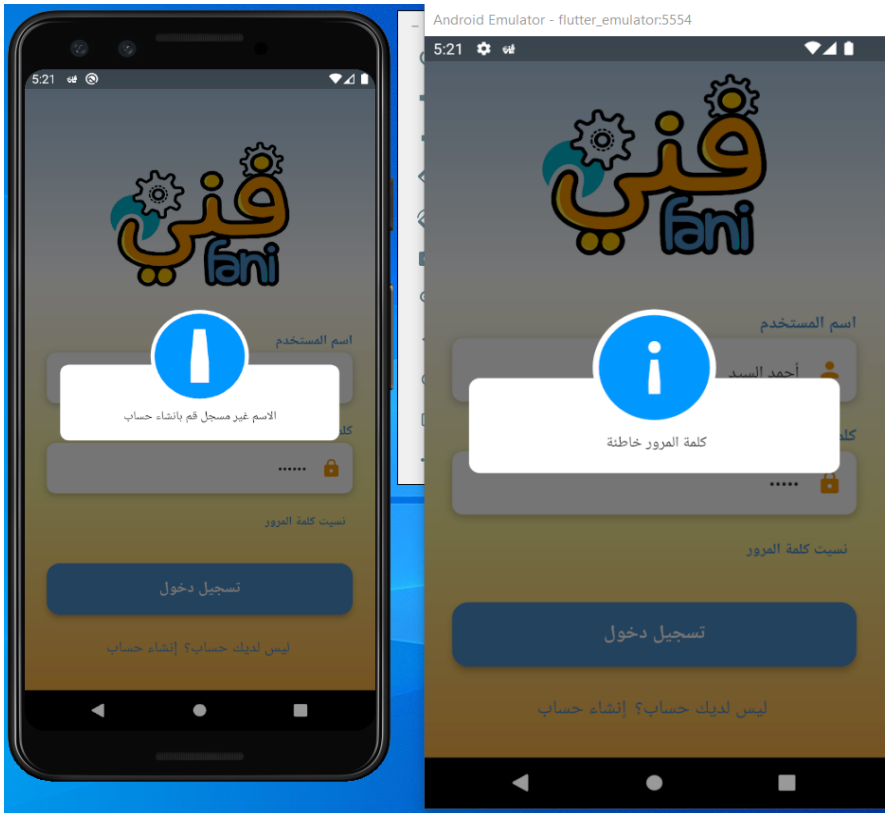


Figure 11.1: Handel log-in

forget password feature for workers and users:

Figure 12: forget password page

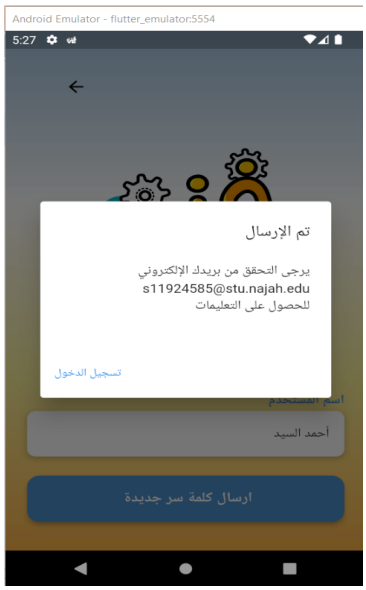


Figure 12.1: email-password

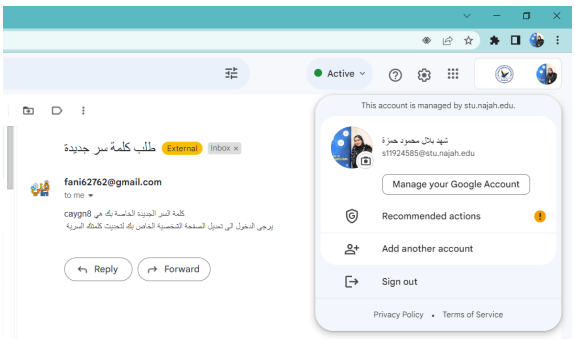


Figure 12.2: send password

when login correctly the notification send from system to the worker and user as shown:

Figure 13: login correctly as client & as worker

4.2.1.1. client side

- home page

The home screen for the client contain the available services, each showing the range of prices for the available technician if any exist, in addition to endDrawer

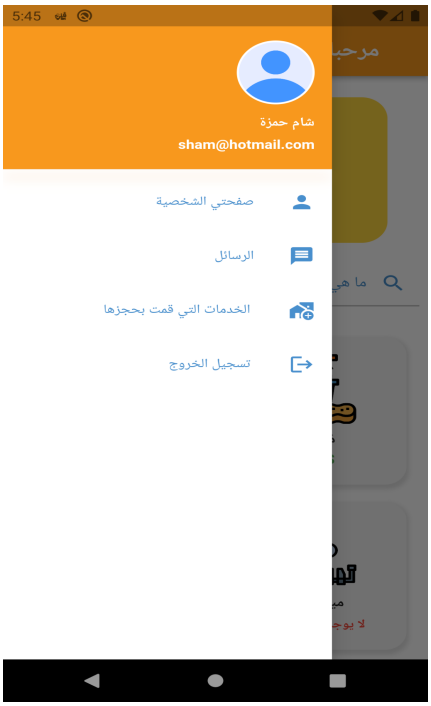


Figure 14: user endDrawer

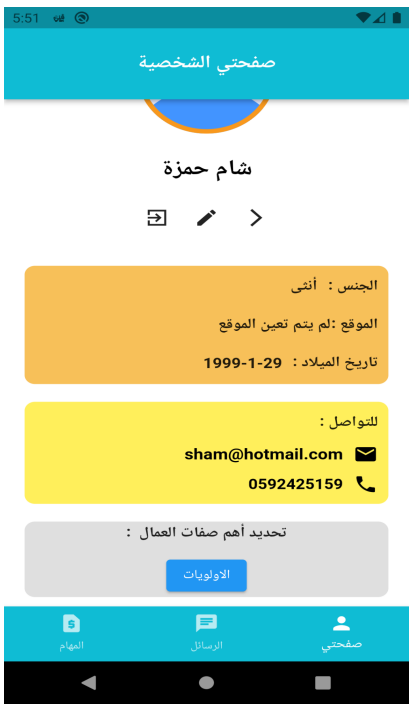


Figure 15: user profile

for making an order follow the steps:
select the main service type [only one is possible] , then choose the specific type of service [as many as needed].

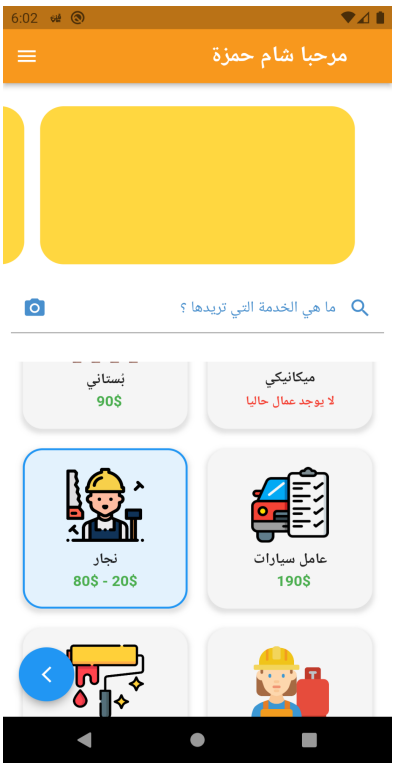


Figure 16: order steps

Figure 16.1: service

then select the date and time, enter all your free hours, and finally you can choose the service to be repeated as needed

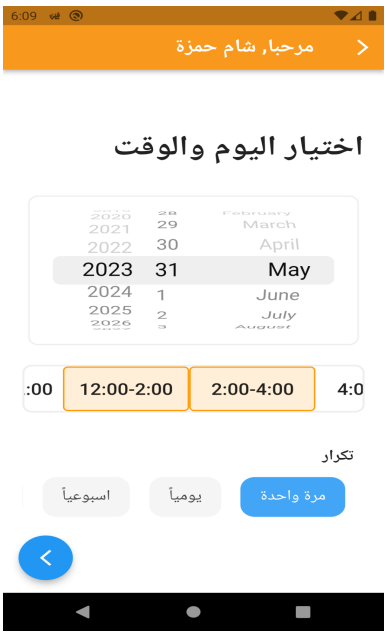


Figure 16.2: date and time

then a list of workers will appear, for the first part the appear the workers who are free in one or more of the selected hours previously, all of them can be displayed and ordered, either by price or general rating or using general ordering
the general ordering depend on the list of preferences which the user can determine, and order the worker depending on the matching with these preferences
the second part include the workers who are not free but match the top three preferences with the highest rate.

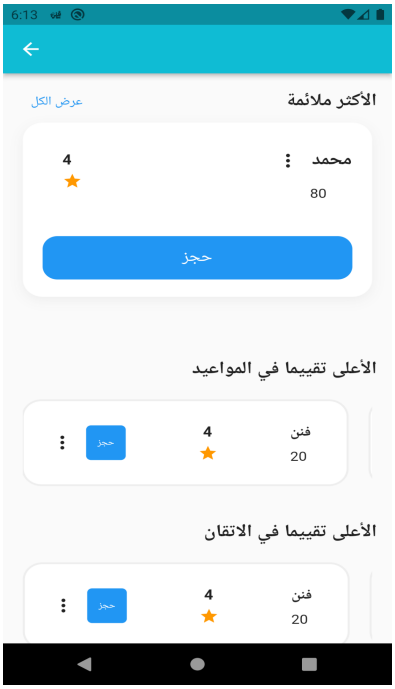


Figure 16.3: workers options

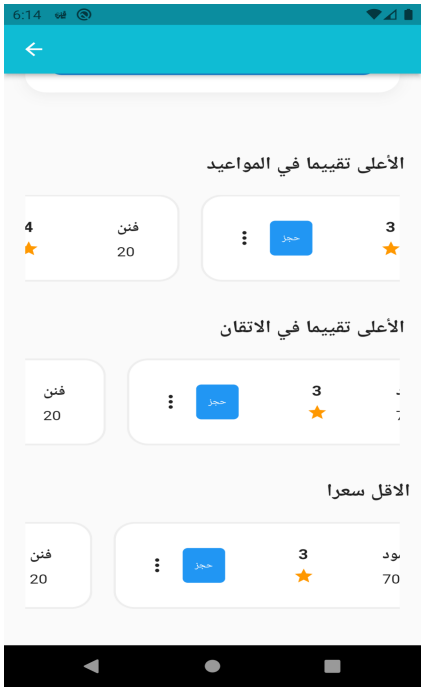


Figure 16.3.1: workers rates

he can navigate to see other less-rated workers.

see all button:

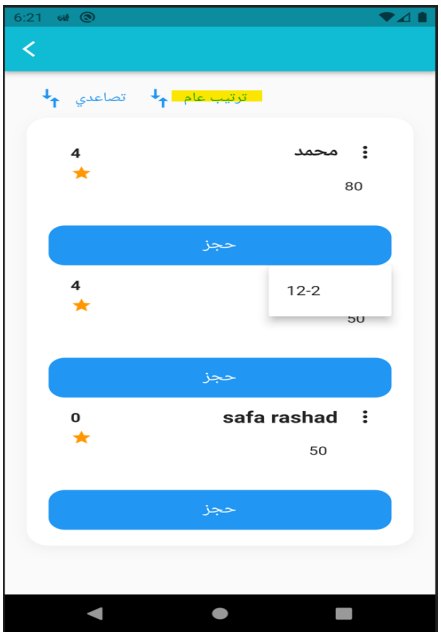


Figure 16.3.2: workers ranking asc

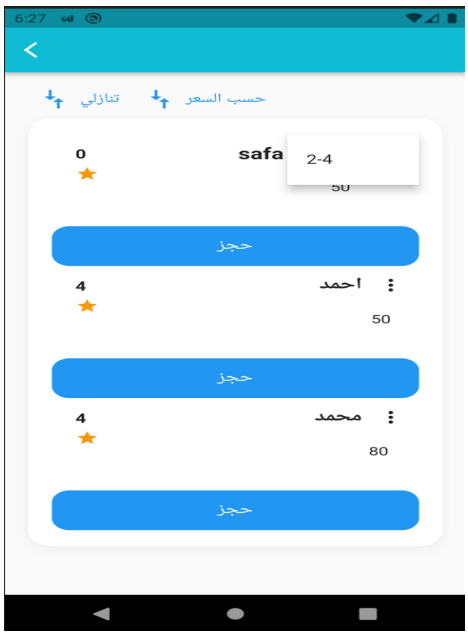


Figure 16.3.3: workers ranking desc

Figure 17: Select hour

Figure 17.1: Confirmation

after that the order and the notification will be sent to the technician if he accepts it, it will appear on the requests page for the client as shown:
* worker when accepts or deny the order the notification also will be sent to the user.

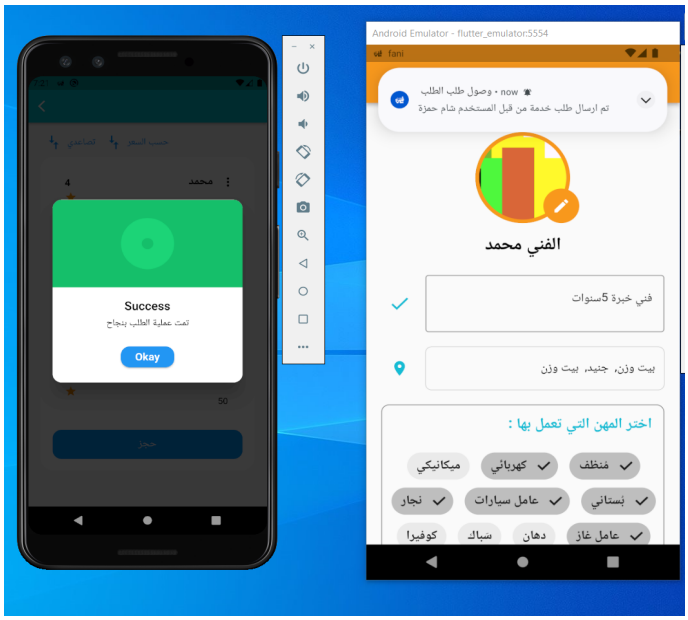


Figure 18: success order

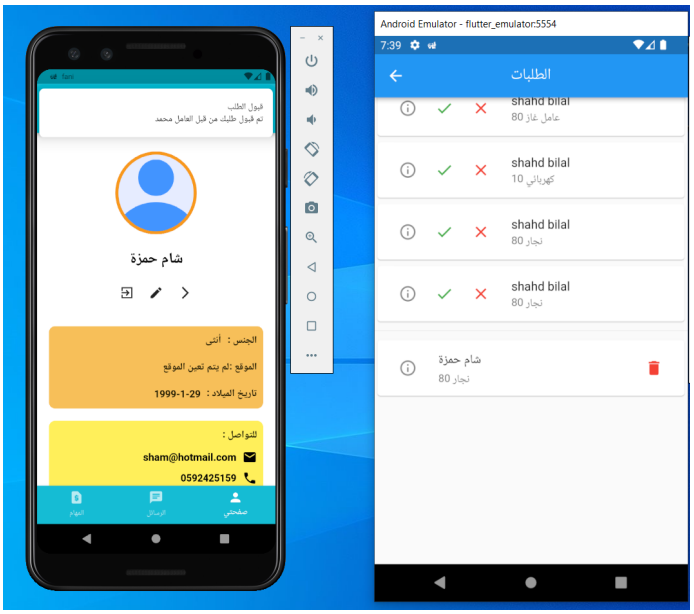


Figure 18.1: agree worker

When it is done the client is asked to rate according to some factor and the general rating will be calculated

Figure 19: user dashboard

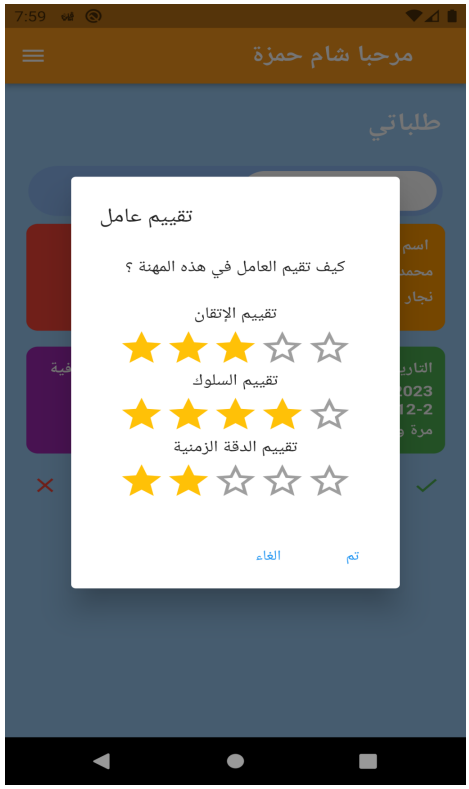
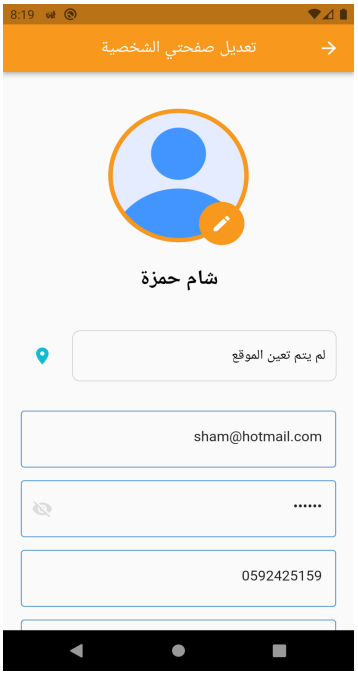


Figure 19.1: user rating worker

all other orders are stored according to its situation.

Figure 19.2: previous order

user edit profile page:



it can be updated his information any time

Figure 20: user edit profile

edits user location using map:

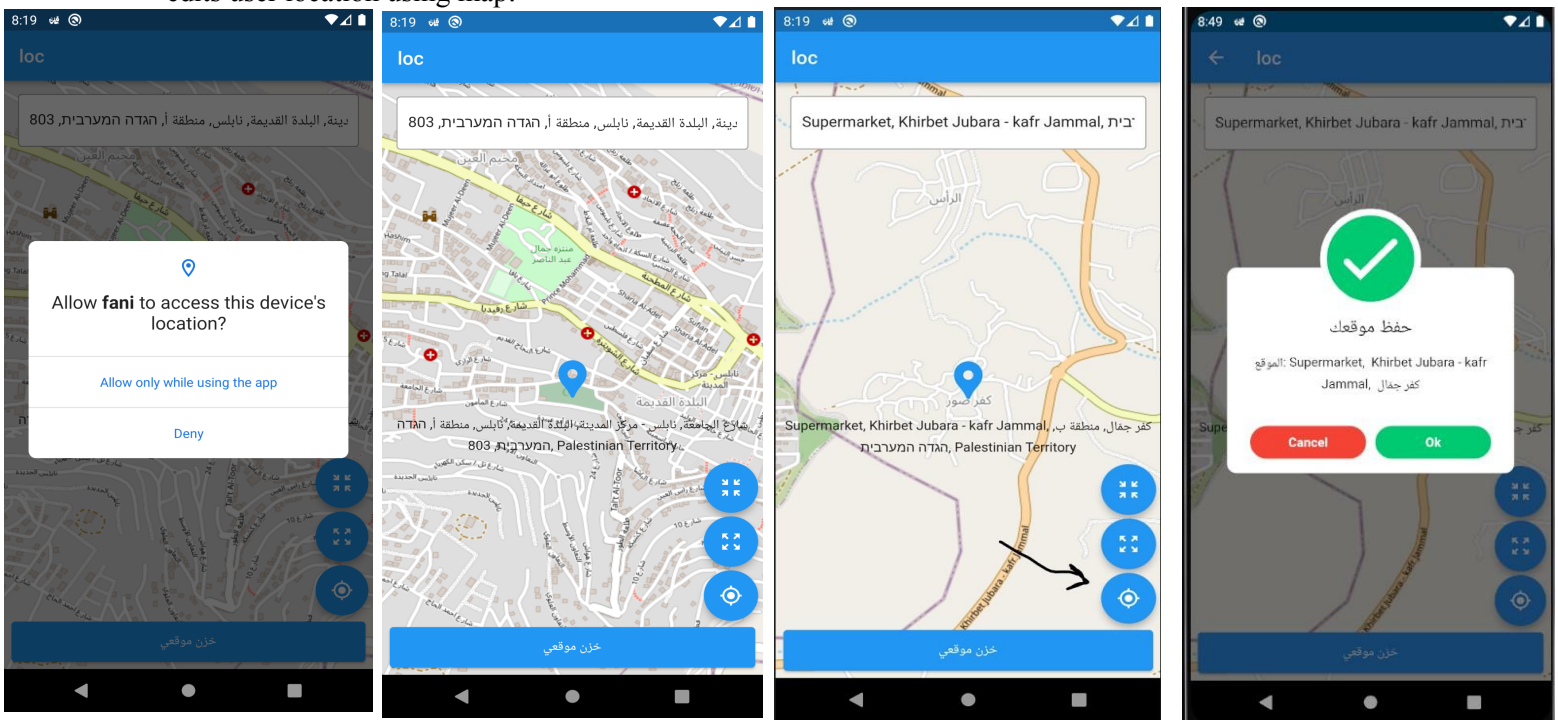


Figure 20: user map

edit user profile image:

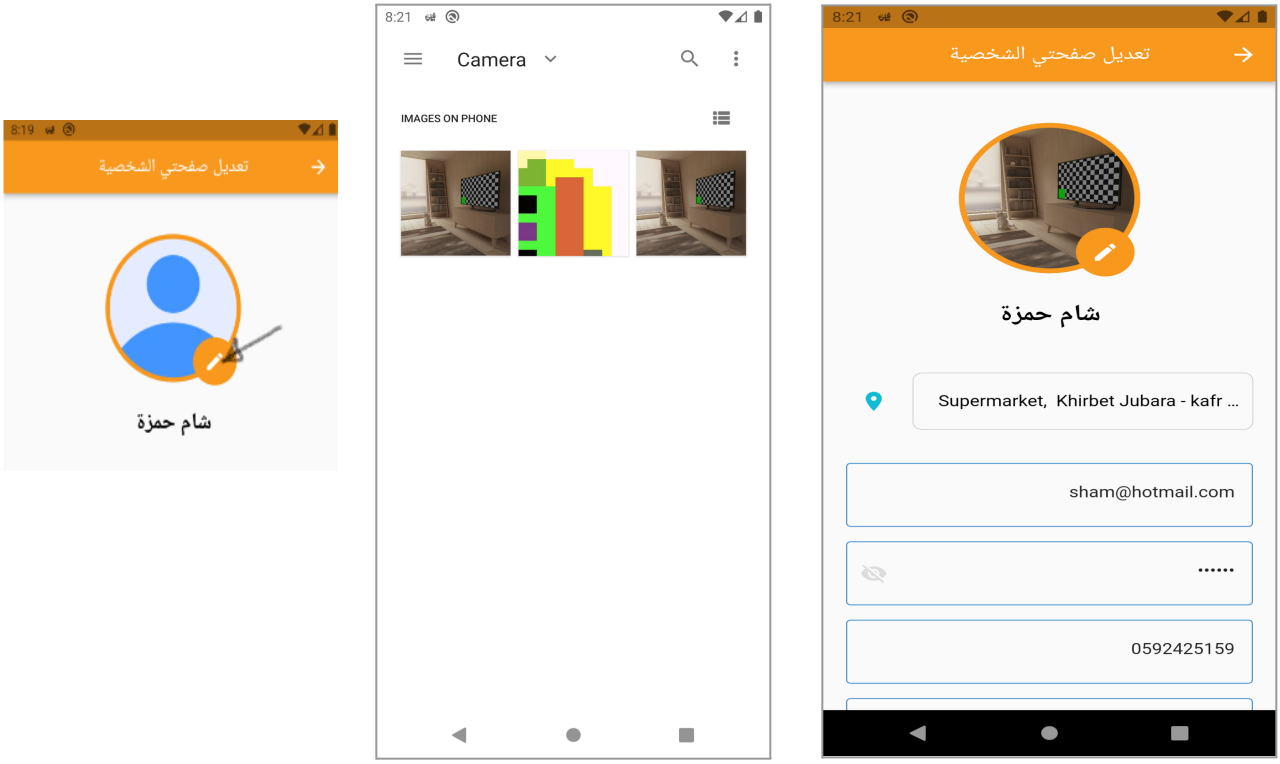


Figure 21: user profile image

edit user date of birth:

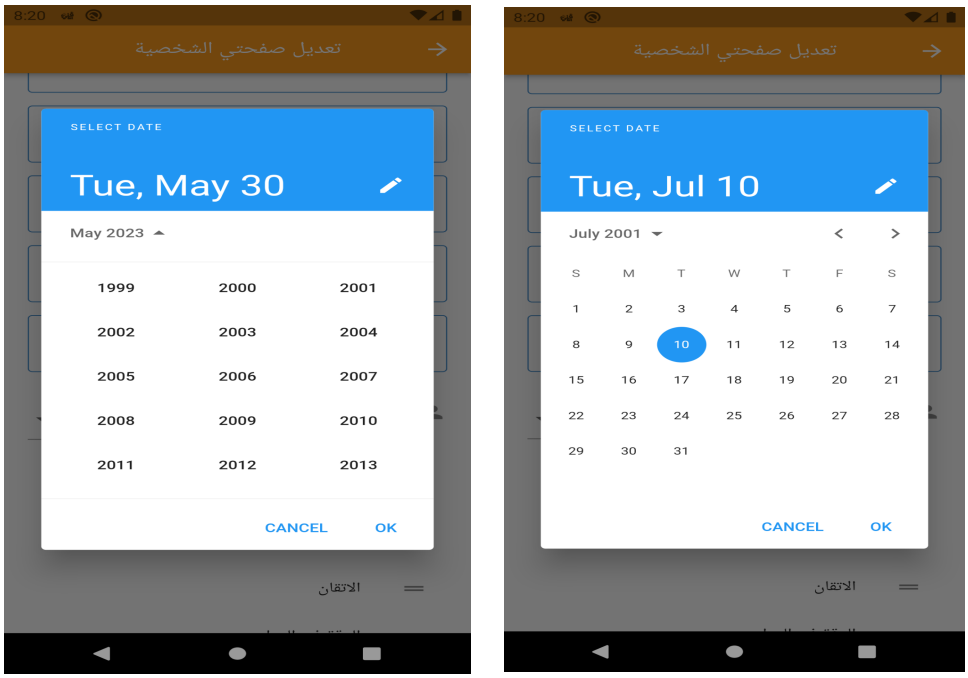


Figure 22: user date

if the location was not accurate using the map the user can edit it manually

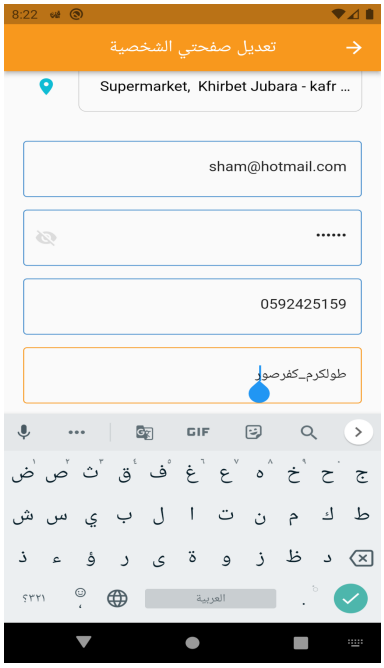


Figure 23: user edit location

the changes will appear on his profile

Figure 23: user update profile

A chat feature has been integrated into the application, allowing clients to communicate with registered workers. Users can initiate conversations, send messages, and receive responses from workers through the chat service. Additionally, users have the ability to send messages to the admin for any inquiries or assistance they may require. This feature enhances communication and facilitates seamless interaction between users, workers, and admin.

* user can see the profile of any worker on a system

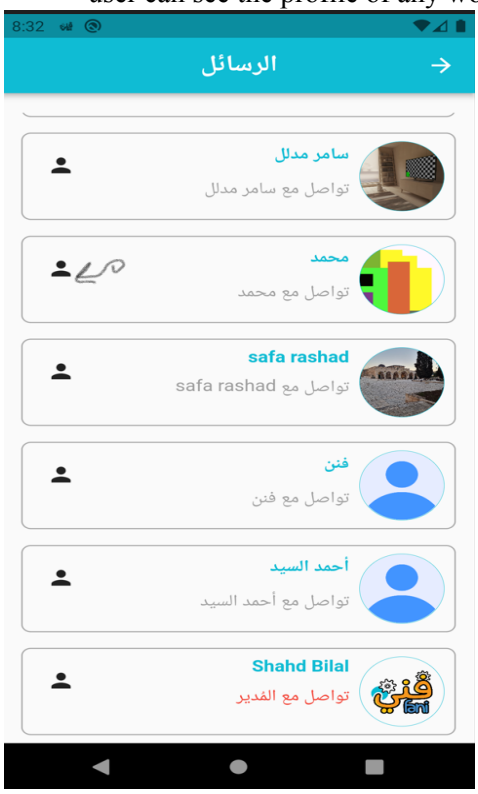
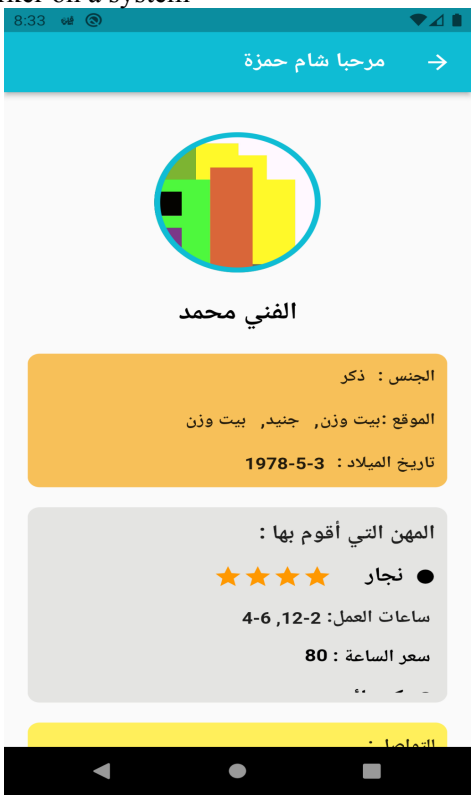


Figure 24: user chat member



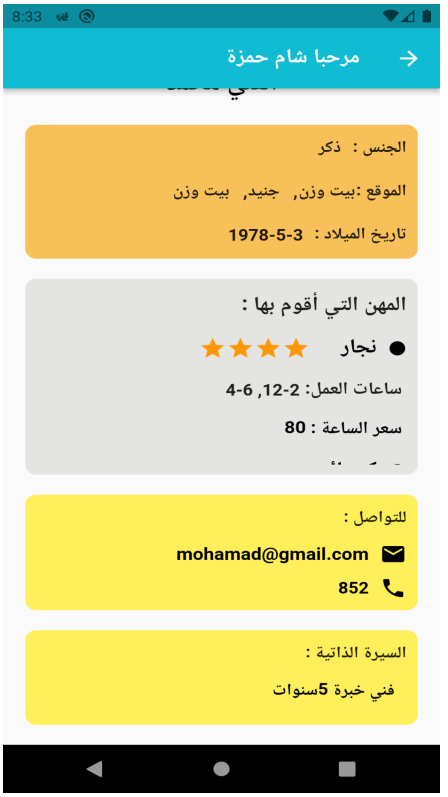


Figure 24.1: user see worker

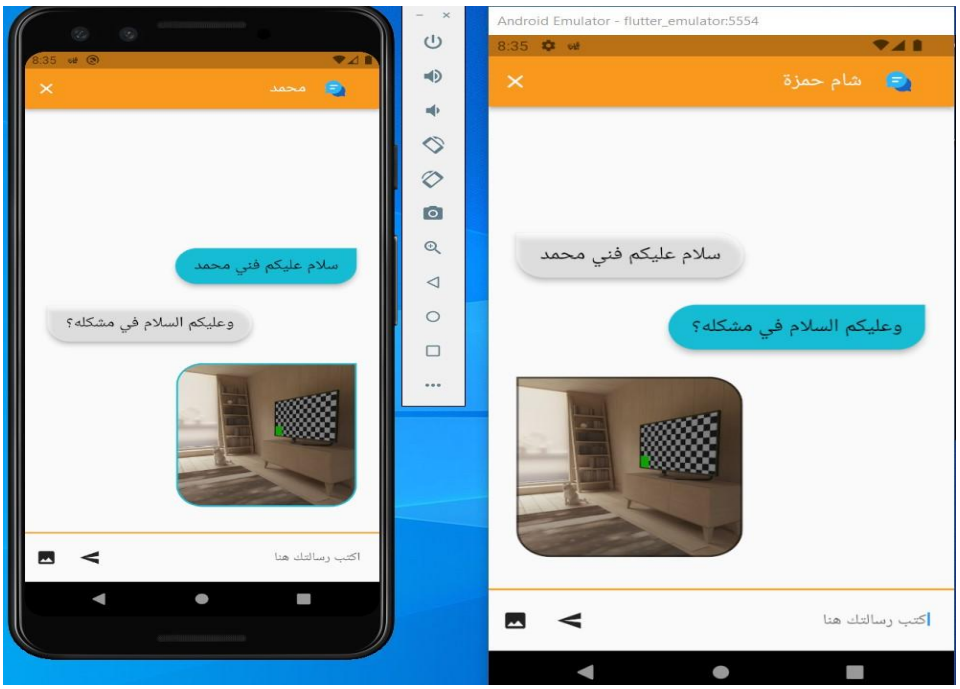


Figure 24.2: user worker chat

4.2.1.2. worker side

when the user registers as a technician, it will enter its edit profile page where he can see and edit his information.

the technician can change his information, and select to work in more than one service where he needs to select the price and the working hour

Figure 25: worker edit page

Figure 25.1: worker edit hours

edits worker location using map:

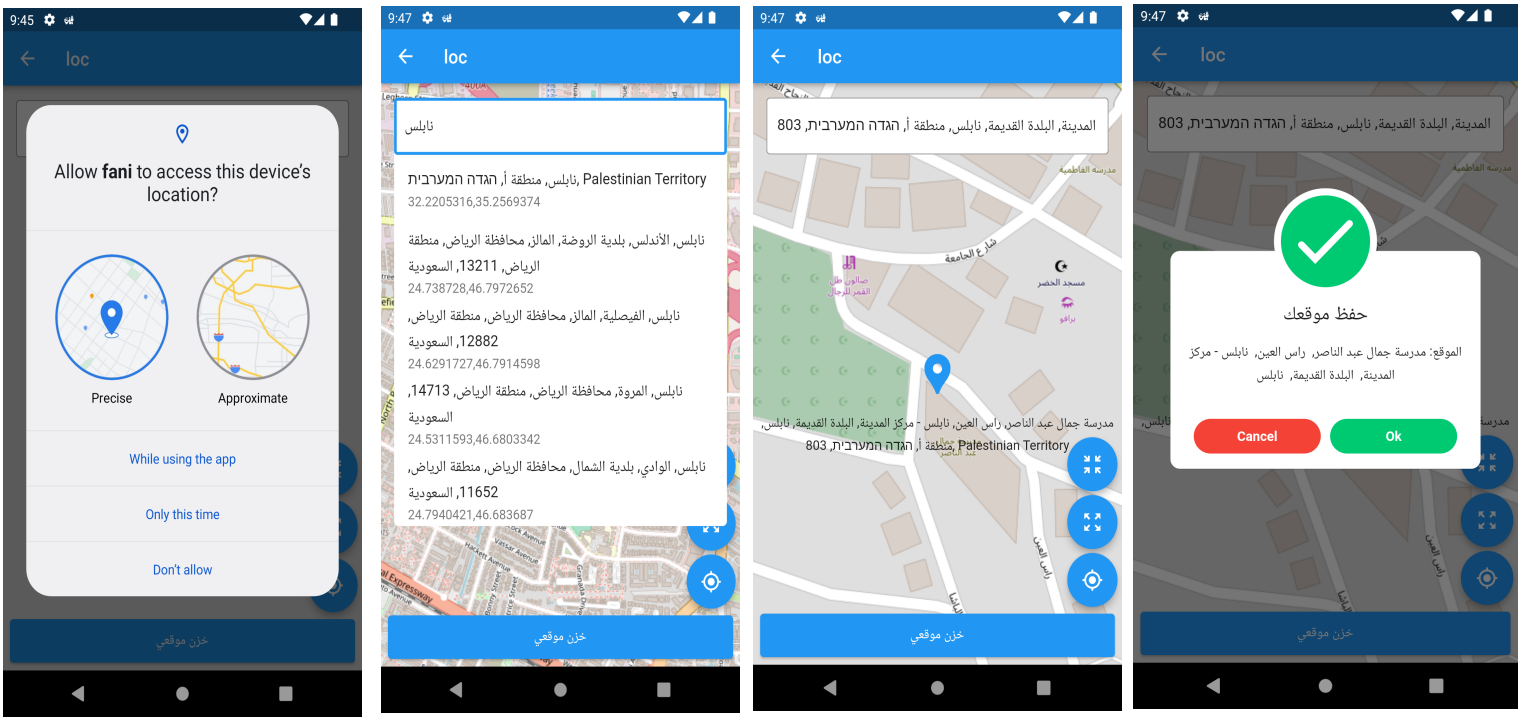


Figure 25.1.2: worker map

edit user profile image:

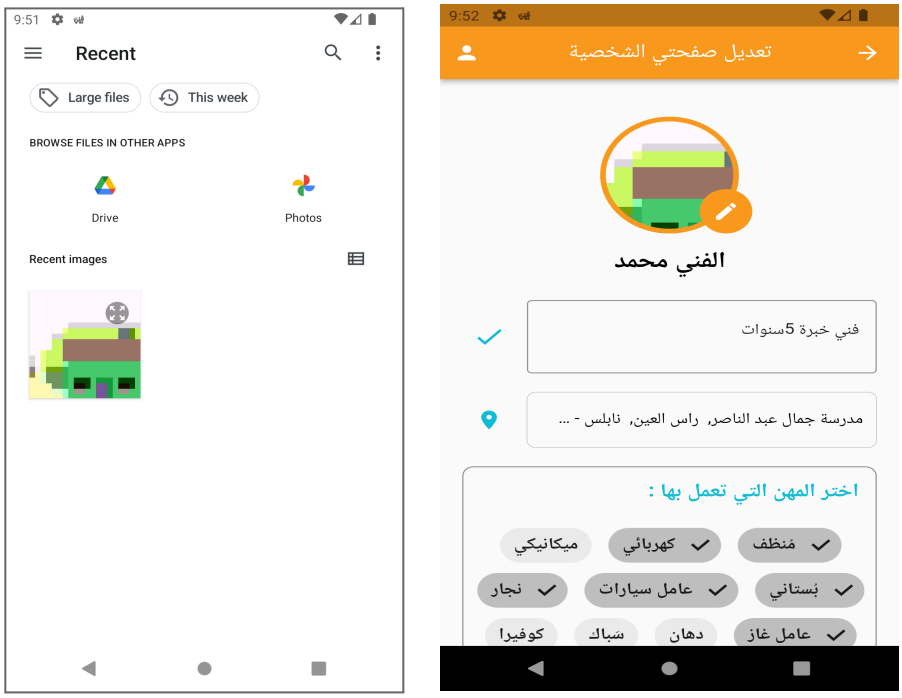


Figure 26: worker profile image

edit worker bio:

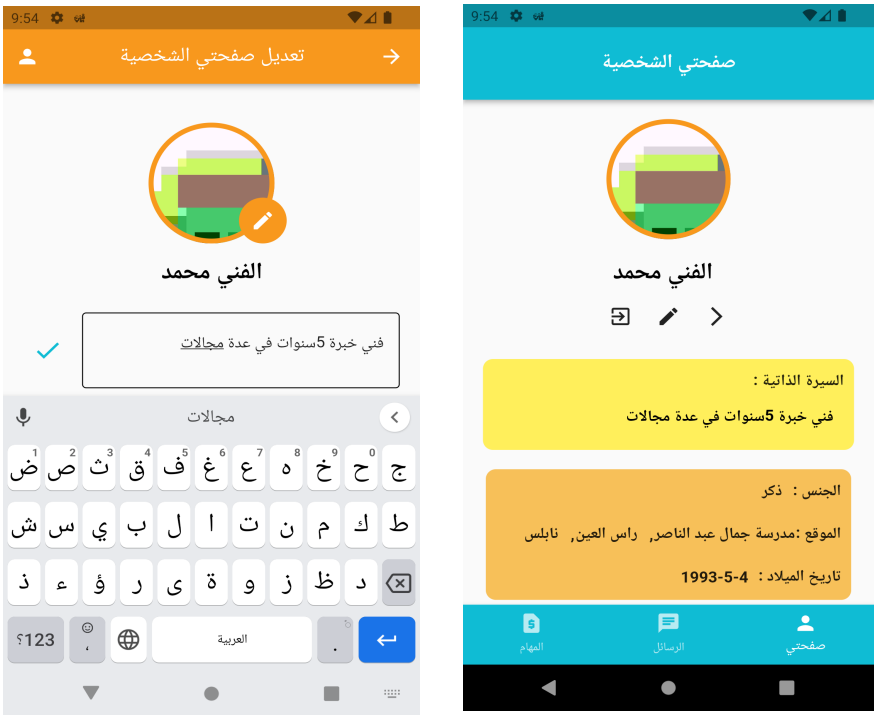


Figure 27: worker edit bio

edit phone and gender easily:

Figure 28: edit gender

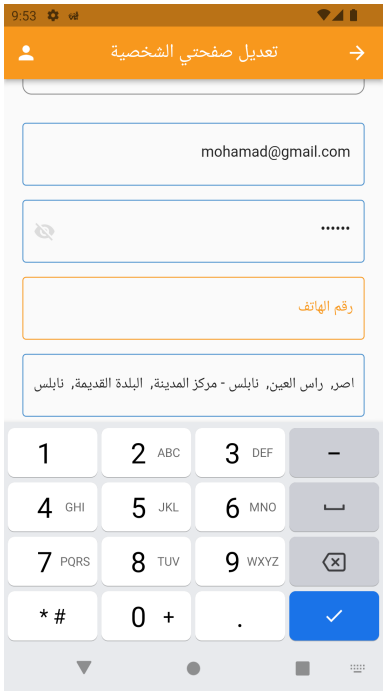


Figure 28: edit phone

the changes will appear on his profile

Figure 29: worker update profile

The message service between workers and clients and also with admins as we see above in figure 24.2.
* worker can see the profile of any users on a system

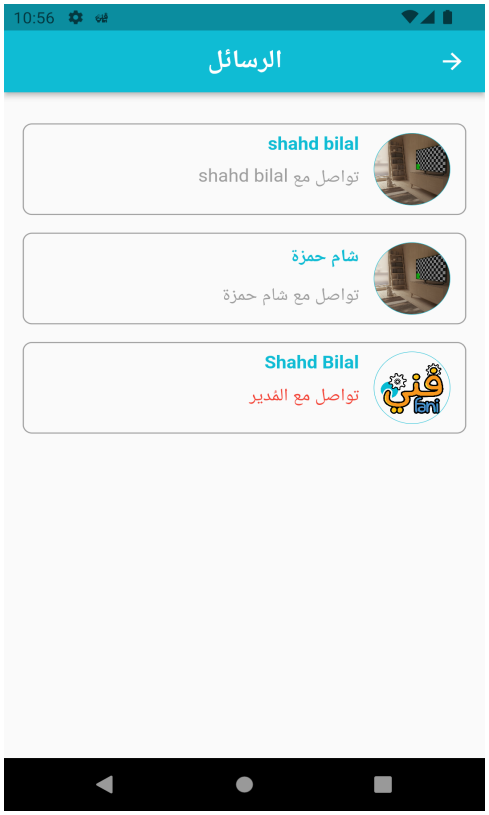


Figure 30: worker chat member

Figure 30.1: worker admin chat

Figure 30.2: worker see user

the order page for technician show the orders which sent to him with full information, and he can accept or deny

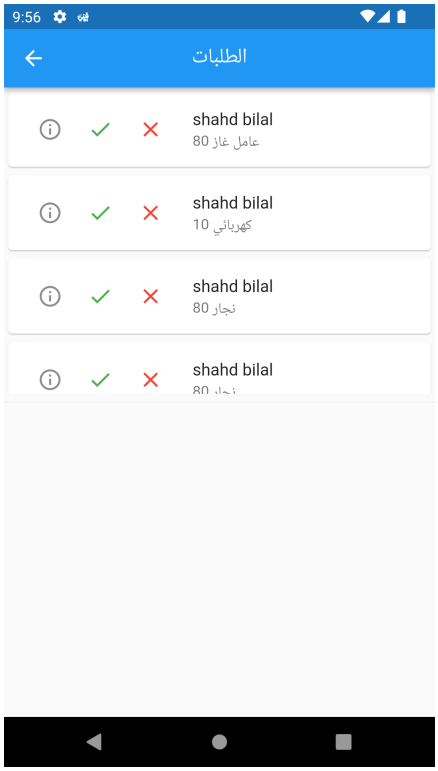


Figure 31 : worker see order

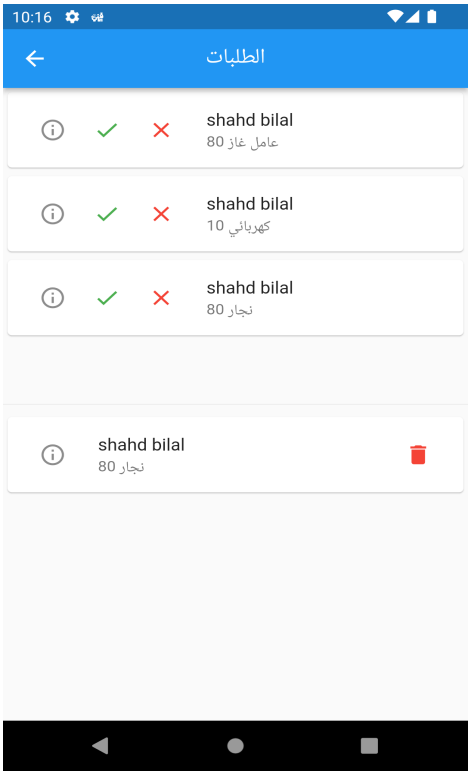


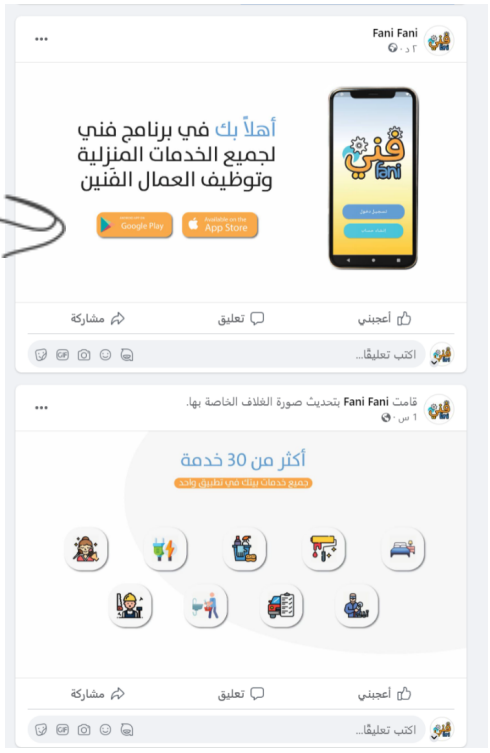
Figure 31.1 : worker agree order

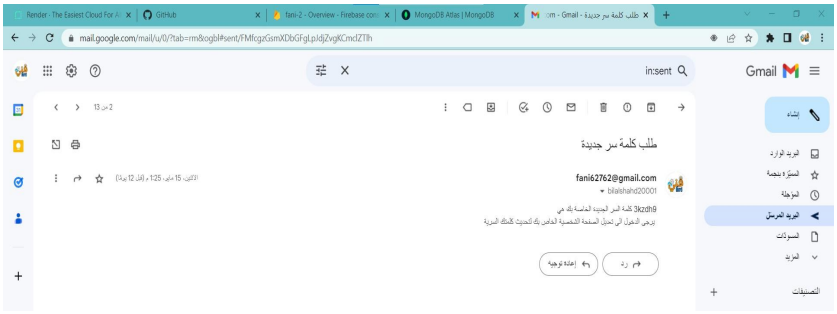
Figure 31.2 : order information

4.2.2. web pages

Figure 32: web main page

Following the development of our fani application, we recognized the need for a manager role responsible for supervising services, monitoring requests, conducting analysis, and managing worker assignments. To address this, we decided to create dedicated web pages for both the manager and the worker. This enables the manager to efficiently oversee services, and analyze data, while also providing the worker access to valuable insights regarding customer satisfaction and performance. By utilizing web pages, we can effectively meet these specific requirements and enhance the overall functionality of our application.





4.2.2.1. Admin Side

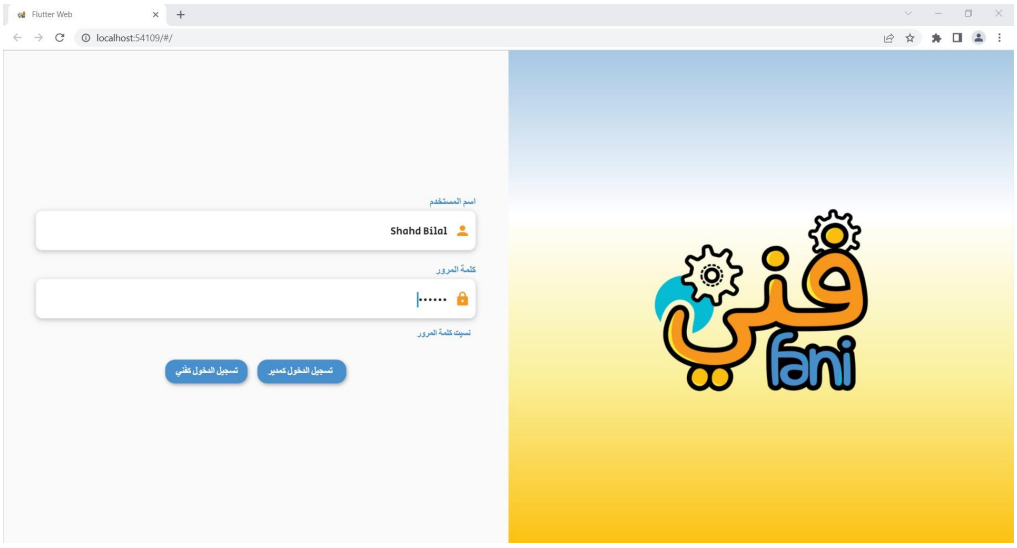
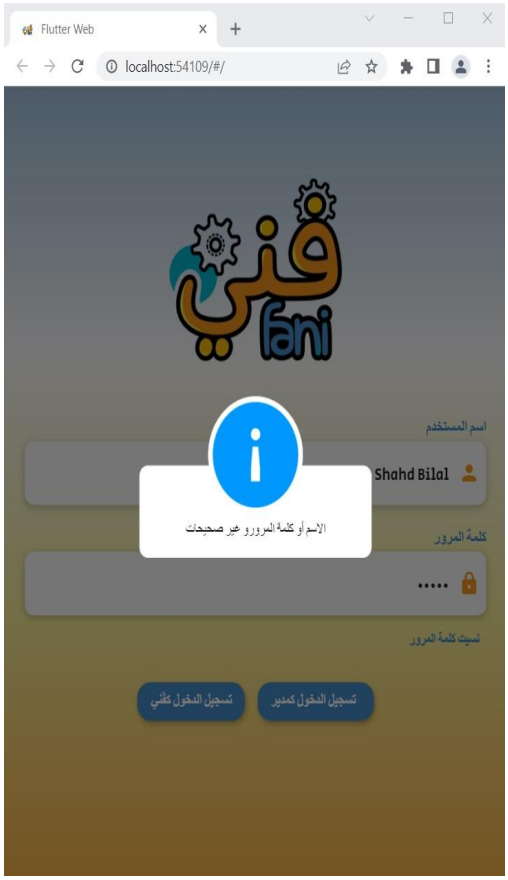


Figure 32: web login page

Our web login page accommodates both administrators and workers, allowing them to log in securely. Furthermore, we have designed the page with consideration for various screen sizes to ensure a responsive and user-friendly experience.



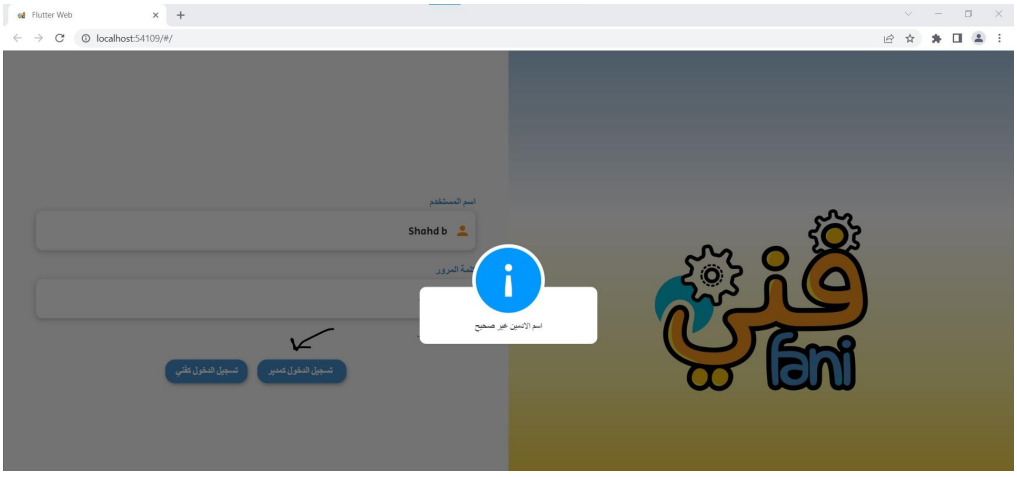
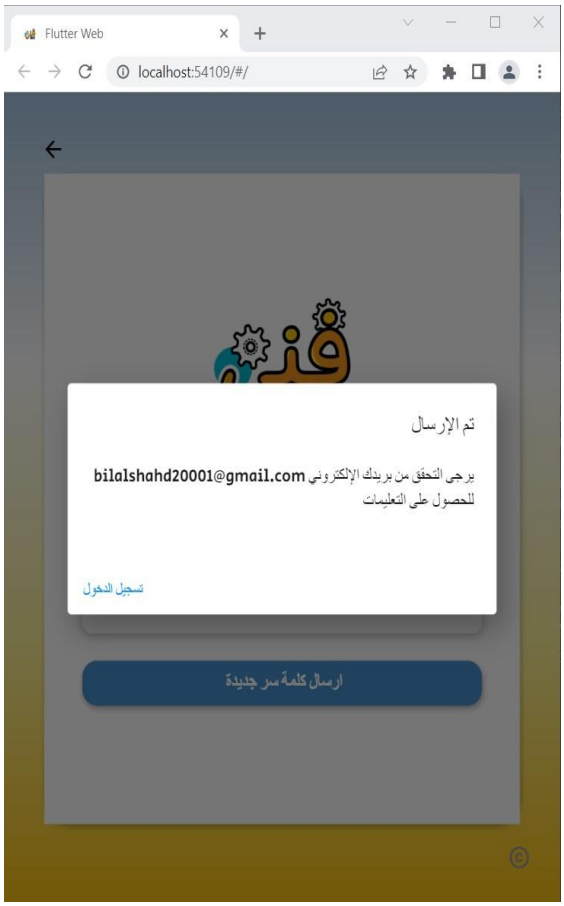


Figure 32.1: web handle login page

In case of entering an incorrect name or password as an admin, a descriptive message will be displayed to provide guidance and inform the user about the error. This feature ensures that users receive clear feedback when attempting to log in with incorrect credentials, allowing them to rectify the issue and successfully access the system.



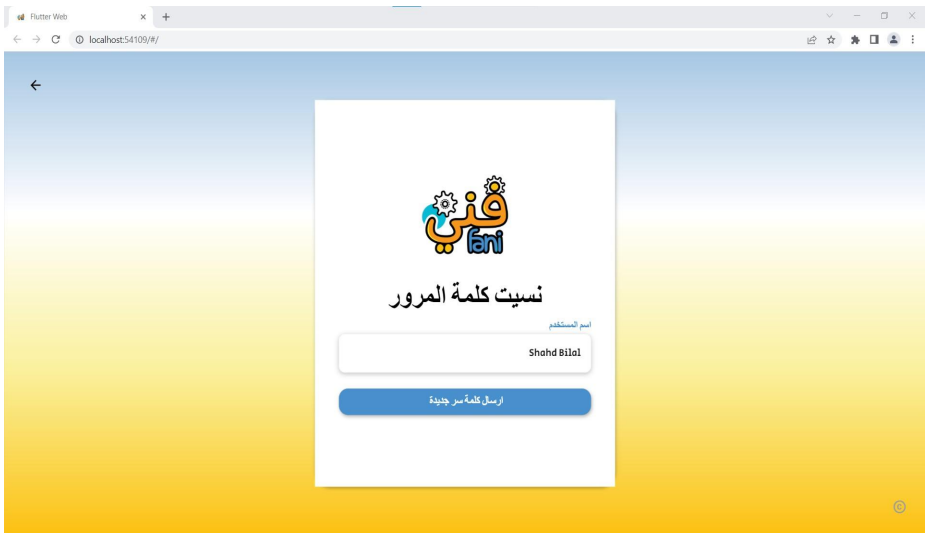


Figure 33: web forget password

In the event that the admin forgets their password, they can retrieve a new password by providing their name. A new password will be sent to the admin's registered email address, enabling them to regain access to their account. This secure and convenient password recovery process ensures that admins can easily recover their passwords in case of forgetfulness while maintaining the security of their accounts.

```
_id: ObjectId('645506c0b0f11cd3acde96d7')
name: "Shahd Bilal"
email: "bilalshahd20001@gmail.com"
password: "udjh4q"
gender: "أنثى"
phone: "059666777693"
image: "https://firebasestorage.googleapis.
date: "2001-5-16"
```

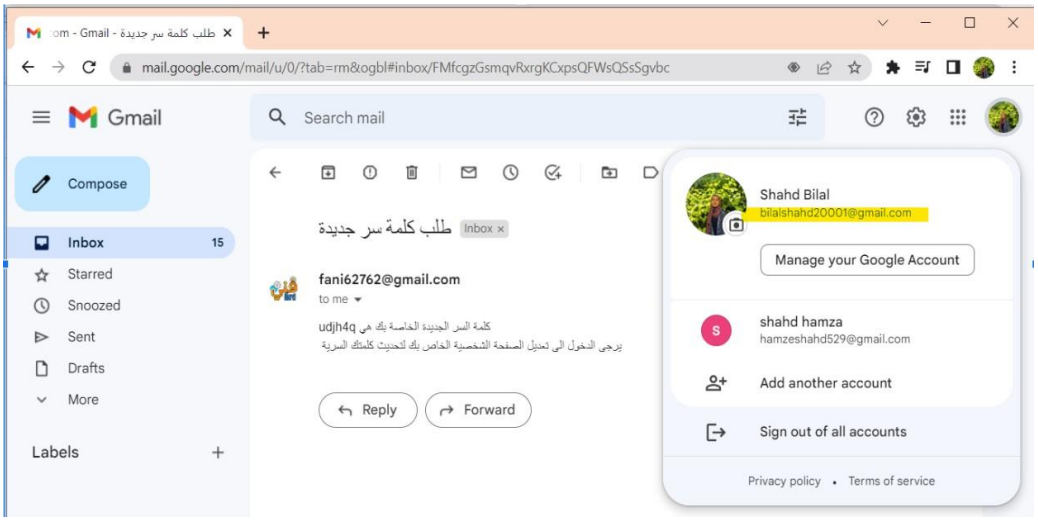


Figure 34: password backend

Dashboard for the admin

customer page:

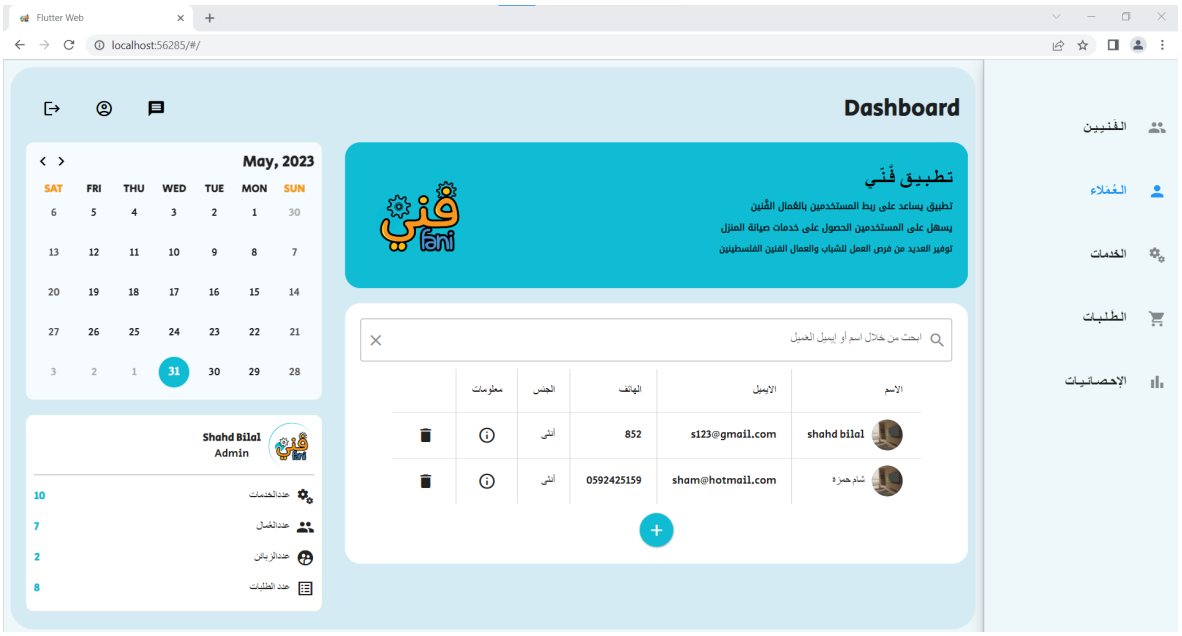
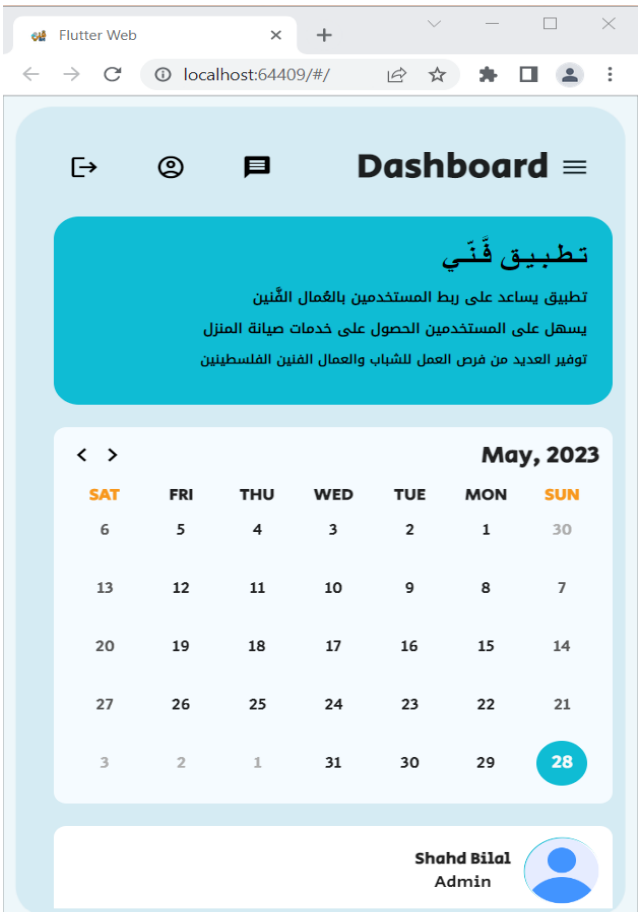
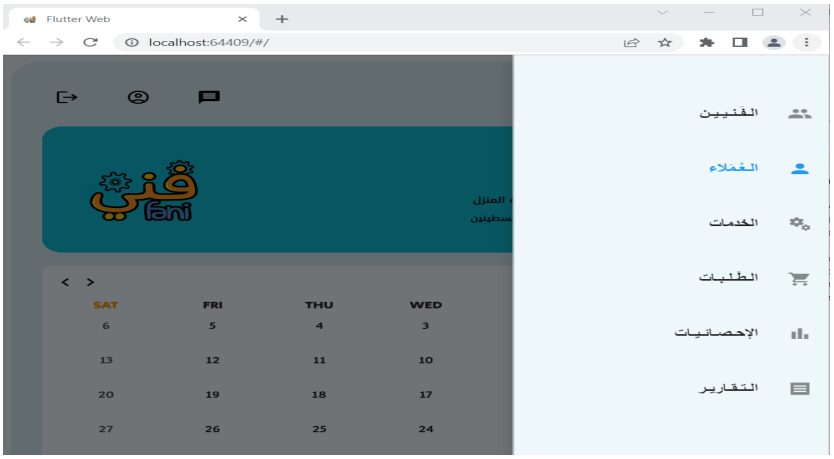


Figure 34: customer web page





Welcome to the main page for administrators. Here, you will find a concise project analysis that includes key metrics such as the number of workers, users, requests, and services, as well as a calendar to facilitate tracking important dates and requests. Additionally, there is a user-friendly list view that allows easy navigation, empowering you to effortlessly select and access the desired information, as depicted in the accompanying picture.

Admin can add or delete users

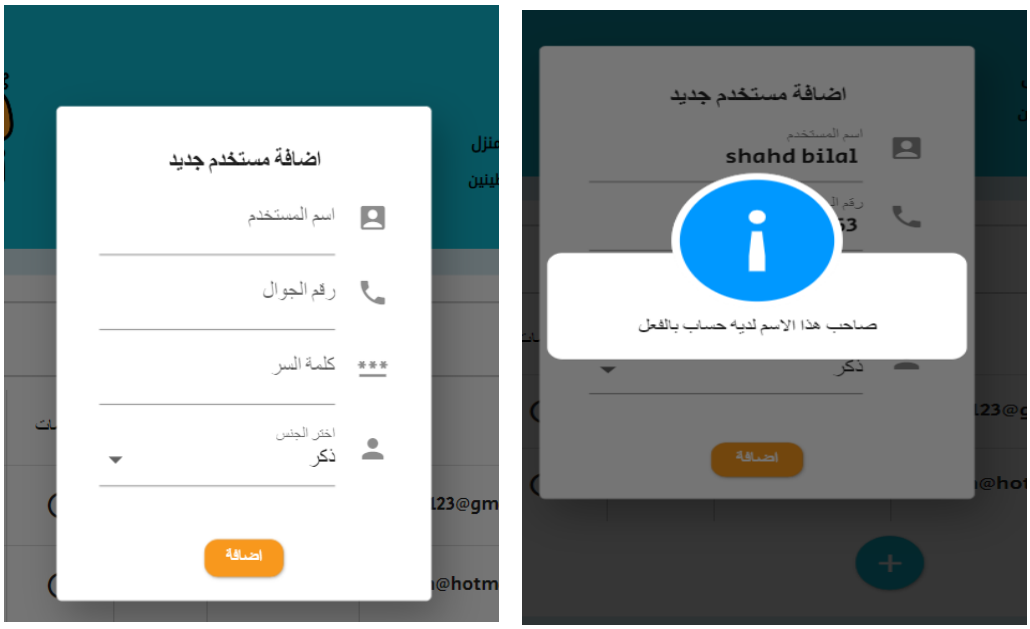


Figure 35: add new customer

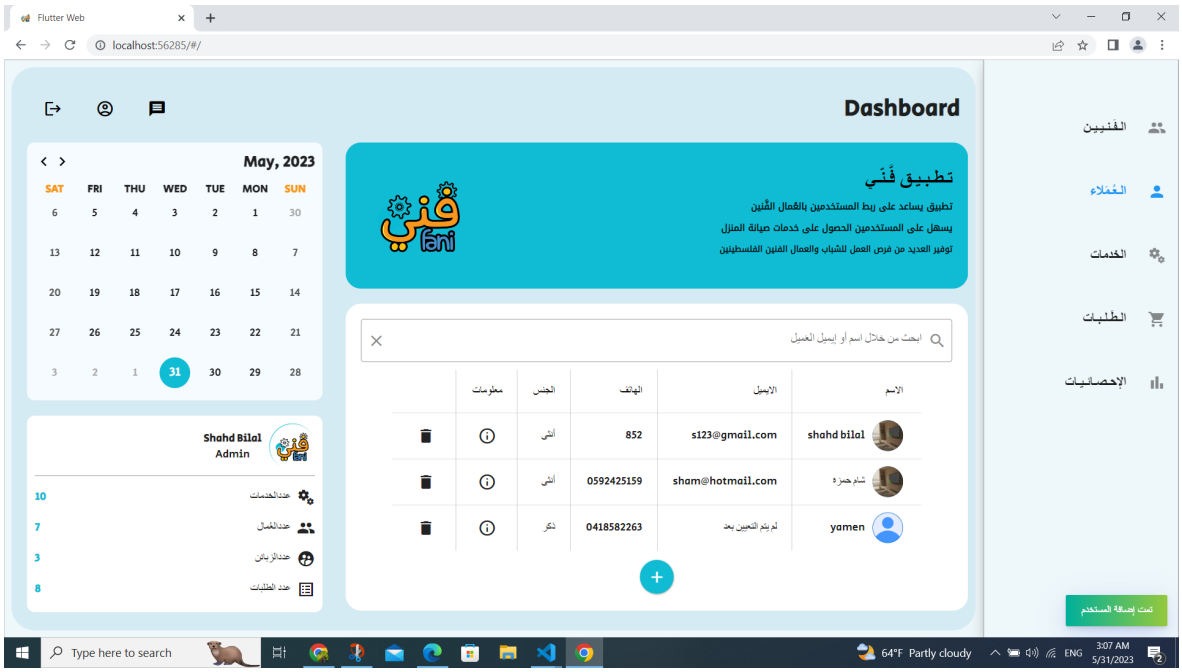
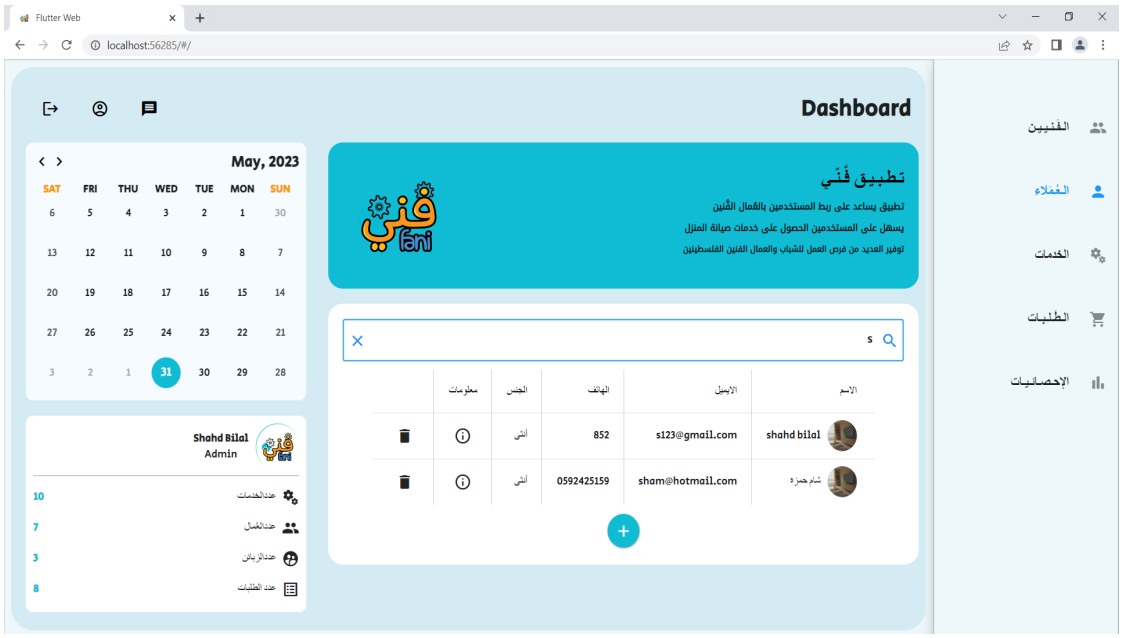


Figure 34.1: customer added



we can search by name or email for the user

Figure 35: search web page

admin also can see the profile for the specific user and the orders he orders:

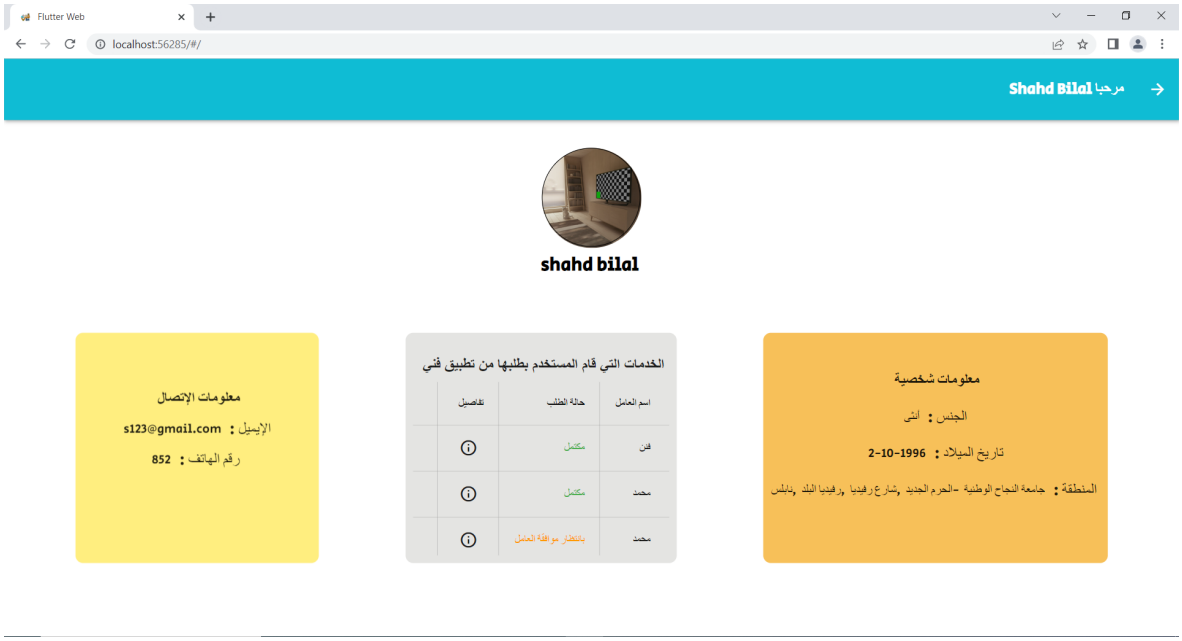


Figure 36: admin see user

it works with any size page with scroll vertically to see all information

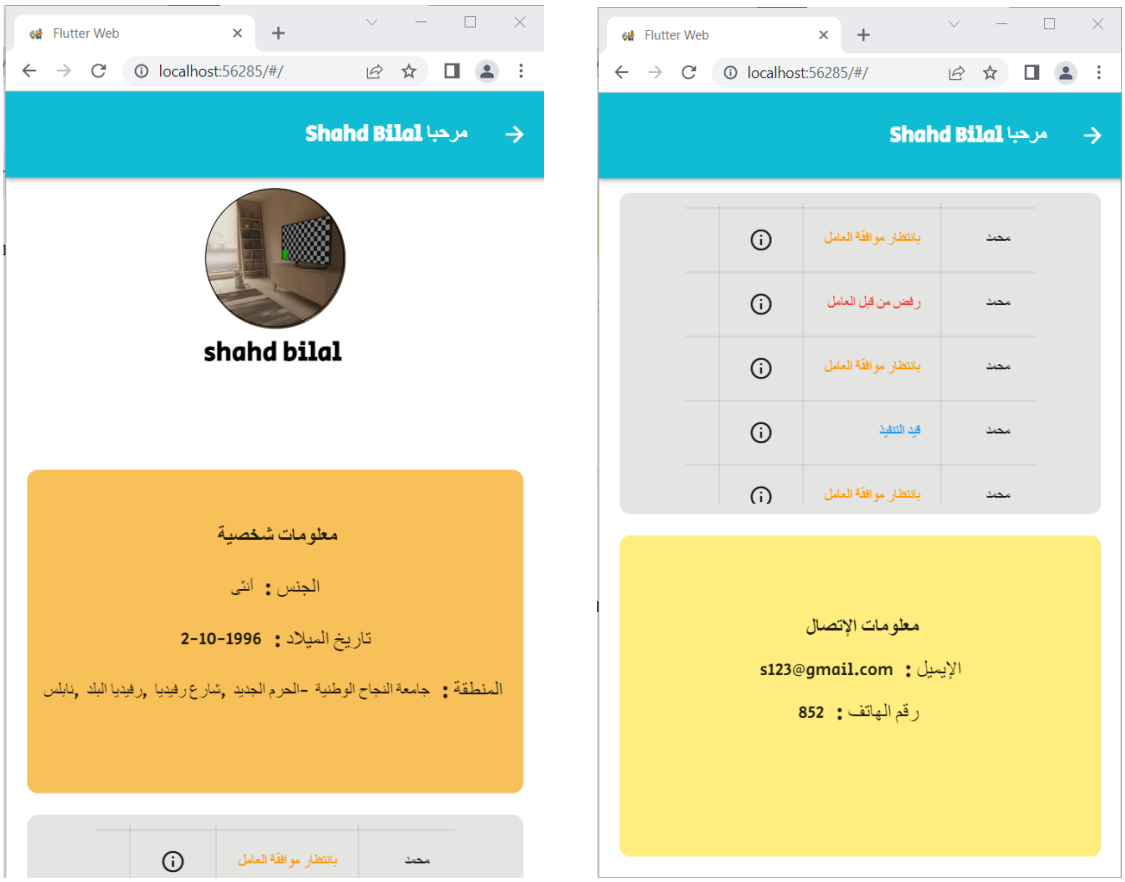


Figure 36.1: admin see user platform

The admin also possesses the ability to send and receive messages from any user within the system. This feature allows seamless communication between the admin and users, facilitating efficient collaboration and enabling the timely exchange of information and updates.

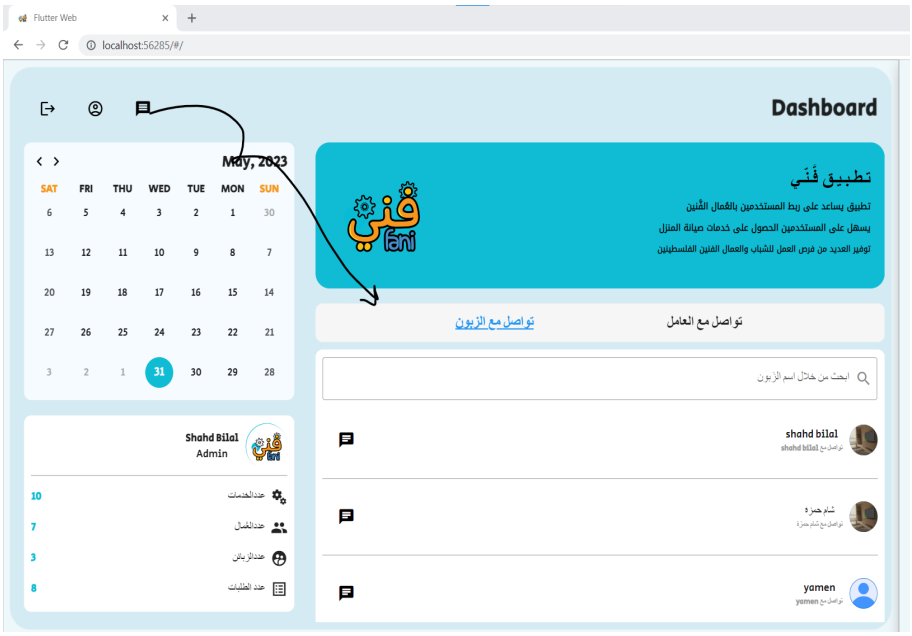


Figure 37: admin chat member

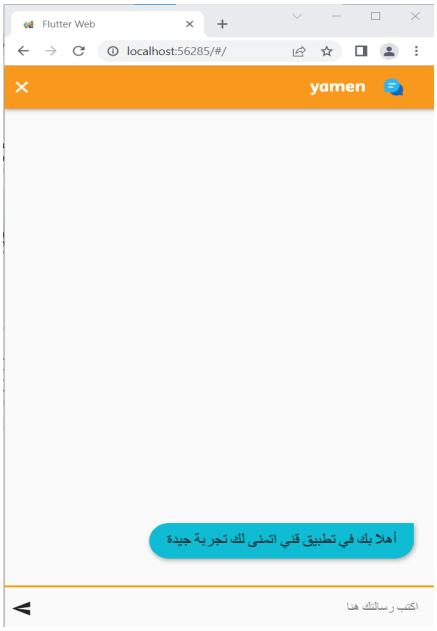


Figure 37.1: web chat page

workers page:

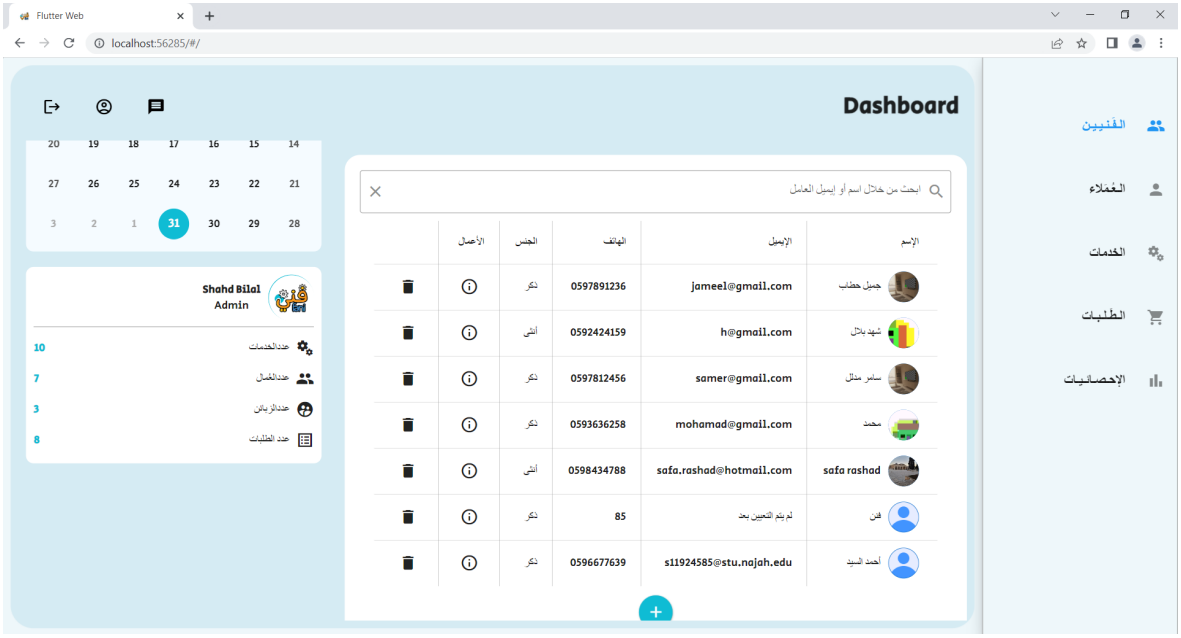


Figure 38: worker web page

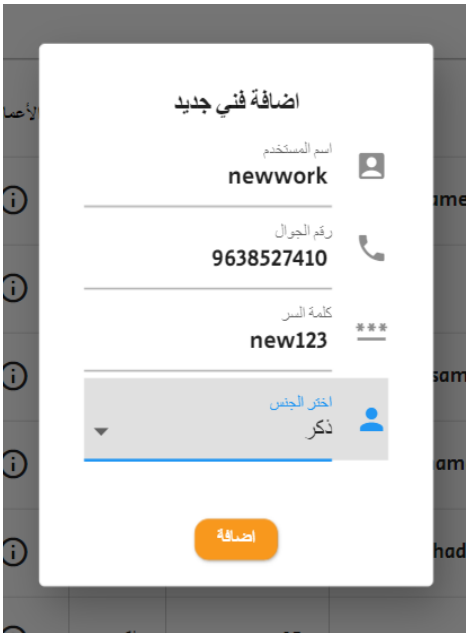


Figure 36: add new worker

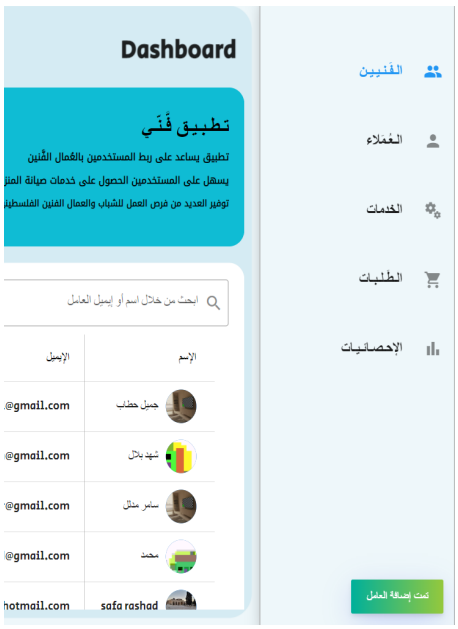


Figure 36.1: worker added

admin also can see the profile for the specific worker and his works:

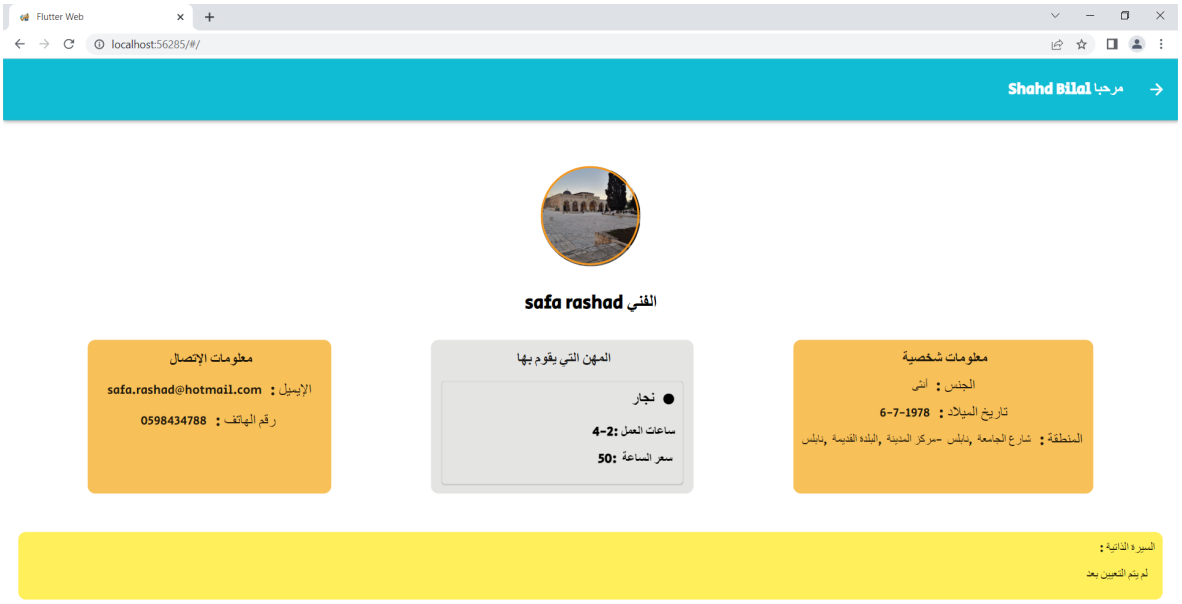
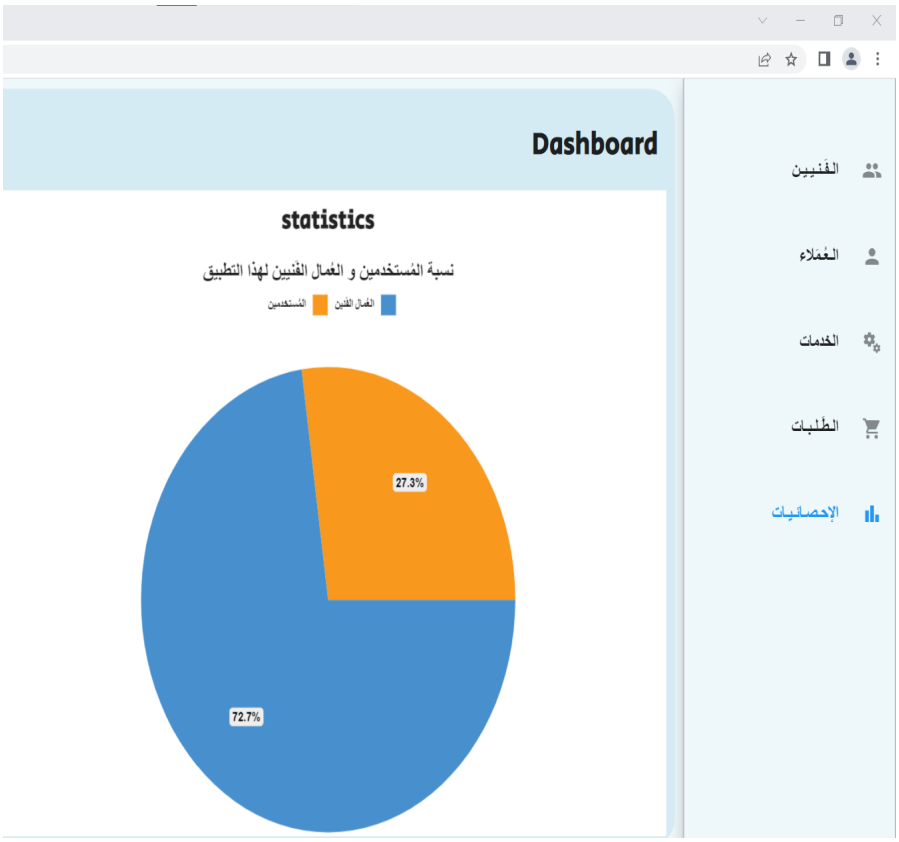


Figure 36.1: admin see worker

also, the admin can communicate with any workers.

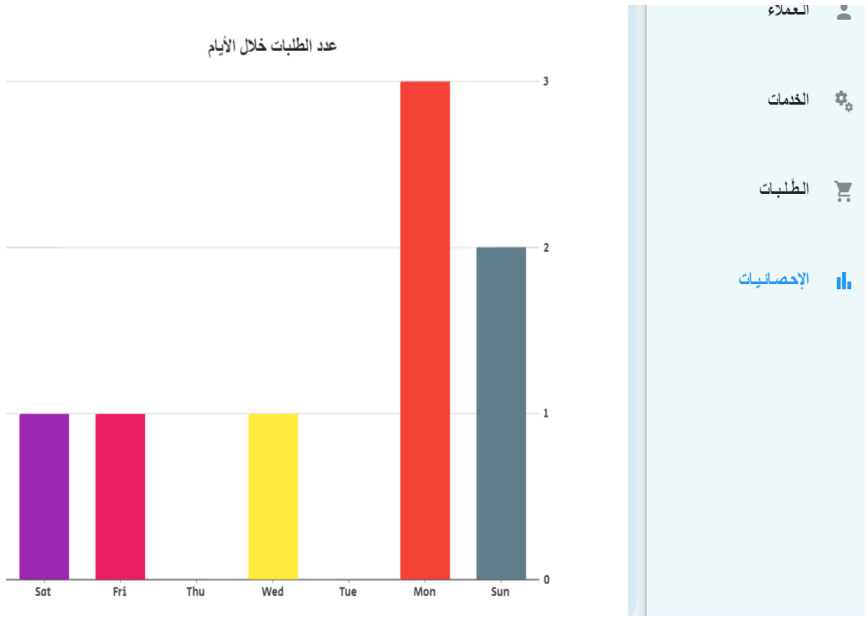
Figure 37: admin worker chat page

Admin can see general statics for works:



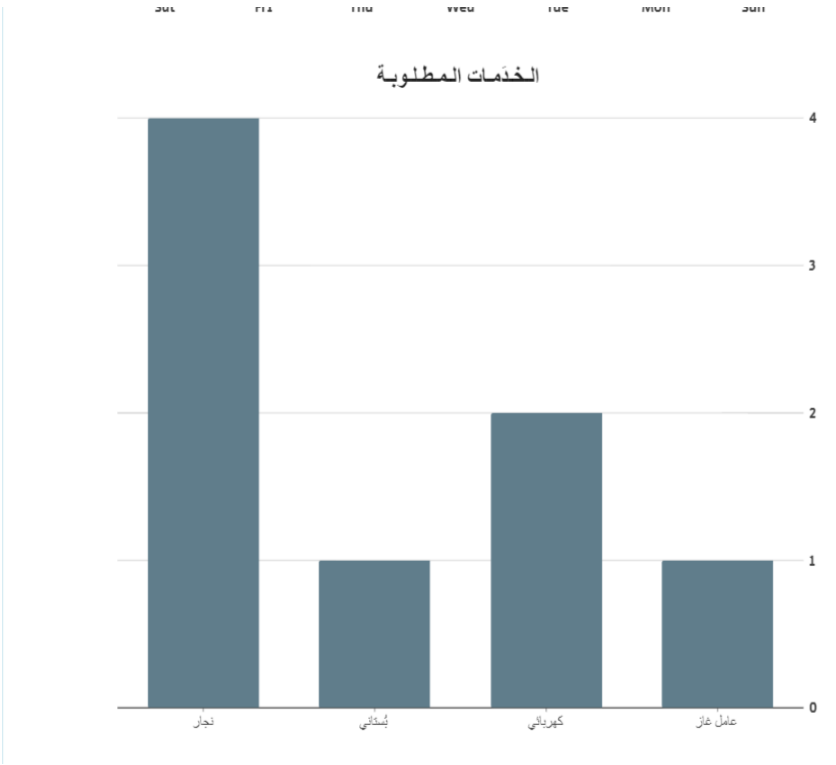
like here percentage normal users and workers on our app

Figure 38: percentage users and workers



and see which day have more orders.

Figure 39: orders and days



what is the best service I have?

This graph shows the most services ordered

Figure 40: orders and services

and finally the percentage of male and female users and workers on our app:

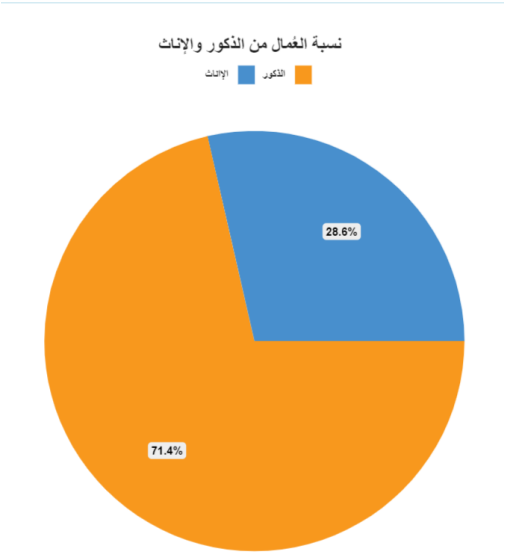


Figure 39: workers gender

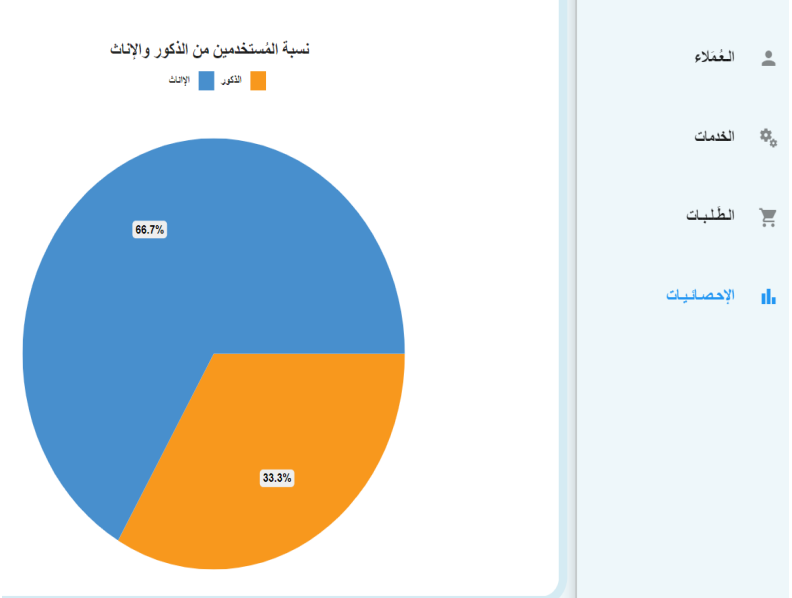


Figure 40: users gender

admin can see his profile page and edit it:

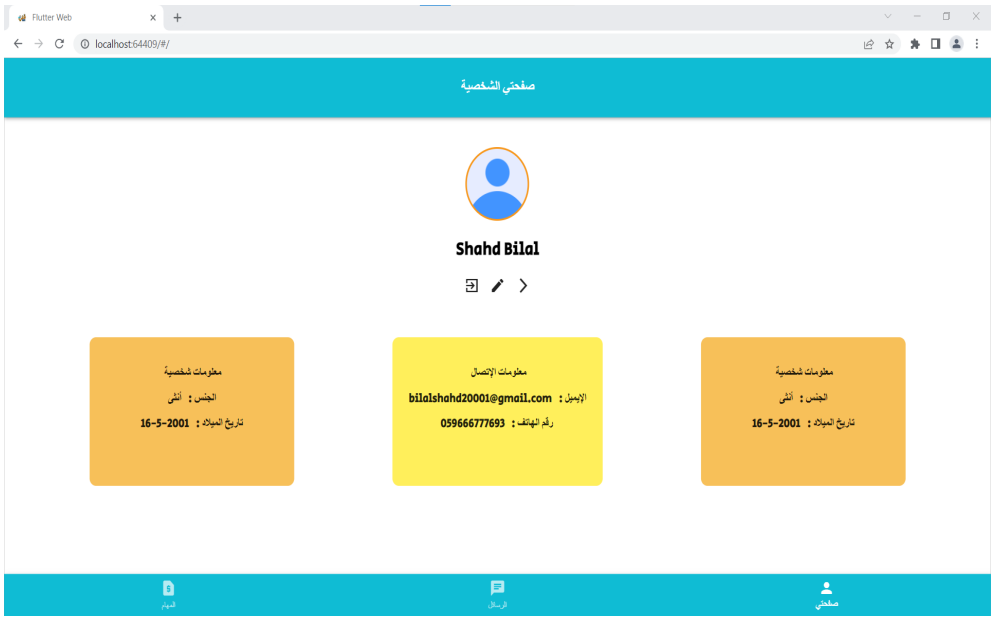
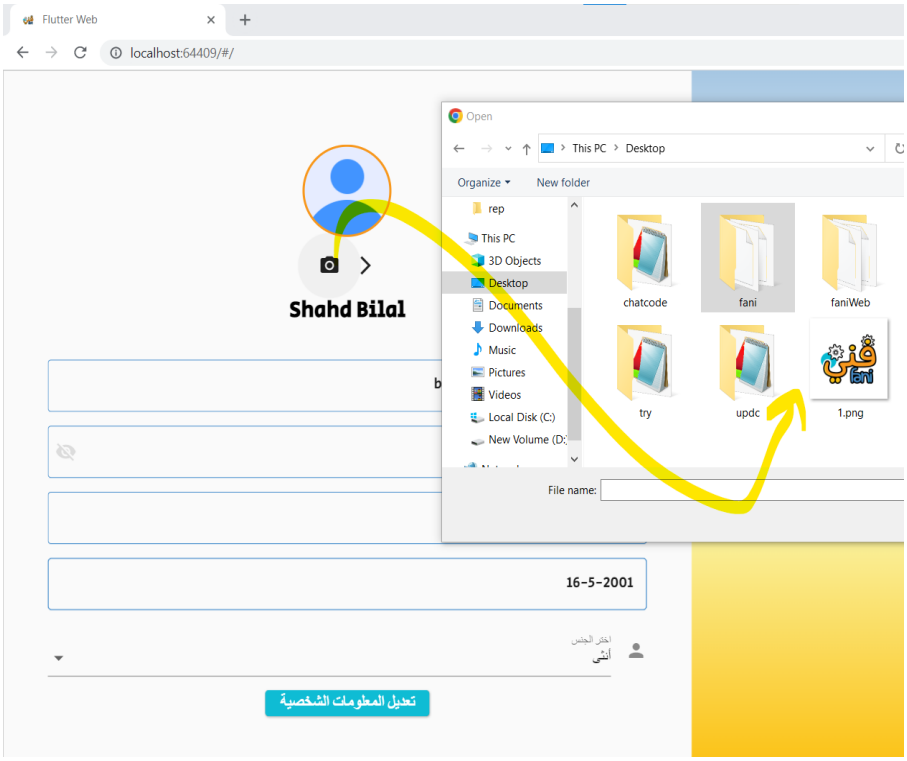


Figure 41: Admin profile page

edit his image:



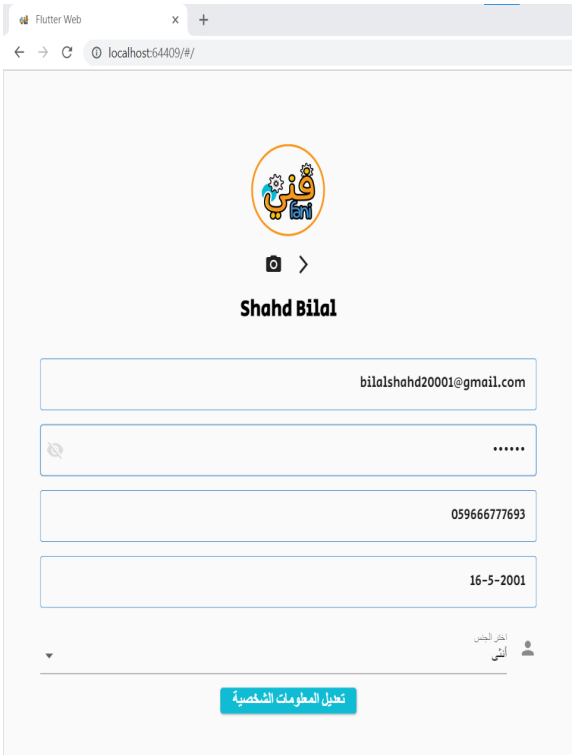


Figure 42: Admin edit image

edit date of birth:

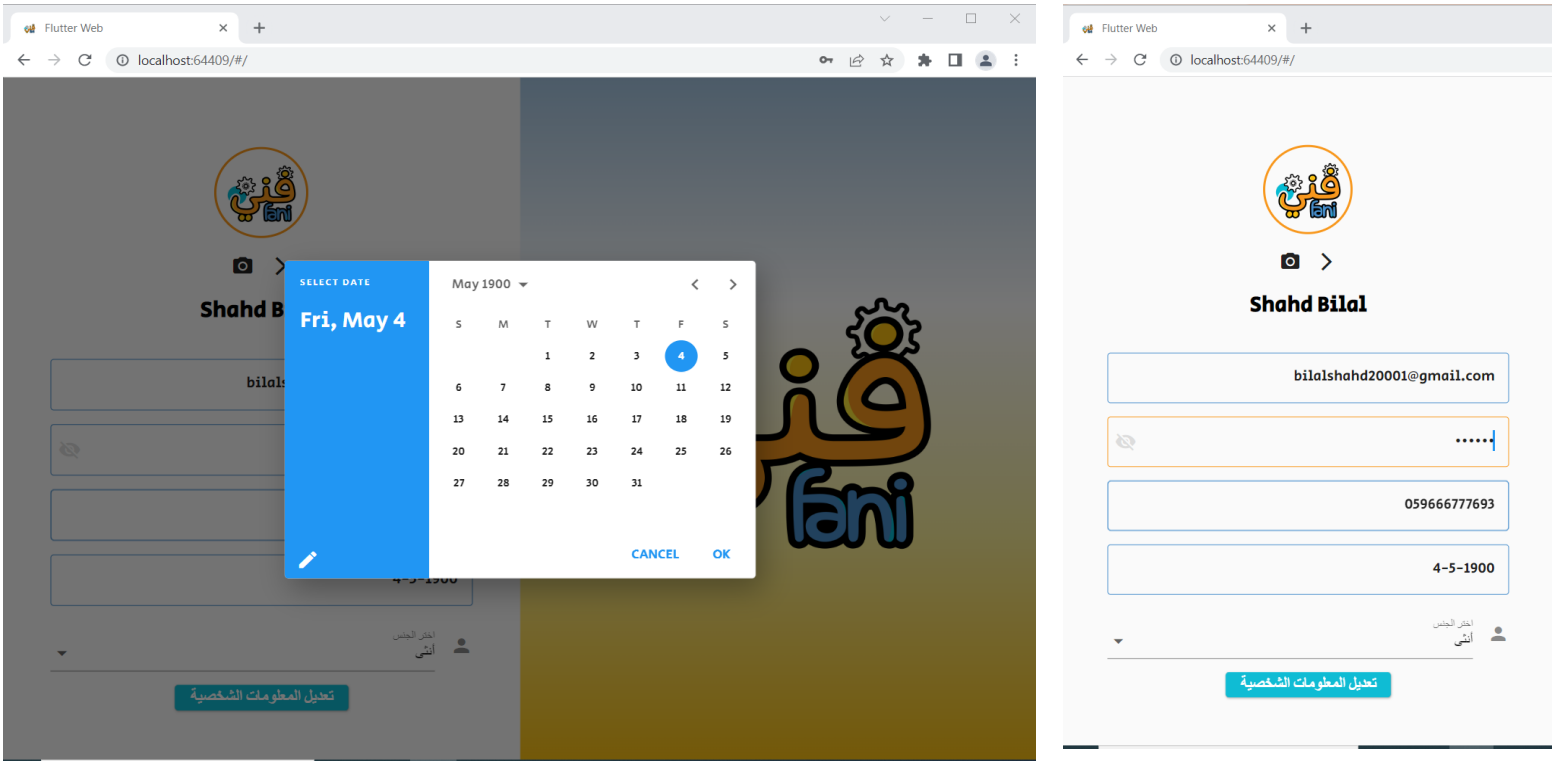
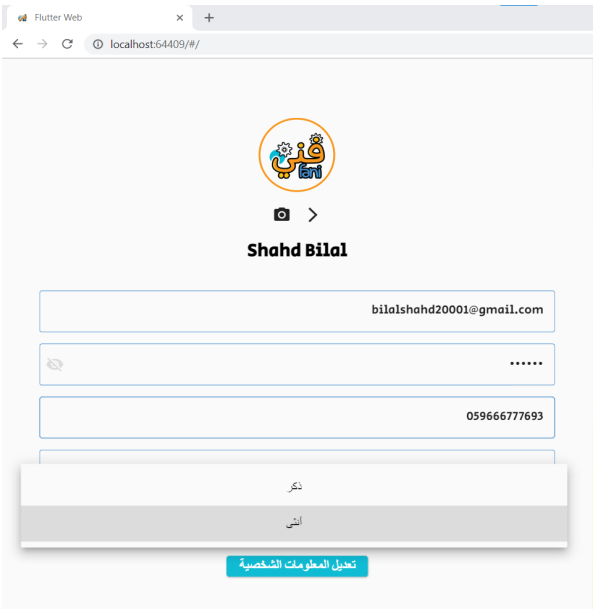


Figure 43: Admin edit date

for edit gender:



so easily we can edit any field.

Figure 44: Admin edit gender

orders page:
this page contains all orders.

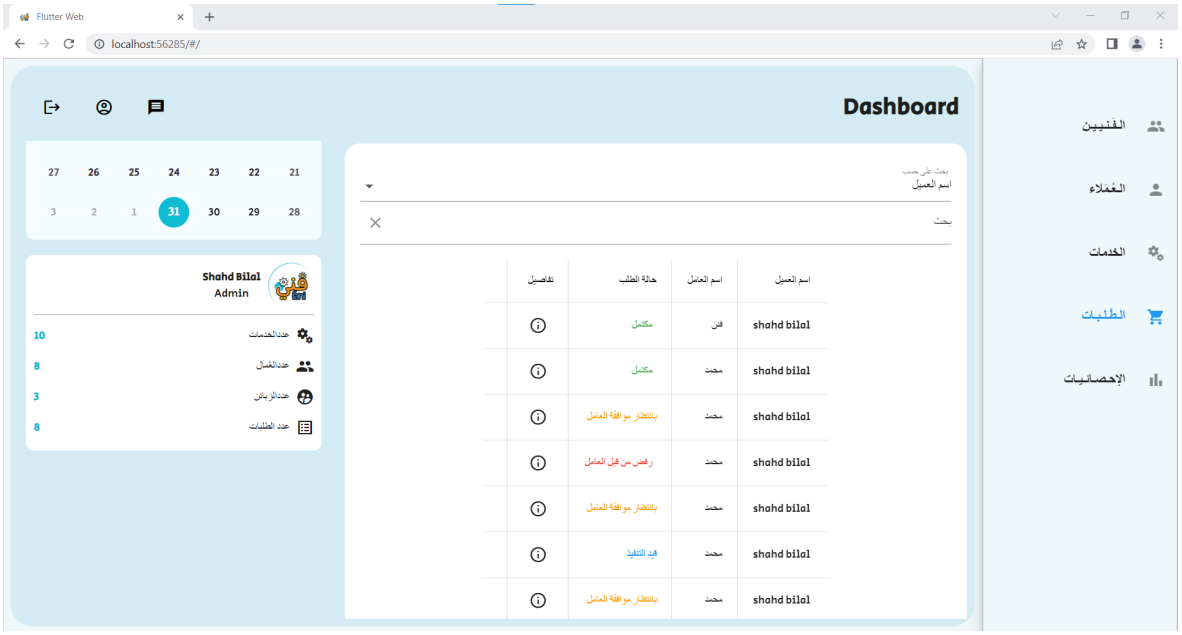


Figure 45: orders web page

Figure 46: order information web page

search for specific order using many fields.

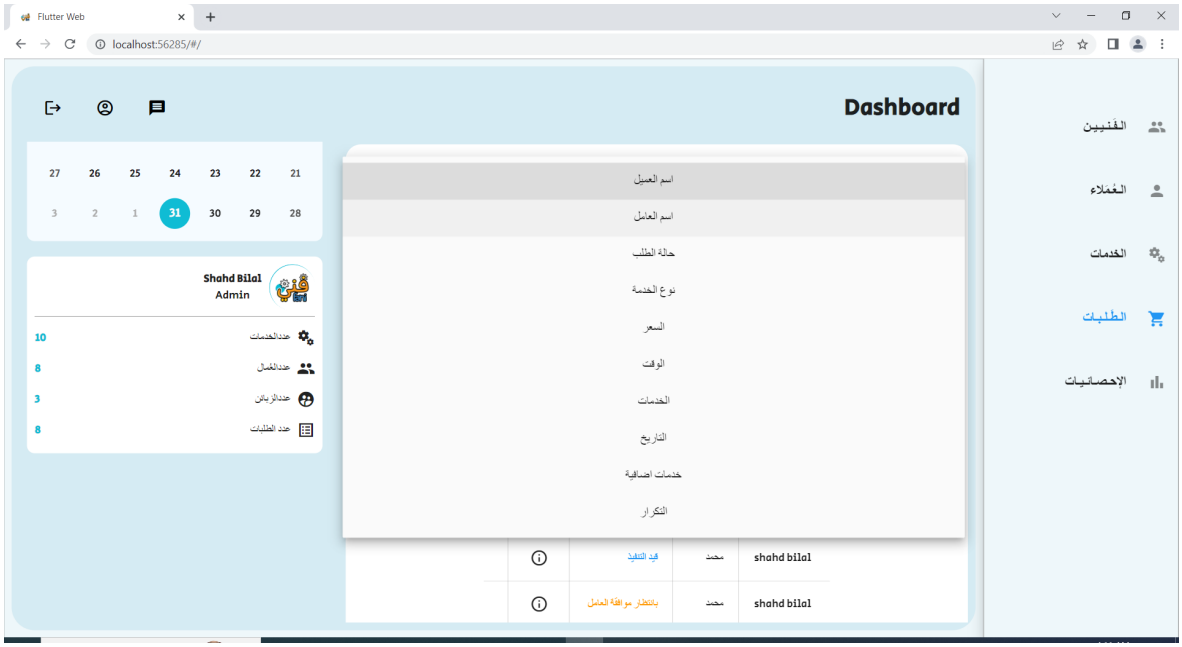
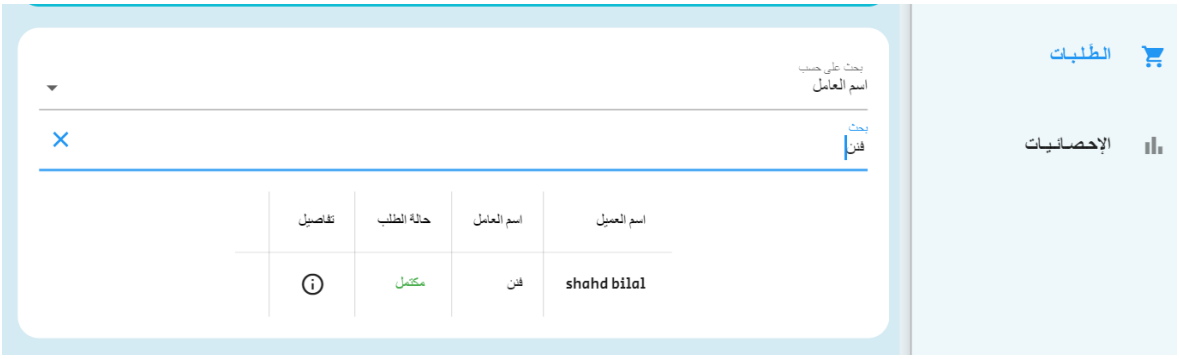
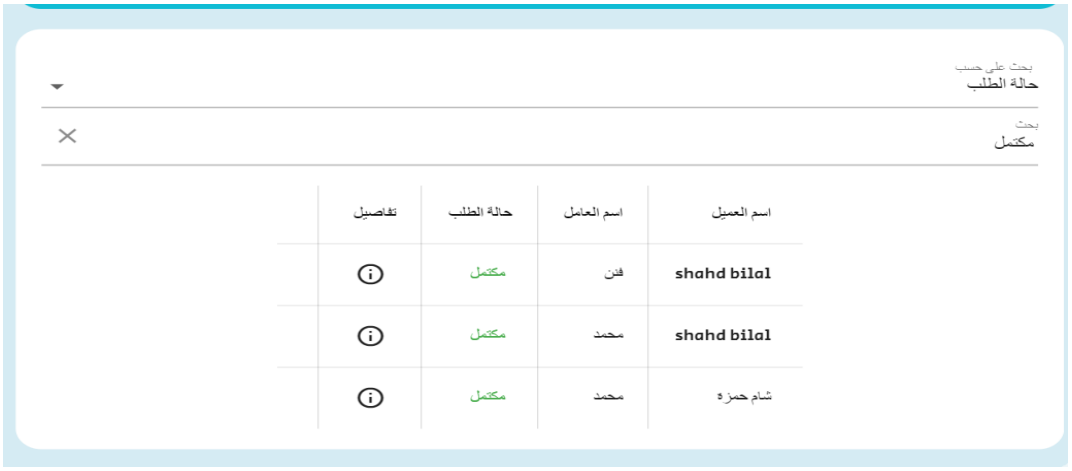


Figure 47: orders search web page





Also worker can delete a specific user or a specific worker
but when deleting a worker maybe the user already reverse this worker so
when the admin deletes the worker the email will arrive to the worker telling him he is removed and the notification arrives to the user to let him chose other worker
and the same when an admin deletes a user, the Gmail will arrive to the user and notification for the work.

like here when an admin deletes worker leen worker all the users ordered from him will receive a notification

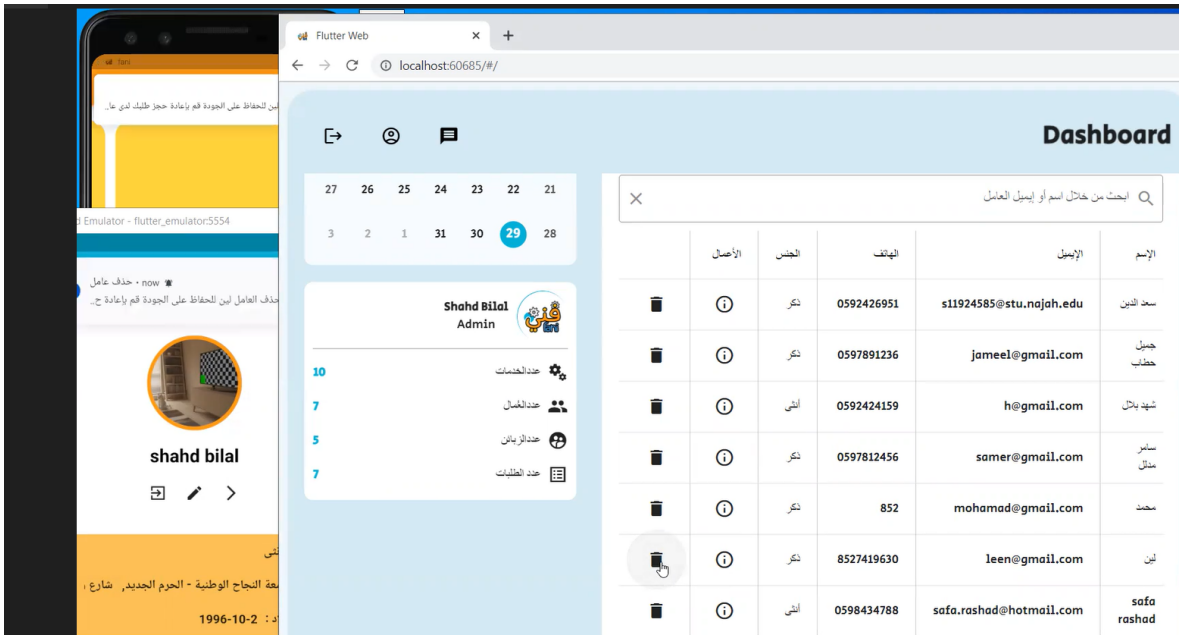
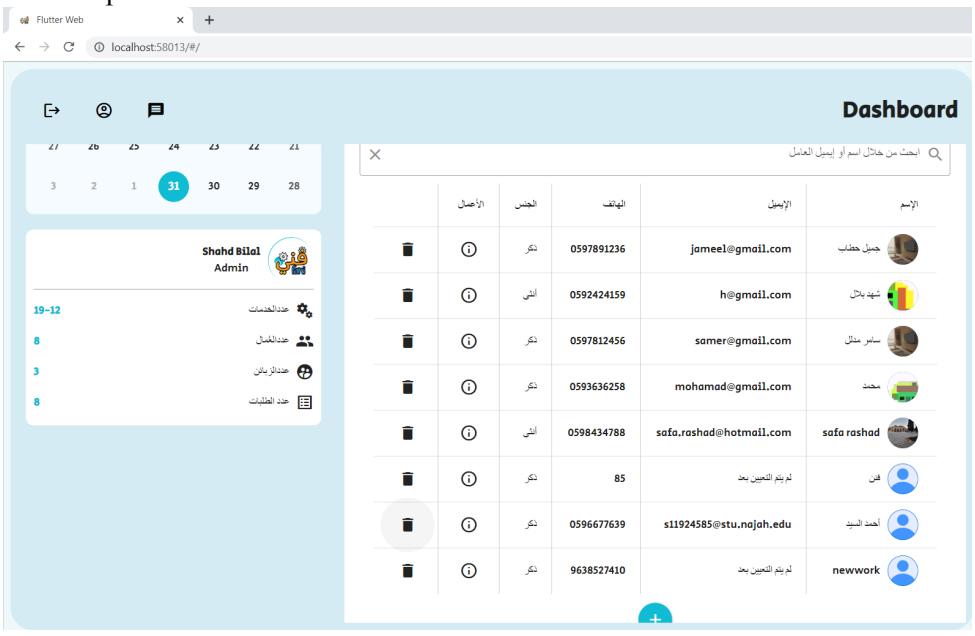


Figure 48: delete worker notification

for example delete Ahmad al said:



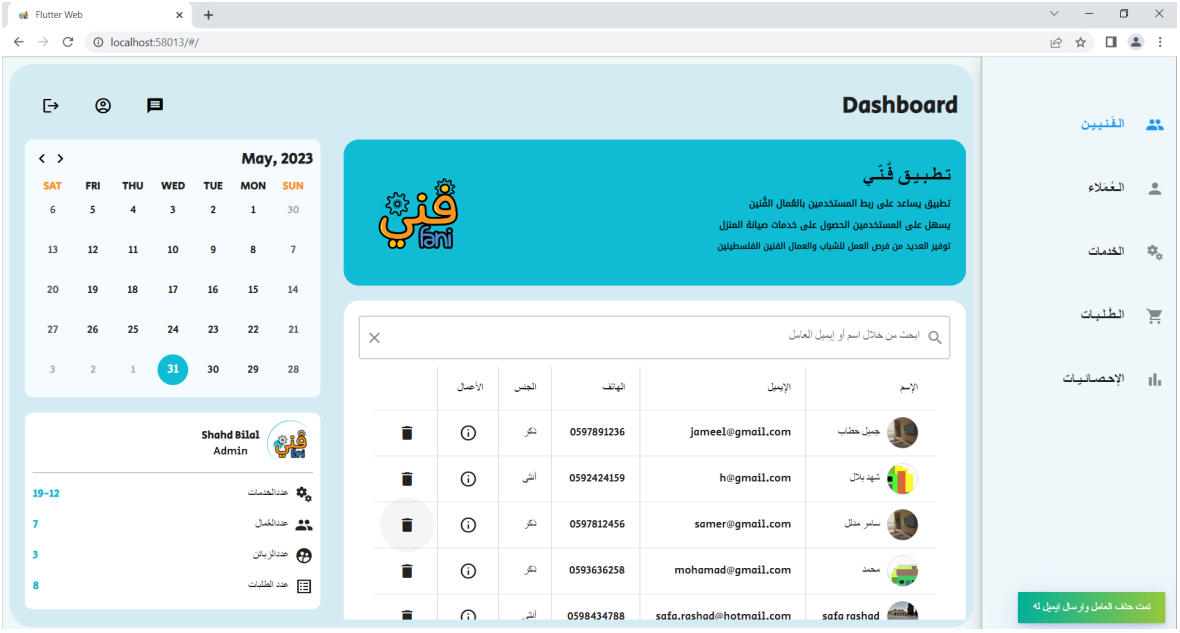


Figure 49: workers deleted success

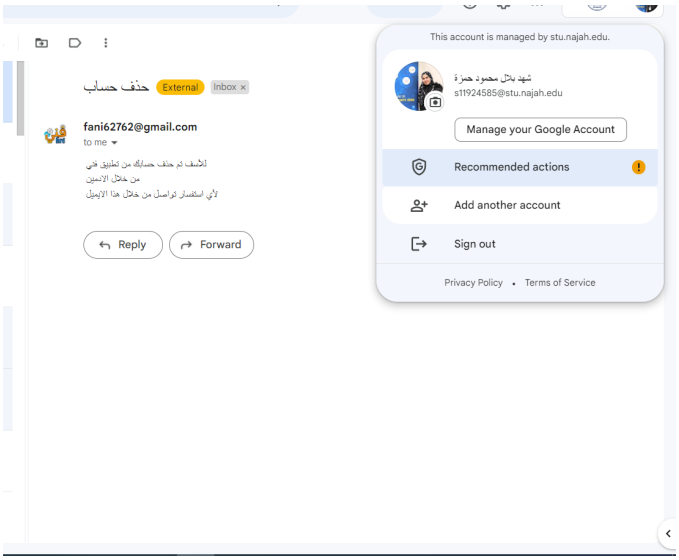


Figure 50: deleted worker Gmail.

and the same process for deleting users

server page:
this page contains all services.

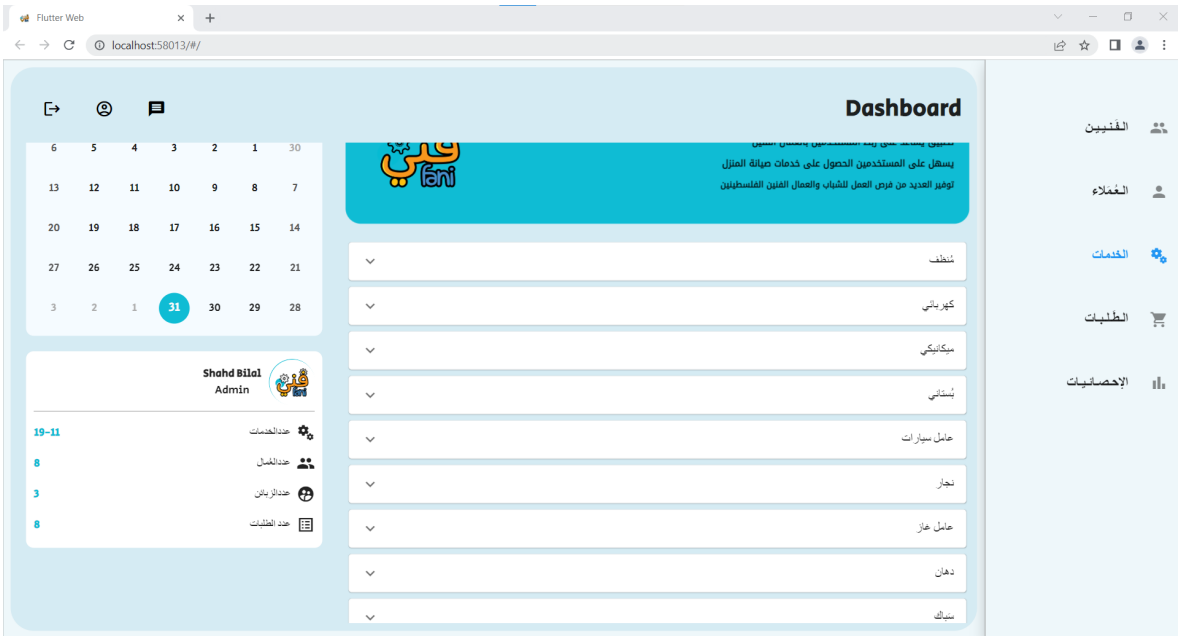


Figure 51: server web page

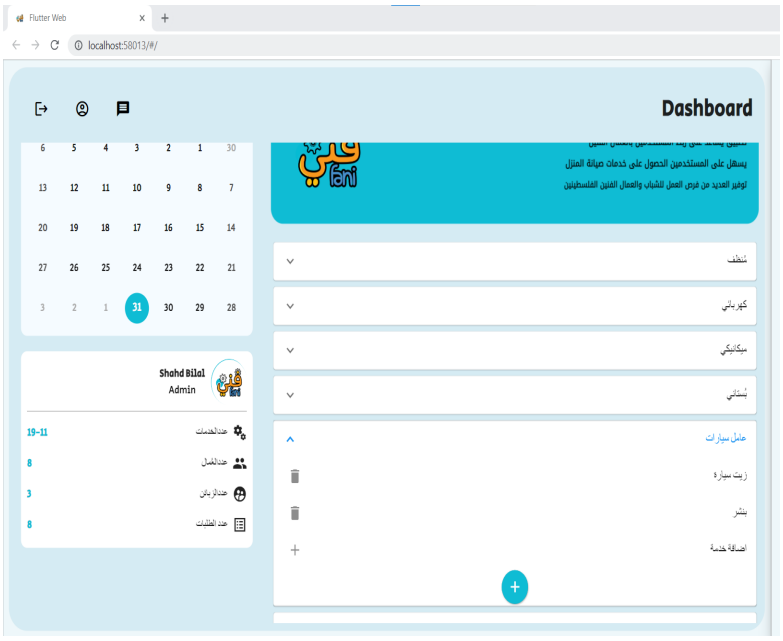


Figure 51.1: add server web page

4.2.2.2. Worker Side

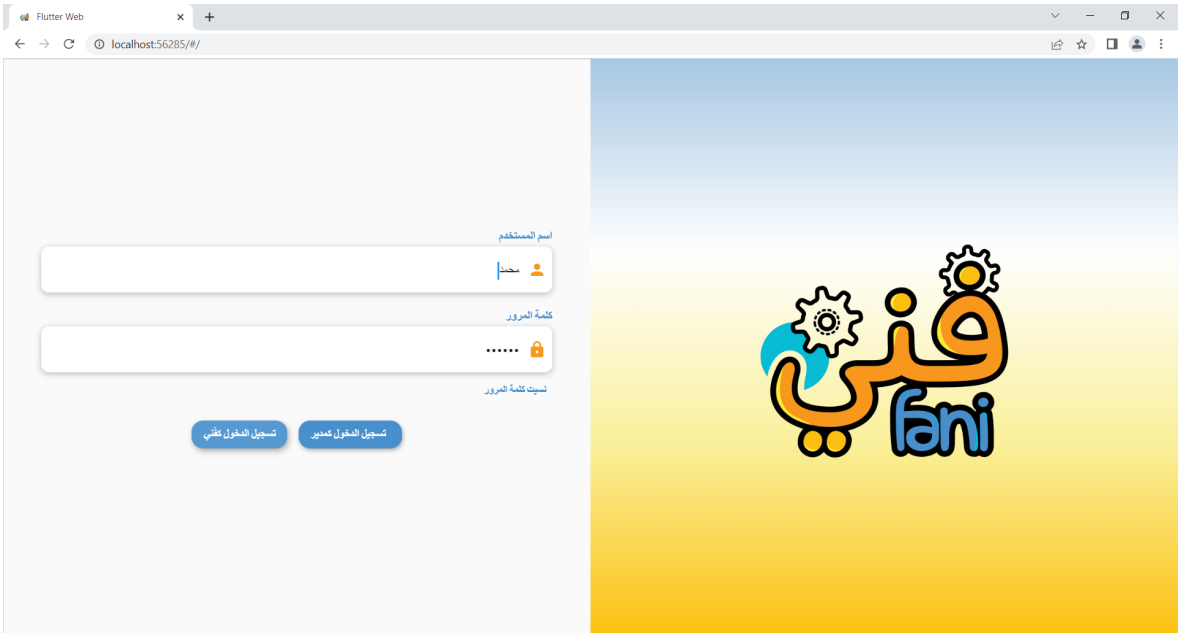


Figure 52: worker login page

dashboard for the worker

the same design as the admin dashboard but the data here relate to the just worker how login

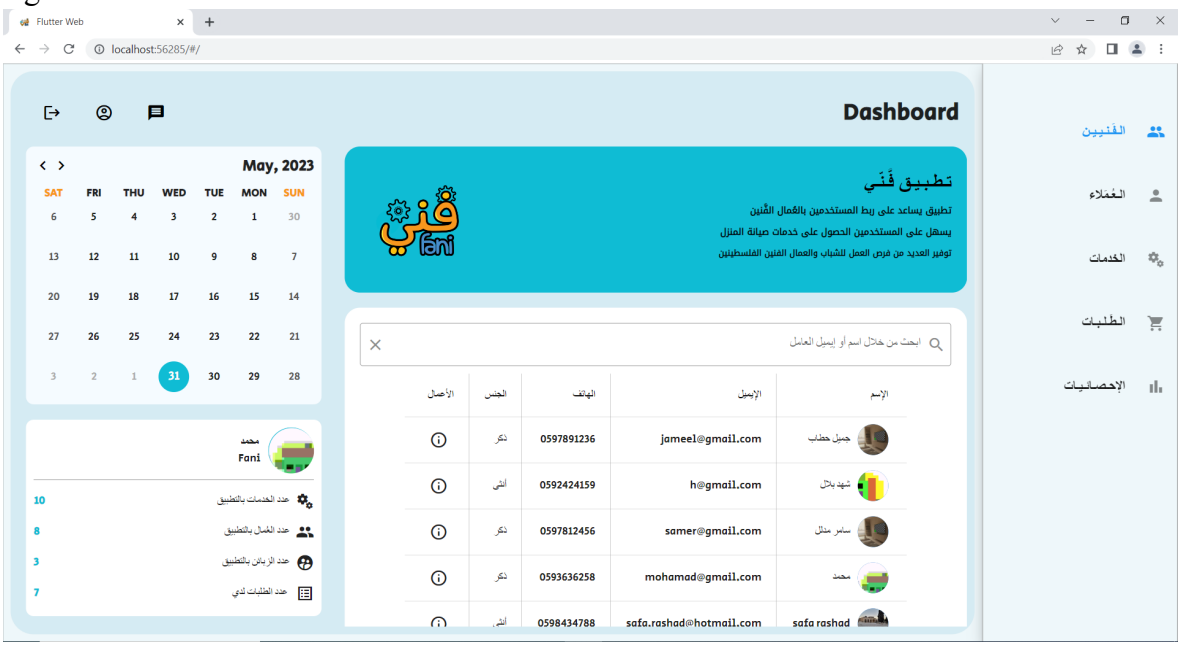


Figure 53: worker dashboard

As a worker, you have access to view all users and their profiles within the application. Furthermore, as a worker, you can see the specific orders that each user has placed with you. This feature allows you to track and manage the services or tasks requested by individual users, ensuring efficient workflow and effective communication between you and your clients.

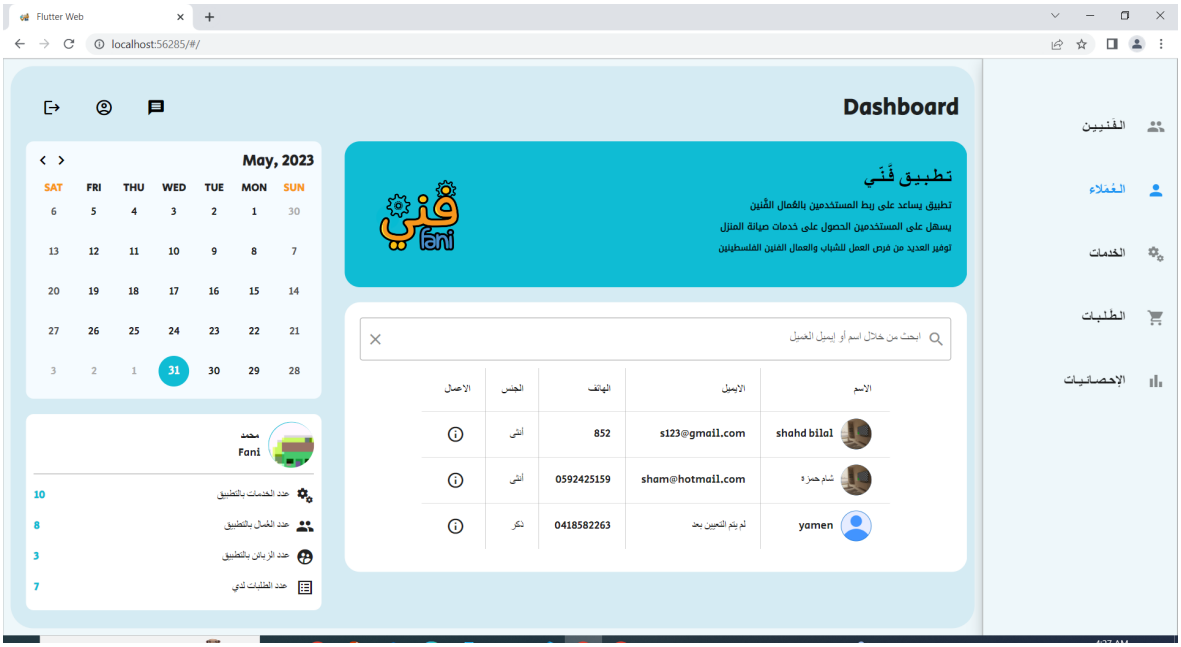


Figure 54: worker_customer

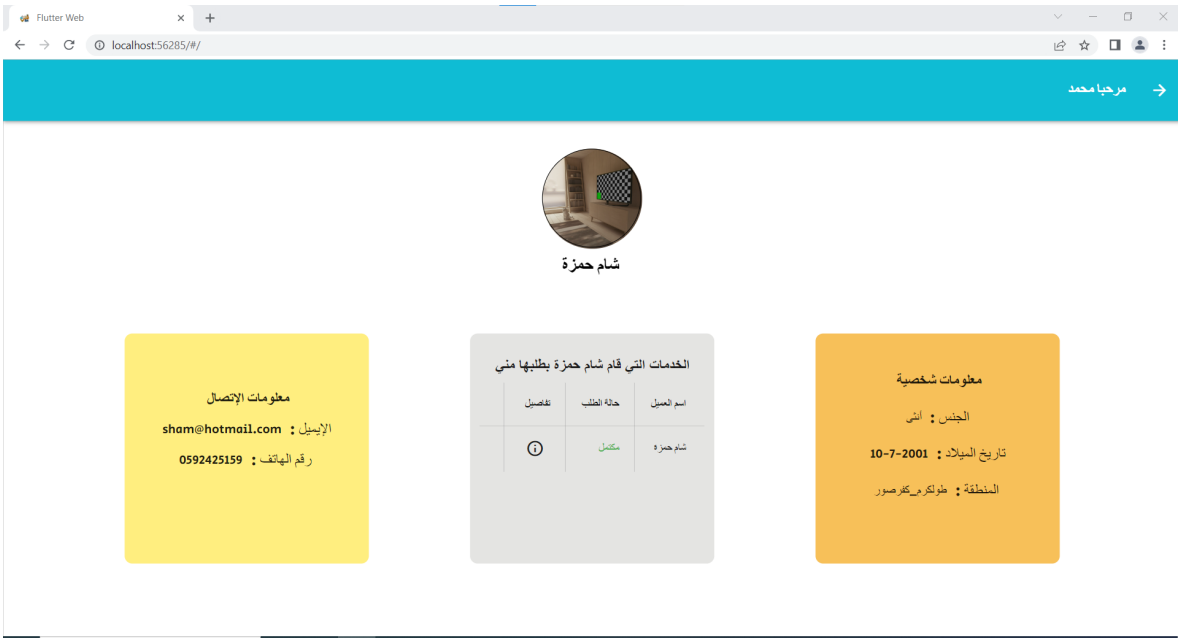


Figure 55: worker see profile customer

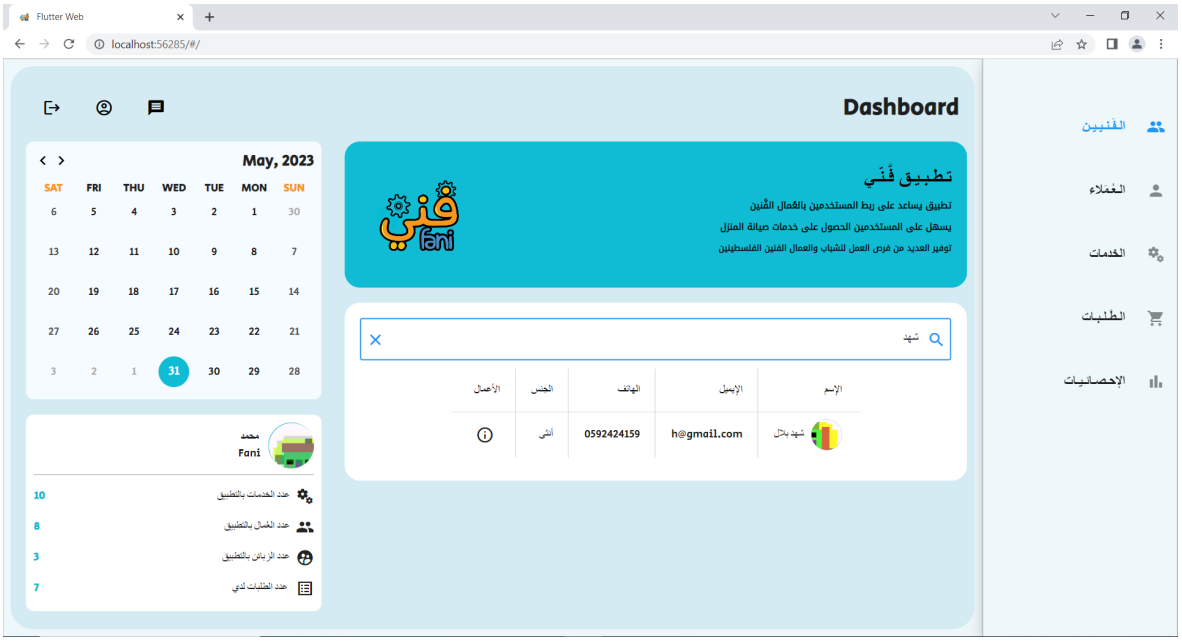


Figure 56: worker_worker search page

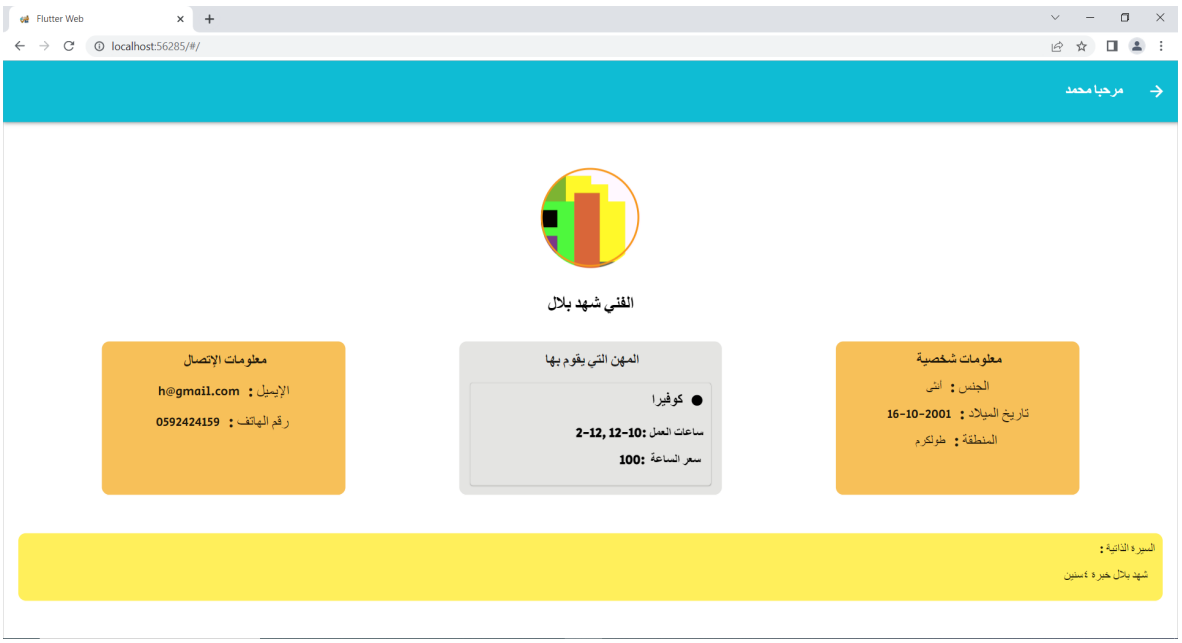


Figure 57: worker see profile worker

by using web page works can communicate with any other worker or any user and with the admin

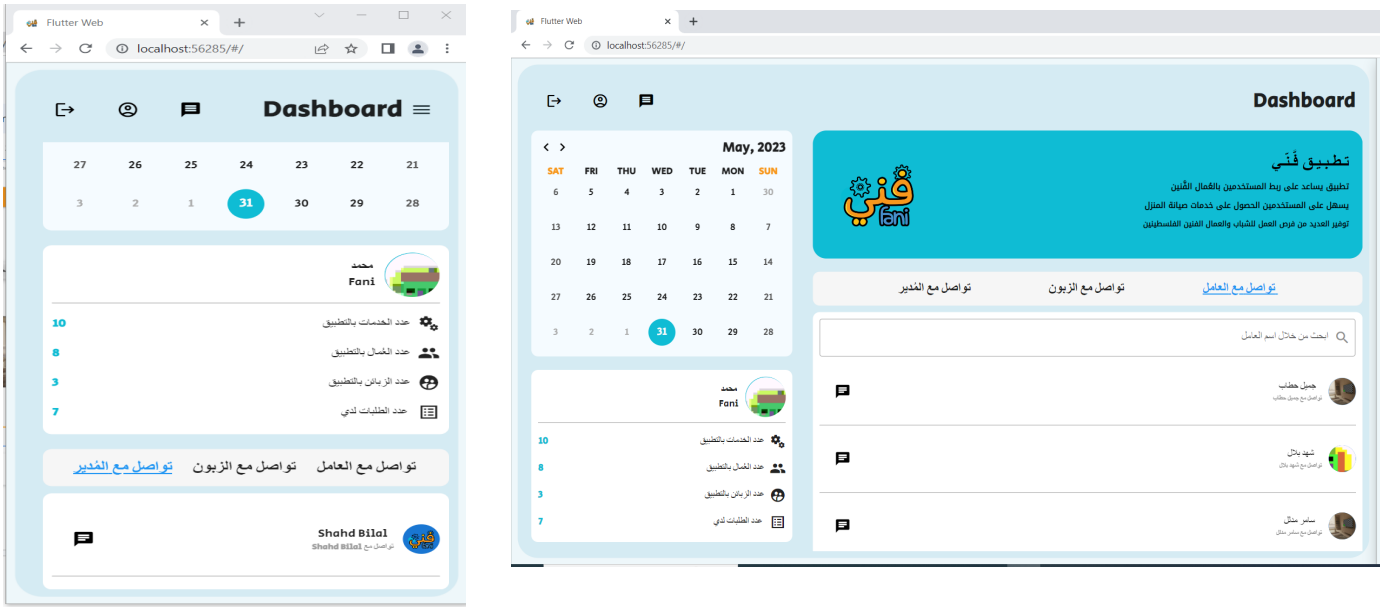


Figure 58: worker web chat member

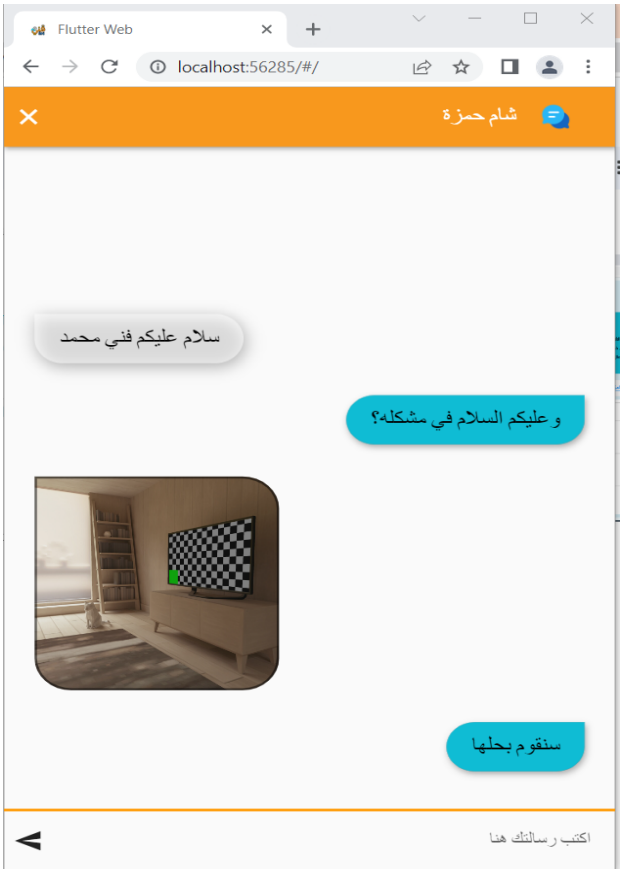


Figure 59.1: worker-user web chat

Figure 59.2: worker-admin web chat

The orders page for worker displays a comprehensive list of all the orders placed by users. It provides detailed information about each order, By having access to this information, workers can effectively prioritize their workload and ensure timely completion of each order.

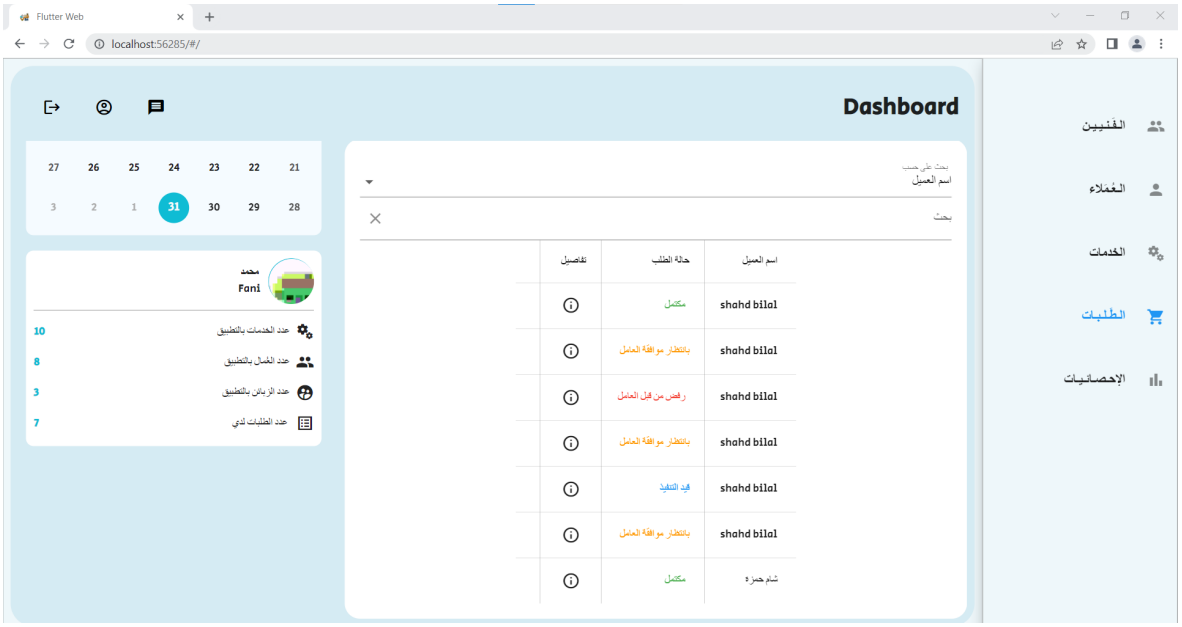


Figure 60: worker-orders web

statistic page for a worker:

Figure 61: worker order and days

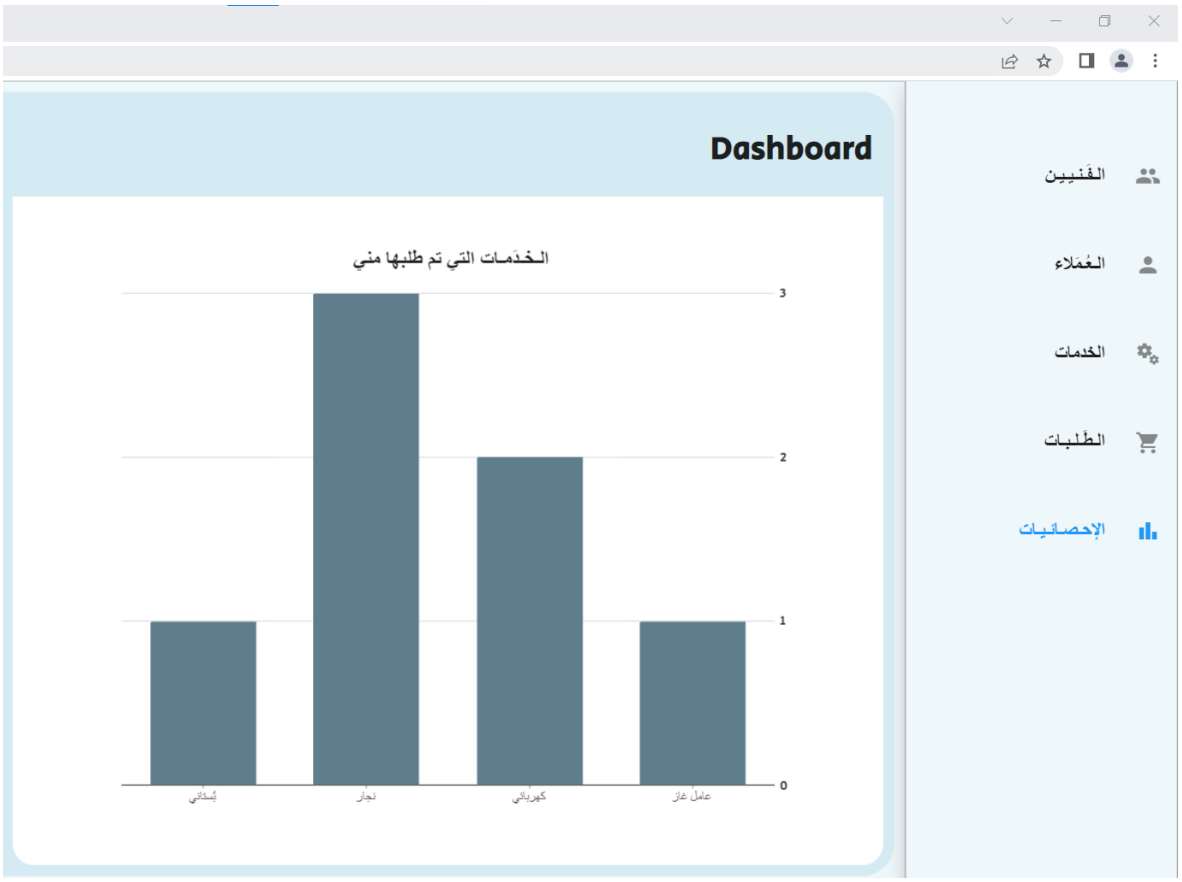


Figure 62: worker services and order

worker can see his profile page and edit it:

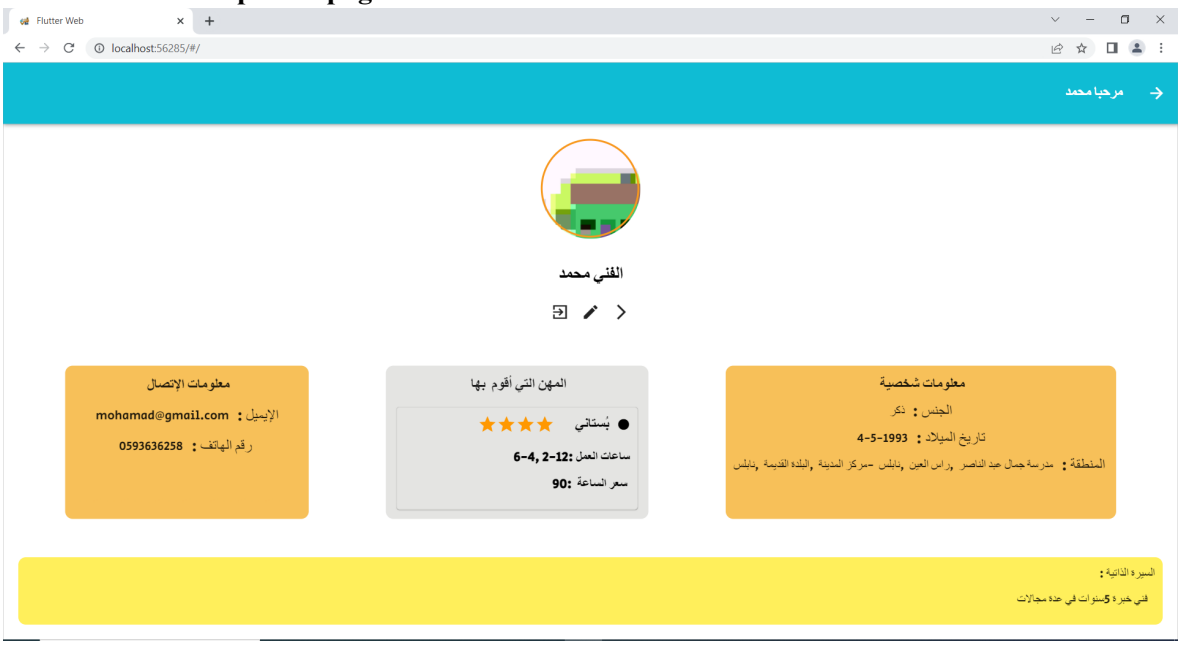


Figure 63: worker profile web page

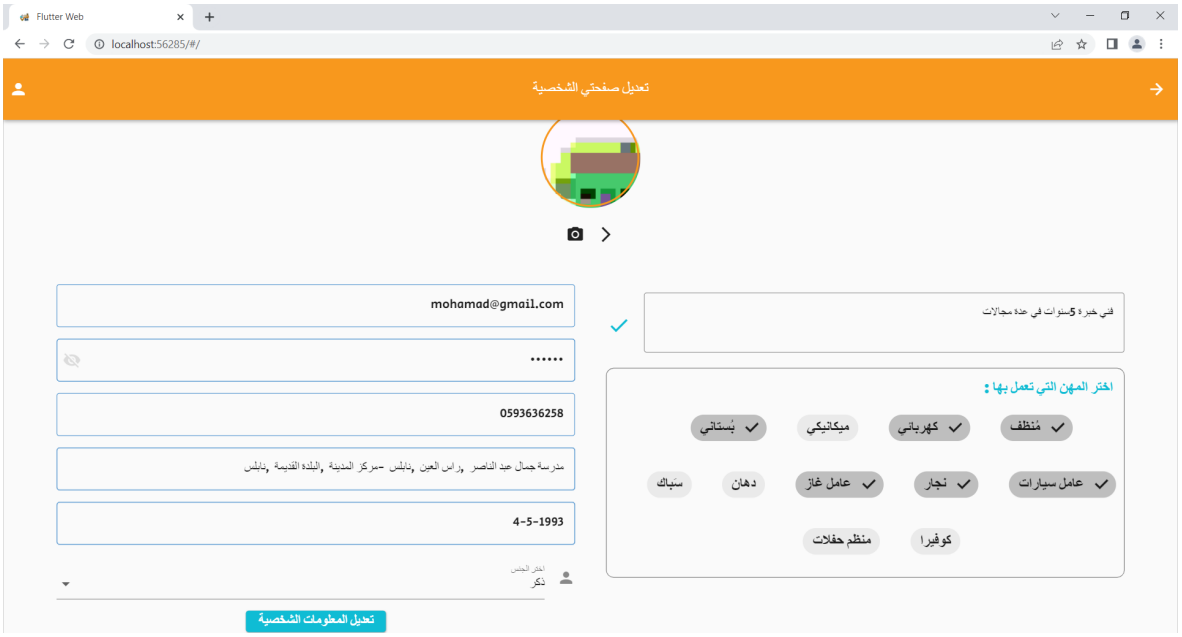


Figure 64: worker edit profile web page

5. Results and Discussion

The developed application has successfully fulfilled its intended purpose, providing high efficiency and a user-friendly experience, as demonstrated by the visuals presented earlier. The target audience, primarily parents and individuals with limited technological knowledge, guided our focus on creating a simple, intuitive, and visually clear design to ensure ease of use for all users.

However, the app development journey presented various technical and practical challenges. On the technical front, frequent updates and evolving software versions posed obstacles that sometimes clashed with our existing infrastructure. Establishing a robust and logical database system that efficiently retrieves and stores data proved demanding.

Practical challenges included accurate location mapping within Palestine. The available maps lacked the same level of accuracy and seamlessness compared to other regions due to the ongoing challenges posed by an occupation seeking to undermine our presence. Collecting relevant and practical data tailored to the local environment and community required careful deliberation and adaptation to ensure alignment with the unique context.

Despite these challenges, the application was successfully developed according to our vision, yielding satisfying outcomes. Users can access a diverse range of technical and professional services, which can be expanded and customized to cater to evolving needs. The application allows users to select their desired service and connect with workers who meet their specific criteria.

Emphasizing user satisfaction, the application prioritizes transparency and accountability. It provides a transparent and efficient platform that facilitates seamless communication between users and workers. This fosters an environment conducive to exceptional craftsmanship, excellent customer service, and safeguards against fraudulent activities or misconduct. Overall, the application has achieved the desired objectives and addresses the identified challenges in the maintenance service industry.

6. Conclusions and Recommendations

In this project, we have successfully developed an application that provides users with convenient access to a wide range of professional and technical services. The application is designed to be user-friendly, catering to people of all age groups and ensuring that everyone can easily benefit from its features.

6.1. Summary

Our application aims to meet diverse needs in daily life by offering both primary and ancillary services. Users can enjoy flexible scheduling options, allowing them to arrange appointments with service providers based on their specific circumstances and requirements. We also provide alternative choices to accommodate user preferences.

Customer satisfaction and quality service are paramount in our platform. To address any concerns or issues, we have included feedback mechanisms for users to provide feedback on their experiences. Additionally, we offer multiple communication channels to facilitate seamless interaction between users, service providers, and a dedicated third-party mediator whenever necessary.

6.2. Future work

As we witness continuous progress in technology, it is essential for our application to remain dynamic and responsive to emerging discoveries and advancements. Therefore, we have identified several areas for future work and improvement:

* Integration of Artificial Intelligence: We have already incorporated artificial intelligence into the system to prioritize user preferences. In the future, we can further leverage AI capabilities to suggest relevant services based on real-time household conditions. For example, the application can sense and link to smart home devices, providing personalized service recommendations based on the user's current needs.

* Expansion of Services: As user needs evolve, we aim to expand the range of services offered on the platform. This can include incorporating new categories of services or partnering with additional service providers to ensure a comprehensive and diverse range of options for our users.

* Continuous User Feedback and Improvement: We value user feedback and believe it is crucial for the ongoing improvement of our application. We will continue to actively seek feedback from users, analyze their suggestions, and implement necessary updates and enhancements to enhance their experience.

By focusing on these areas for future work, we aim to ensure that our application remains up-to-date, user-centric, and adaptable to the ever-changing needs and technological advancements.

7. References

1. TaskRabbit: TaskRabbit app. Available on Google Play Store.
2. Handyman (Errands): Handyman app (Errands). Available on Google Play Store.
3. Firebase: Firebase website. Available at: https://firebase.google.com/
4. MongoDB: MongoDB website. Available at: https://www.mongodb.com/.