

Project Team



Ahmed Mohamed Ibrahim



Eslam Mohamed Ali



Zeyad Elsayed Elsman



Mahmoud Abdellah Mohamed



Mariam Abdulrady Heshmat



Nada Mostafa Amin



Hager Yasser Abdulgaber

Acknowledgment

We would like to thank (Dr. Elnomery Zanaty) and (Dr. Hamdi Hassan)

And (Dr. Mohamed Heshmat) for their effort and for helping us in this project

And We would like to express our special thanks of gratitude to our professor (*Dr. Hanan Hamed*) who keeps any effort in encouraging us to do a great job, providing our group with valuable information and advices to be better each time. Thanks for the continuous support and kind communication which had a great effect regarding to feel interesting about what we are working on. We are thankful to him.

Finally, we would also like to thank our friends who helped us a lot in finalizing this project within the limited time frame.

Acronyms & Abbreviations

App	Application
UI	User interface
UX	User Experience
PHP	Hypertext preprocessor
DFD	Data Flow diagram
ERD	Entity-Relationship Diagram
IDE	integrated development environment
iOS	iPhone Operating system
MacOS	Macintosh operating system
Js	JavaScript
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
DevOps	Development Operations
SQL	Structured Query Language
Api	Application programming interface

Abstract

In an age marked by rapid urbanization and hectic lifestyles, the demand for accessible, on-demand services within our neighborhoods has become increasingly apparent. This project delves into the creation and deployment of a cutting-edge mobile application and web application, poised to revolutionize the way we engage with essential services in our local communities.

This project is a pioneering initiative that brings together a diverse range of services under one user-friendly platform. From connecting residents with skilled cleaners to facilitating instant access to plumbing and electrical professionals, our project seeks to streamline and simplify the often-cumbersome process of securing essential services.

By harnessing the power of technology, our goal is to empower individuals to efficiently meet their household needs while fostering a sense of community support. This report will explore the features, functionality, and potential impact of this innovative application, illustrating how it stands to reshape the landscape of neighborhood services and enhance the overall well-being of residents. Join us as we unpack the intricacies of this transformative project, poised to redefine the way we experience and engage with essential services in our neighborhoods.

Table of Contents

Acknowledgment	1-I
Acronyms & Abbreviations	1-II
Abstract	1-III
List of Figures	1-VIII
List of Tables	1-X
1 Chapter 1 Introduction.....	1
1.1 Background	1
1.2 Problem Statement	1
1.2.1 Providers of these services	2
1.2.2 The availability of these services	2
1.2.3 Prices for these services	2
1.3 Objectives.....	3
1.3 Solving.....	4
2 Chapter 2 Requirements and Constraints	6
2.1 Requirements.....	6
2.1.1 Introduction	6
2.1.2 Functional Requirements.....	6
2.1.3 Non-Functional Requirements	7
2.1.4 System Constraints	7
2.1.5 Summary	8
2.2 Constraints.....	8

2.2.1	Scope Constraints	8
2.2.2	Quality Constraints.....	9
2.2.3	Resource Constraints	9
2.2.4	Technical Constraints	9
2.2.5	Risk Constraints	10
2.2.6	Communication Constraints	10
2.2.7	Legal and Compliance Constraints	10
3	Chapter 3 System Analysis	12
3.1	Use case diagram.....	12
3.1.1	Use Case Diagram Components:.....	12
3.1.2	Use case scenario:	14
3.2	Data Flow diagram (DFD)	18
3.2.1	URBAN DFD	19
3.3	Entity-Relationship Diagram (ERD).....	21
3.4	Activity Diagram.....	22
4	Chapter 4 Tools and Design	25
4.1	Tools Product Perspective	25
4.1.1	Application Function.....	25
4.1.2	Main Work	25
4.1.3	Website Content	26
4.1.4	Mobile Content using Xamarin.	28
4.2	UI/UX design	30

4.2.1	Logo of application	30
4.2.2	Intro of application and Welcome Page	31
4.2.3	Select Your Role	32
4.2.4	Description Page.....	33
4.2.5	Information Page	34
4.2.6	Finger checking	35
4.2.7	Login page.....	36
4.2.8	Sign in using phone	37
4.2.9	Password Forget	38
4.2.10	Sign up page.....	39
4.2.11	Services Page	40
4.2.12	Map Choose your provider.	41
4.2.13	Information of the provider.....	42
4.2.14	Provider request	43
4.2.15	Successful Page.....	44
4.2.16	Profile Page	45
4.2.17	Settings Page	46
4.2.18	Change Password	47
4.2.19	Ask AI Page	48
4.2.20	Requests Page	49
4.2.21	Provider Sign in	50
4.2.22	Provider Requests	51

4.2.23 Provider Profile	52
4.3 Website Dashboard Ui:	53
4.3.1 Admin Sign in	53
4.3.2 Main Page:.....	53
4.3.3 Show Services:	54
4.3.4 Show Providers:	54
4.3.5 Show Users:.....	55
4.3.6 Add Service:	55
4.3.7 Add Provider:	56
5 Chapter 5 System implementation	58
5.1 Mobile App Code (Xamarin):	58
5.1.1 Page 1	58
5.1.2 Page 2	65
5.1.3 Page 3	67
5.1.4 Page 4	74
5.2 Website Dashboard Code:	132
5.2.1 Main Page:.....	132
5.2.2 Show Services	148
5.2.3 Add Service	160
5.3 Backend Code (.net):.....	173
Resources	177

List of Figures

Figure 1-1 Nearby services application	4
Figure 3-1 Use Case Diagram.....	13
Figure 3-2 Context Level	19
Figure 3-3 DFD Level 1.....	20
Figure 3-4 ERD.....	21
Figure 3-5 Activity Diagram.....	23
Figure 4-1 Logo of application	30
Figure 4-2 Intro and welcome page for App.....	31
Figure 4-3 Select Role.....	32
Figure 4-4 Description Page	33
Figure 4-5 Information Page	34
Figure 4-6 Finger checking	35
Figure 4-7 Login	36
Figure 4-8 Sign in using phone.....	37
Figure 4-9 Forget Password	38
Figure 4-10 Sign up page	39
Figure 4-11 Services Page	40
Figure 4-12 Map to choose any provider.....	41
Figure 4-13 Info of provider	42
Figure 4-14 Provider Request	43
Figure 4-15 Successful Page	44
Figure 4-16 Profile Page	45

Figure 4-17 Setting Page.....	46
Figure 4-18 Change Password	47
Figure 4-19 Ask Ai Page.....	48
Figure 4-20 Requests page.....	49
Figure 4-21 Provider sign in page.....	50
Figure 4-22 Provider Requests page	51
Figure 4-23 Provider Profile page.....	52
Figure 4-24 Admin Login	53
Figure 4-25 Main Page.....	53
Figure 4-26 Show service	54
Figure 4-27 Show Providers Page.....	54
Figure 4-28 Show Users Page.....	55
Figure 4-29 Add Service Page	55
Figure 4-30 Add Providers Page.....	56
Figure 5-1	173
Figure 5-2	173
Figure 5-3	174
Figure 5-4	174
Figure 5-5	175
Figure 5-6	175
Figure 5-7	176

List of Tables

Table 3-1 login	14
Table 3-2 Set location.	14
Table 3-3 Search for services.....	14
Table 3-4 Choose Services.....	15
Table 3-5 Request Service-provider	15
Table 3-6 Describe the problem.....	16
Table 3-7 Edit profile.....	16
Table 3-8 Show Requests.....	16
Table 3-9 Confirm the request	17
Table 3-10 Manage system.	17
Table 3-11 Add Providers	17

Chapter 1

Introduction

1 Chapter 1 || Introduction

1.1 Background

This project serves customers, whether they are original residents of the city or have recently moved to a particular city and do not know much about the nearby services in this city.

Through this application (whether you use a mobile application and a web application), the customer can choose the service he wants, such as (plumbing, electricity-related services, house cleaning-related services, pest control services, painting services...and so on, such necessary services in any home).

The customer can attach photos related to the requested service to help choose the appropriate worker to perform this requested service.

This project will make it much easier for people who were looking for workers to perform these services and took a long time to search for them and may not find them.

This project is implemented as a mobile application and a web application; To suit all human uses on various devices.

1.2 Problem Statement

Many people move to certain cities and reside there, but when they face a problem with household chores such as plumbing, electricity, and such problems, they also face a problem in finding a technician who is an expert in these problems.

It is not easy to bring a worker to solve such problems when you are confident and know the extent of his professionalism and experience in his field of work and to ensure that the work is carried out on time.

After researching and tracking these problems, we found that these problems are divided into:

1. Providers of these services.
2. The availability of these services.
3. Prices for these services.

1.2.1 Providers of these services

We noticed that people take a lot of time to reach home service providers, yet they complain about the lack of professionalism of these workers.

It is also difficult for people to trust the providers of these services when they do not have any previous knowledge of them.

1.2.2 The availability of these services

In many cities and neighborhoods there are no workers for such services and it is very difficult to find these workers.

1.2.3 Prices for these services

Due to the dire economic conditions, there is a significant increase in all prices, including the prices of home services such as maintenance, plumbing, electrical faults, and so on.

But there are many workers in these services who exploit people and increase their wages above the normal level, which forces many people to dispense with these workers.

1.3 Objectives

- Connect local residents with reliable and skilled home service professionals.
- Streamline the process of booking and scheduling home services within the designated area.
- Enhance customer satisfaction by ensuring timely and quality home service deliveries.
- Offer a diverse range of home services, including maintenance, repairs, and renovations, catering to the specific needs of the local community.
- Prioritize user convenience through a user-friendly interface and seamless communication channels.
- Foster local economic growth by supporting and promoting local home service businesses.
- Ensure cost-effectiveness for users while maintaining fair compensation for service providers.
- Promote safety and security by vetting and verifying the credentials of all registered service professionals.
- Utilize technology to provide real-time updates, notifications, and tracking features for scheduled services.
- Continuously innovate and expand service offerings based on the evolving needs and preferences of the community.

- Strive for inclusivity by ensuring accessibility features for all users and promoting diversity among service providers.



Figure 1-1 Nearby services application

1.3 Solving

This project was built to solve the problems mentioned previously and to achieve the aforementioned goals as well.

The idea of the project is that it provides home services (such as plumbing, electricity-related services, cleaning services, home maintenance, and so on), all of which the customer can obtain through a mobile application or web application and choose the service that suits him.

There are many service providers who are highly experienced and trusted workers. Companies employ them and ensure their ability to perform these services to the fullest extent and at very reasonable prices.

Chapter 2

Requirements &

Constraints

2 Chapter 2 || Requirements and Constraints

2.1 Requirements

2.1.1 Introduction

This chapter outlines the system requirements for the proposed project, which focuses on providing assistance to individuals within a specific geographical area by facilitating access to nearby services.

Understanding the requirements is crucial for the successful development and implementation of the system.

2.1.2 Functional Requirements

2.1.2.1 User Registration and Authentication:

- The system shall allow users to register accounts using valid credentials.
- Users must be authenticated before accessing personalized services.
- Passwords must be securely stored using industry-standard encryption techniques.

2.1.2.2 Service Information Retrieval:

- The system shall provide a search functionality for users to find nearby services.
- Users shall be able to filter search results based on service type, ratings, and distance.

2.1.2.3 Service Listings:

- The system shall display detailed listings for each service, including contact information.

2.1.2.4 Geo-Location Integration:

- The system must integrate with geo-location services to identify the user's location.
- Users shall have the option to manually input their location if geo-location services are unavailable.

2.1.3 Non-Functional Requirements

2.1.3.1 Performance:

- The system must respond to user queries within 3 seconds.

2.1.3.2 Usability:

- The user interface shall be intuitive and accessible to individuals with varying levels of technical proficiency.

2.1.3.3 Security:

- User data, including personally identifiable information, must be stored securely and comply with data protection regulations.

2.1.4 System Constraints

The system must operate on common web browsers (Chrome, Firefox, Safari, Edge).

2.1.5 Summary

This chapter outlined the functional and non-functional requirements, as well as system constraints, for the proposed project. The next chapter will delve into the system design, which is informed by these requirements.

This is a high-level overview, and you would typically go into more detail for each requirement, including use cases, scenarios, and potential challenges. Adapt and expand based on the specific needs and complexity of your project.

2.2 Constraints

2.2.1 Scope Constraints

2.2.1.1 Project Scope:

The project involves the development of the “The Nearest Services App”, a mobile application and web application that facilitates on-demand home services, including plumbing, electrical work, and cleaning.

2.2.1.2 Scope Changes:

Changes to the initial scope, such as adding new service categories or features, must go through a formal change request process.

2.2.2 Quality Constraints

2.2.2.1 Quality Standards:

“The Nearest Services App” must meet industry-accepted standards for user experience, security, and responsiveness.

2.2.2.2 Testing and Validation:

Rigorous testing, including functional testing, security testing, and user acceptance testing, will be conducted to ensure the app meets quality criteria.

2.2.3 Resource Constraints

2.2.3.1 Human Resources:

The project team consists of 7 members with expertise in mobile app development, UI/UX design, Web development.

2.2.3.2 Equipment and Facilities:

Adequate computing resources and testing devices must be provided to the project team.

2.2.4 Technical Constraints

2.2.4.1 Technology Requirements:

“The Nearest Services App” will be developed using (Xamarin , PHP , React.js). Any proposed changes to these technologies must be evaluated for compatibility.

2.2.5 Risk Constraints

2.2.5.1 Risk Management:

Risks such as app security vulnerabilities, compatibility issues with certain devices.

2.2.5.2 Unforeseen Events:

A contingency plan is in place to address unforeseen events that may disrupt the project, such as server outages or unexpected regulatory changes.

2.2.6 Communication Constraints

2.2.6.1 Communication Plan:

Efficient communication channels must be established between users, service providers, and the support team. In-app messaging and notifications will be crucial for timely updates and addressing user inquiries.

2.2.7 Legal and Compliance Constraints

2.2.7.1 Legal Requirements:

“The Nearest Services App” must comply with data protection laws and industry regulations governing user data and payment transactions.

2.2.7.2 Intellectual Property:

The app must respect all intellectual property rights, and any third-party services or libraries used must be properly licensed.

Chapter 3

System Analysis

3 Chapter 3 || System Analysis

It is a process of collecting and interpreting facts, identifying the problems, and decomposition of a system into its components.

System analysis is conducted for the purpose of studying a system or its parts in order to identify its objectives. It is a problem solving technique that improves the system and ensures that all the components of the system work efficiently to accomplish their purpose. (1)

Analysis specifies what the system should do.

3.1 Use case diagram

A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. It shows how a system interacts with external entities. So, it is relatively sparse the details of how the system behaves internally and how the external environment is arranged.

3.1.1 Use Case Diagram Components:

- Actor
- Use cases
- Associations
- System boundary

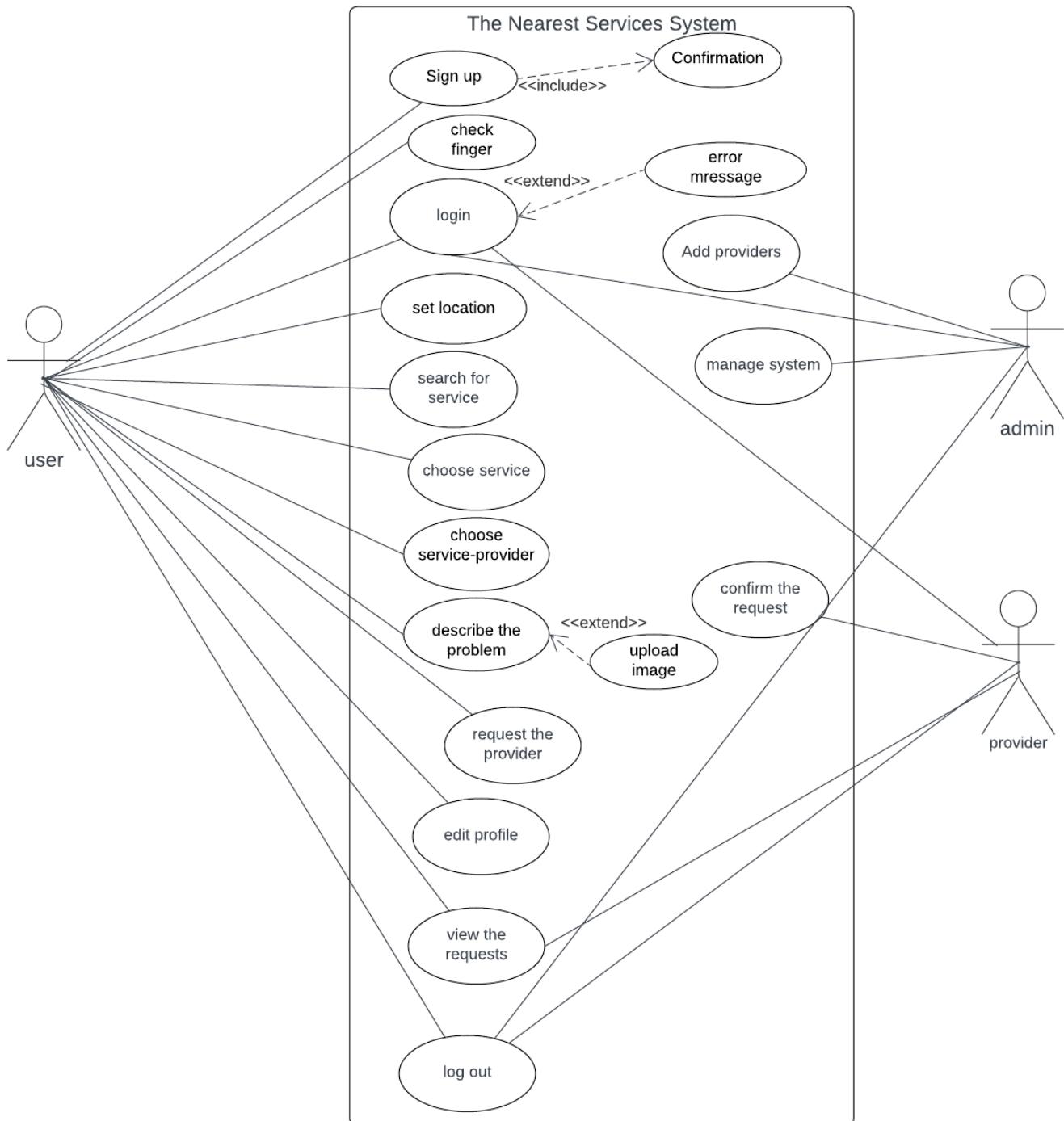


Figure 3-1 Use Case Diagram

The use case diagram shown in figure (3-1) illustrates the interaction between the system and the main actors (Admin, customer and provider).

3.1.2 Use case scenario:

Table 3-1 login

System	Nearest Services System
Use case	login
Actor	User
Data	The user should insert his identification information as (User name, password, Email).
Stimulus	User can browse the app

Table 3-2 Set location.

System	Nearest Services System
Use case	Set location
Actor	User
Data	The user should insert his current location
Stimulus	System confirm the user location so he can search for services near

Table 3-3 Search for services.

System	Nearest Services System
Use case	Search for services
Actor	User

Data	After setting the location, The user can search for the service he wants
Stimulus	Facilitate access to the required service quickly

Table 3-4 Choose Services

System	Nearest Services System
Use case	Choose Service
Actor	User
Data	Click on the service to know the people available to perform the service through the map

Table 3-5 Request Service-provider

System	Nearest Services System
Use case	Request Service-provider
Actor	User
Data	After choosing the right person to perform the service I click on him and ask for the service
Stimulus	The user gets the right person for him to perform the required service

Table 3-6 Describe the problem

System	Nearest Services System
Use case	Describe the problem
Actor	User
Data	The user can describe the matter that he wants to serve if possible
Stimulus	Accuracy in obtaining the user the best solution for the service to be performed

Table 3-7 Edit profile

System	Nearest Services System
Use case	Edit profile
Actor	User
Data	The user can edit his profile
Stimulus	The system becomes more flexible for user

Table 3-8 Show Requests

System	Nearest Services System
Use case	Show Requests
Actor	User , Provider
Data	The user and the provider could see the requests

Stimulus	The Requests become accessible to the user and the provider and could check it
-----------------	--

Table 3-9 Confirm the request

System	Nearest Services System
Use case	Confirm the request
Actor	Provider
Data	the provider can confirm or discard the request
Stimulus	Make communication easy with user and the provider could review his requests

Table 3-10 Manage system.

System	Nearest Services System
Use case	Manage system
Actor	Admin
Data	Admin is responsible manage the system, The communication between providers and users and the privacy

Table 3-11 Add Providers

System	Nearest Services System
Use case	Add Providers

Actor	Admin
Data	Admin can add or delete providers

3.2 Data Flow diagram (DFD)

Is a graphical tool for representing the flow of data of a system or a process.

It shows the inputs, outputs, transformations, and storage of data in a system.

Learn the characteristics, components, levels, advantages, and disadvantages of DFD with examples and diagrams. (2)

Elements of a DFD:

- Processes.
- Data Flows.
- Data stores.
- External entities.

3.2.1 URBAN DFD

3.2.1.1 Context level:

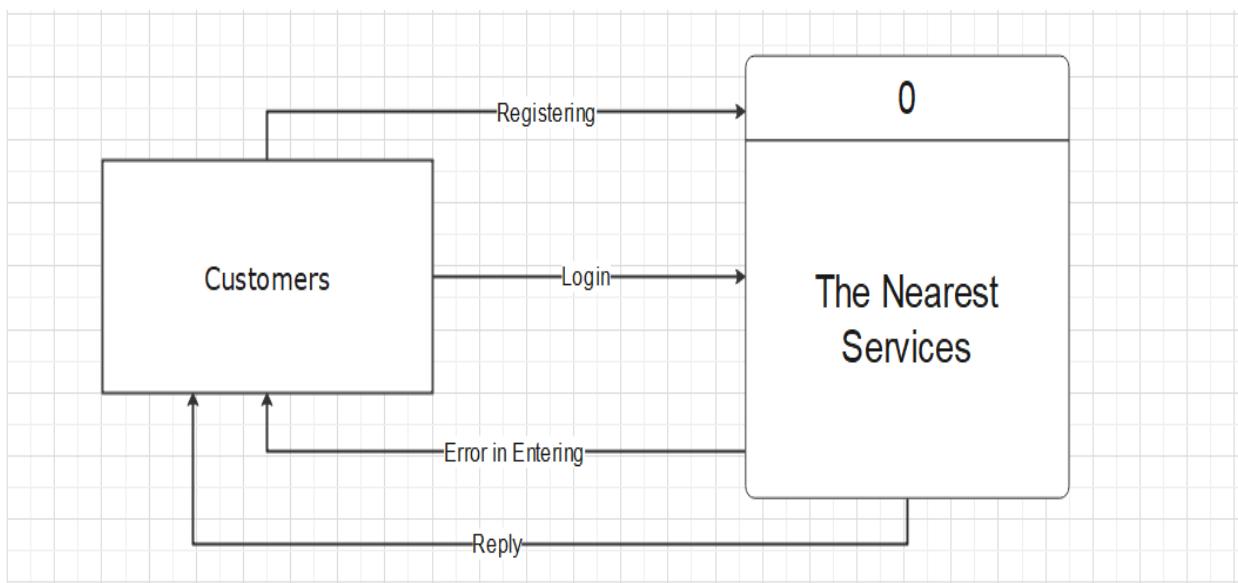


Figure 3-2 Context Level

The DFD shown in figure (3-2) shows the data flow of the context level which includes the main process of the system which is offering services that the application provides and the main entity (Users).

3.2.1.2 Level 1:

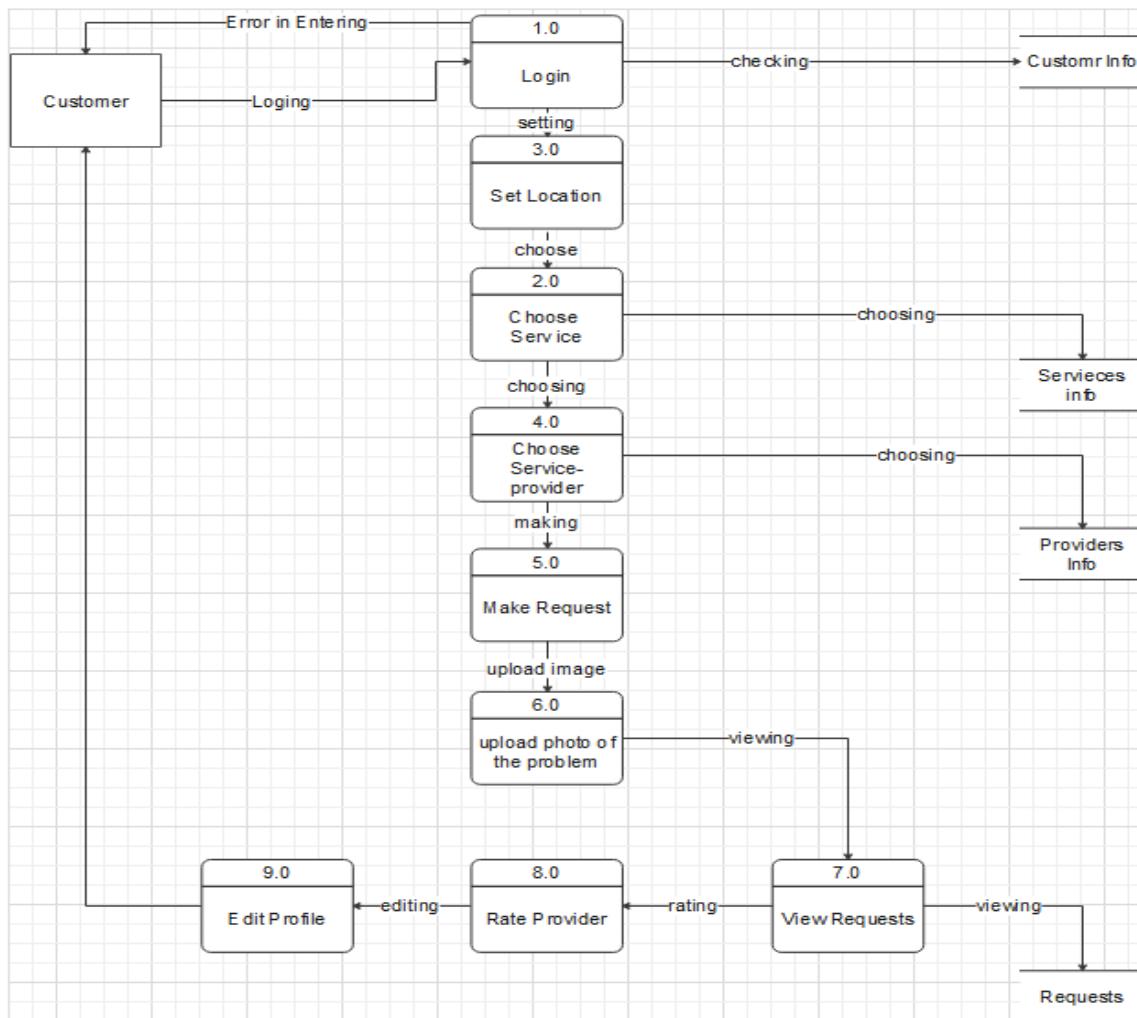


Figure 3-3 DFD Level 1

The DFDs shown in figures (3-3), shows the data flow of the second level which include the analysis of the Enroll, and Demand processes in the first level which includes some process(Signup, Login, Serve, Request) and the main entities related to this process (User and Admin) and the main data store(User data, User Login, Providers).

3.3 Entity-Relationship Diagram (ERD)

Entity Relationship Diagram, also known as ERD, ER Diagram or ER model, is a type of structural diagram for use in database design. An ERD contains different symbols and connectors that visualize two important information: **The major entities within the system scope**, and the **inter-relationships among these entities**.

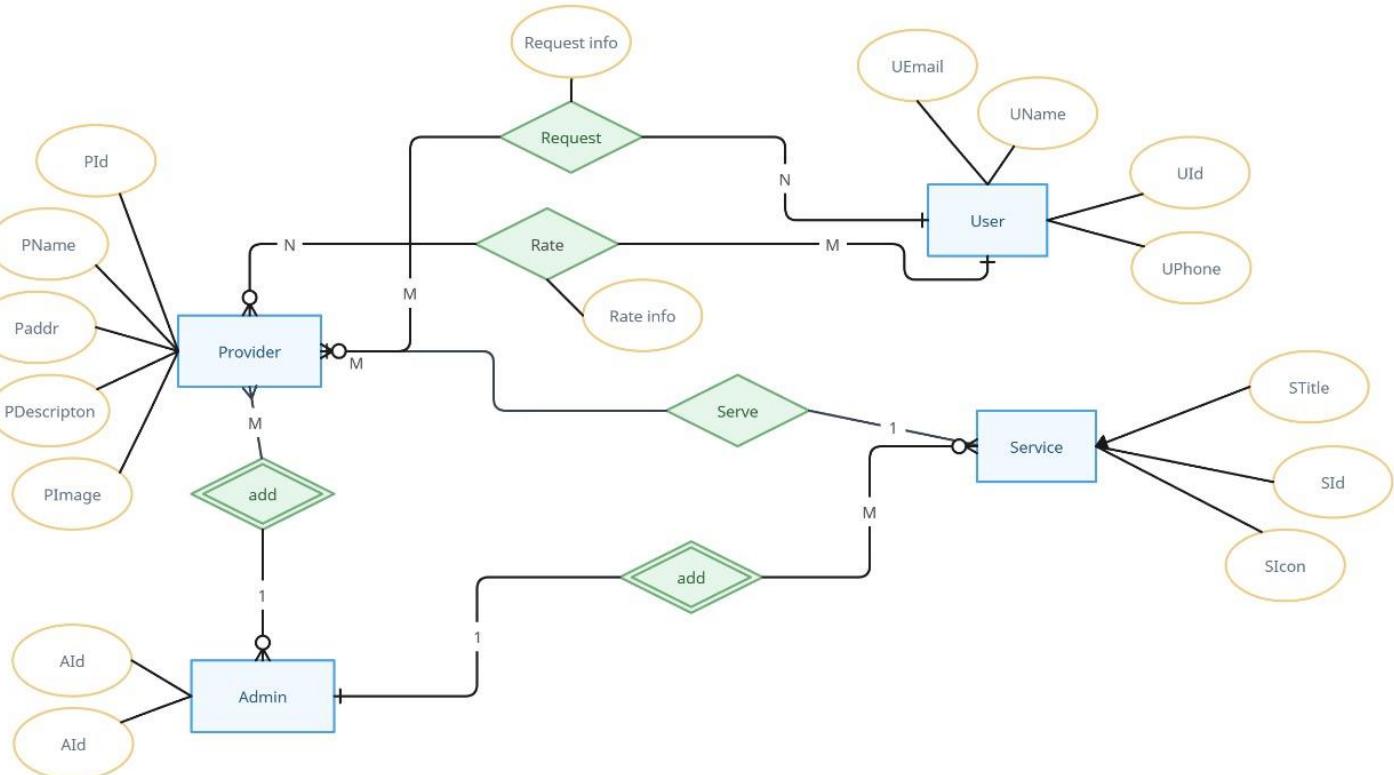


Figure 3-4 ERD

3.4 Activity Diagram

Activity Diagrams describe how activities are coordinated to provide a service which can be at different levels of abstraction.

Typically, an event needs to be achieved by some operations, particularly where the operation is intended to achieve a number of different things that require coordination, or how the events in a single use case relate to one another, in particular, use cases where activities may overlap and require coordination. It is also suitable for modeling how a collection of use cases coordinate to represent business workflows.

INSERVICE

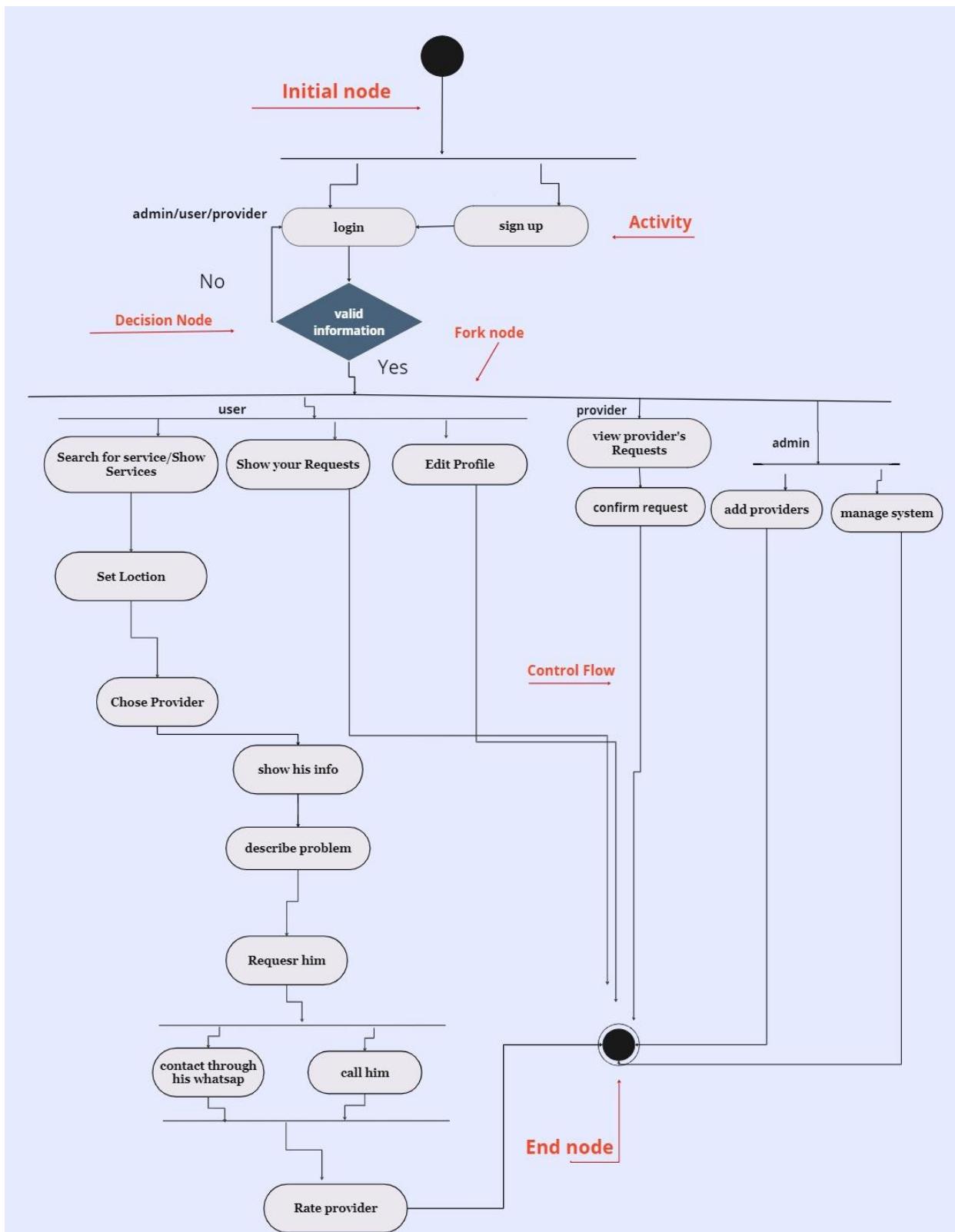


Figure 3-5 Activity Diagram

Chapter 4

Tools and Design

4 Chapter 4 || Tools and Design

4.1 Tools Product Perspective

A Services application contains the following: -

- Admin: It includes all the admins in the application.
- Provider: It includes all the services providers in the application.
- Customer: It includes all the users in the application.

4.1.1 Application Function

By using this System/Application the customer has to sign up/sign in so he can search for the service he wants, choose the closest and best service provider available to you and he can upload a picture of the malfunction or the matter that he wants to serve if possible.

By using this System/Application the provider has to receive the request and confirm it.

By using this System/Application the admin has to sign in so he can view form.

4.1.2 Main Work

Visual Studio: one of the versions of Microsoft's integrated development environment (IDE), is versatile and widely used for various software development purposes. It uses in Mobile App Development. It enables the creation of mobile applications for iOS and Android platforms. Using frameworks like Xamarin or React Native, developers can build cross-platform mobile apps that share a significant portion of their code base, saving time and effort. It is just an example of the uses of Visual Studio. Its wide range of features, language support, and

integration with various frameworks and platforms make it a versatile IDE for different types of software development projects.

Visual Studio Code: is a dual-licensed source-code editor made by Microsoft for Windows, Linux and macOS. In the Stack Overflow 2019 Developer Survey, Visual Studio Code was ranked the most popular developer environment tool, with 50.7% of 87,317 respondents reporting that they use it. Visual Studio Code is a source-code editor that can be used with a variety of programming languages, including Java, JavaScript, Go, Node.js, Python and C++. It is based on the Electron framework, which is used to develop Node.js Web Applications that run on the Blink layout engine. Visual Studio Code employs the same editor component (codenamed "Monaco") used in Azure DevOps (formerly called Visual Studio Online and Visual Studio Team Services).

4.1.3 Website Content

The front-end of a website is merely how the information is presented to the users, and it fetches everything from the backend to display in user browsers. HTML (HyperText Markup Language) is used to design web pages using a markup language. Hypertext defines the link between the web pages. A markup language is used to define the text document within tag which defines the structure of web pages.

HTML is a markup language that is used by the browser to manipulate text, images, and other content to display it in the required format. CSS (Cascading Style Sheets) is a stylesheet language used to design the webpage to make it attractive. The reason for using CSS is to simplify the process of making web pages presentable. CSS allows you to apply styles to web pages. More importantly, CSS enables you to do this independent of the HTML that makes up each web page. JavaScript (JS) is a lightweight, interpreted, or just-in-time compiled programming

language with first class functions. JavaScript is a prototype-based, multi-paradigm, single-threaded, dynamic language, supporting object-oriented, imperative, and declarative (e.g. functional programming) styles. Font Awesome is the Internet's icon library and toolkit, used by millions of designers, developers, and content creators.

Font Awesome contains over 7,000 new icons, so you're sure to find what you need for your project. Plus they're more consistent and easier to use. Font Awesome 6 makes it even easier to use icons where you want to. More plugins and packages to match your stack. Less time wrestling browser rendering.

The backend of a website is the place that contains all the data and relevant information that is to be shown to the visitors with the help of a browser and Code written by back-end developers helps to communicate the database information to the browser.

The PHP Hypertext Preprocessor (PHP) is a programming language that allows web developers to create dynamic content that interacts with databases. PHP is a server side scripting language that is embedded in HTML. It is used to manage dynamic content, databases, session tracking, even build entire ecommerce sites. It is integrated with a number of popular databases, including MySQL, PostgreSQL, Oracle, Sybase, Informix, and Microsoft SQL Server. Laravel is a web application framework with expressive, elegant syntax. We've already laid the foundation —

freeing you to create without sweating the small things. Laravel is a web application framework with expressive, elegant syntax. We believe development must be an enjoyable, creative experience to be truly fulfilling. Laravel attempts to take the pain out of development by easing common tasks used in the majority of web projects, such as authentication, routing, sessions, and caching. MySQL Language is an

opensource relational database management system (RDBMS). Its name is a combination of "My", the name of cofounder Michael Wideness's daughter, and "SQL", the abbreviation for Structured Query Language. MySQL is free and open-source software under the terms of the GNU General Public License, and is also available under a variety of proprietary licenses.

MySQL is a component of the LAMP web application software stack (and others), which is an acronym for Linux, Apache, MySQL, Perl/PHP/Python. MySQL is used by many database-driven web applications, including Drupal, Joomla, phpBB, and WordPress. MySQL is also used by many popular websites, including Facebook, Flickr, Media Wiki, Twitter, and YouTube. XAMPP stands for Cross-Platform (X), Apache (A), MySQL (M), PHP (P) and Perl (P). is a free and open-source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages. Since most actual web server deployments use the same components as XAMPP, it makes transitioning from a local test server to a live server possible.

4.1.4 Mobile Content using Xamarin.

It is a popular cross-platform development framework that allows developers to create mobile applications for iOS, Android, and Windows using a shared codebase. Here are some benefits of using Xamarin for designing mobile apps like as Code Reusability, it is one of the significant advantages of Xamarin is code sharing. Developers can write the core application logic in C# and reuse a significant portion of code across different platforms. This reduces development time and effort compared to building separate native apps for each platform.

Native Performance, Xamarin enables developers to build native-like applications that deliver high performance. Xamarin apps utilize native UI controls, ensuring a native look and feel on each platform. The use of native APIs also allows access to platform-specific features, resulting in better performance and user experience.

Time and Cost Efficiency, Building mobile apps with Xamarin reduces development time and cost. Since the codebase is shared, developers can write and maintain a single codebase instead of separate ones for iOS and Android. This streamlines the development process and reduces the need for platform-specific expertise.

Rapid Development, Xamarin provides a rich set of pre-built UI controls and components, which accelerates app development. These controls can be easily customized and integrated into the app's design. Xamarin.Forms, a UI toolkit, further simplifies the UI development by enabling the creation of a single UI codebase that can be shared across platforms.

Access to Native APIs, Xamarin allows direct access to native APIs, libraries, and device-specific functionalities. This means developers can leverage the full capabilities of each platform and incorporate platform-specific features seamlessly into their apps. This access ensures a high level of device integration and functionality.

Simplified Maintenance, With Xamarin, maintenance becomes easier as changes and updates can be made to the shared codebase, and the modifications will reflect across all platforms. This reduces the effort required for bug fixing, feature enhancements, and updates.

These benefits make Xamarin a compelling framework for designing mobile apps, offering efficiency, performance, code reusability, and access to native features across multiple platforms.

4.2 UI/UX design

4.2.1 Logo of application

This is splash screen for our nearest service application.

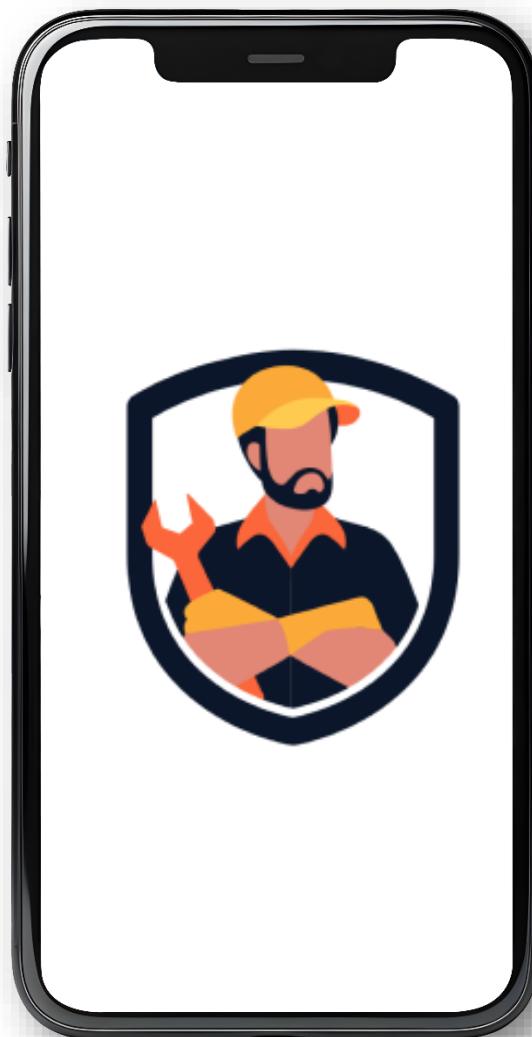


Figure 4-1 Logo of application

4.2.2 Intro of application and Welcome Page

This is an introductory page for the application and welcome page for the application.



Figure 4-2 Intro and welcome page for App

4.2.3 Select Your Role

Here you can select your role in the App (user or provider)



Figure 4-3 Select Role

4.2.4 Description Page

User see short description about the app and link for more information



Figure 4-4 Description Page

4.2.5 Information Page

Here user can see information about the company



Figure 4-5 Information Page

4.2.6 Finger checking

Option to log in using your fingerprint, and you can ignore this step.



Figure 4-6 Finger checking

4.2.7 Login page

In this page, you will login in the application with your account First write your email and then password, or you can login by Google.

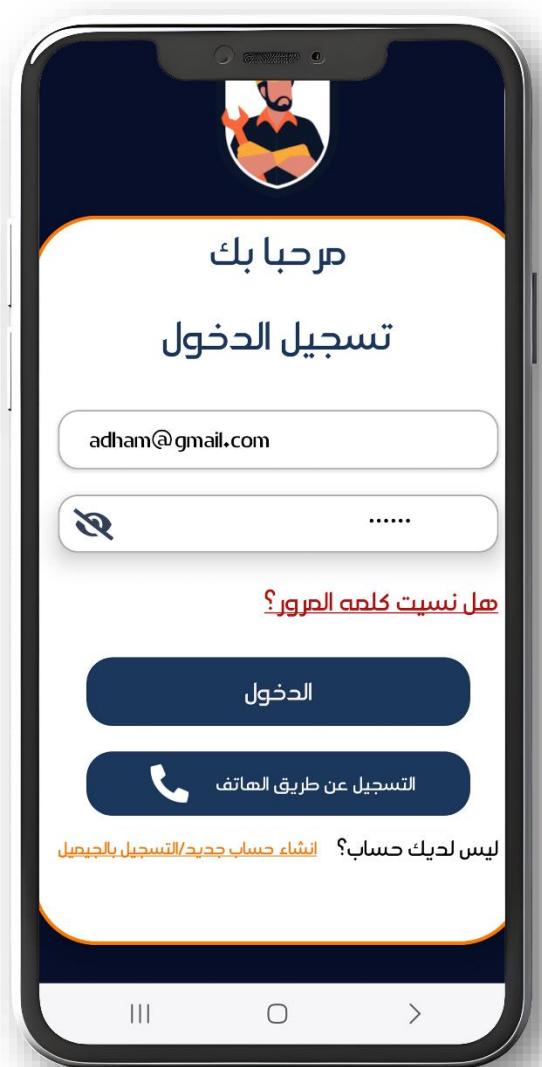


Figure 4-7 Login

4.2.8 Sign in using phone

User can also sign in directly using his phone number

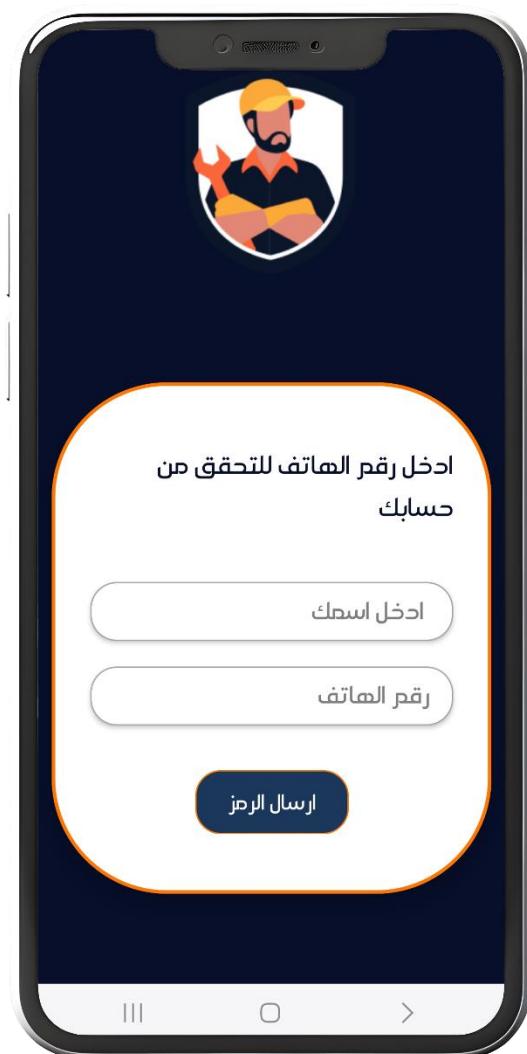


Figure 4-8 Sign in using phone

4.2.9 Password Forget

If the user forget his his password can sign into the App

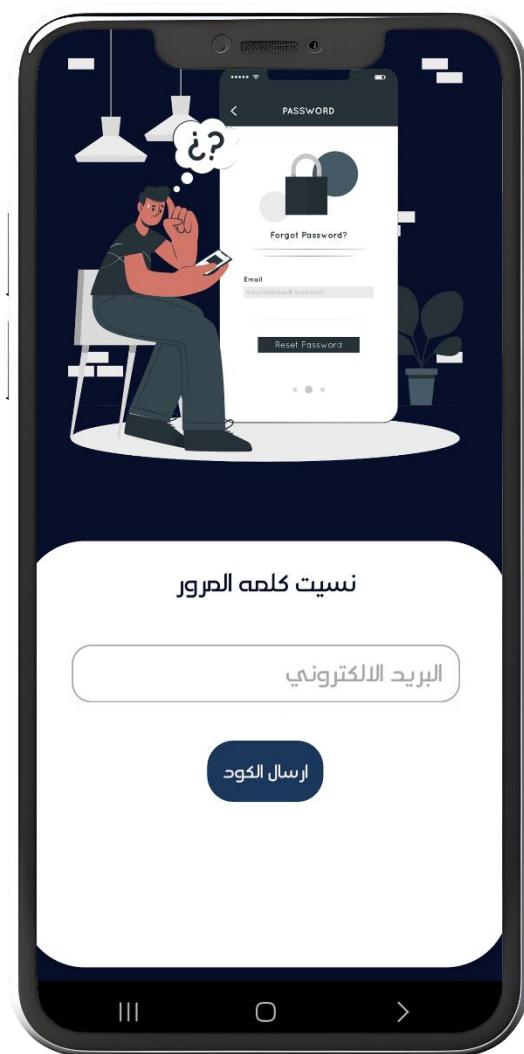


Figure 4-9 Forget Password

4.2.10 Sign up page

You can create an account on the application by entering your name, email, and password, or through your contact on Google.

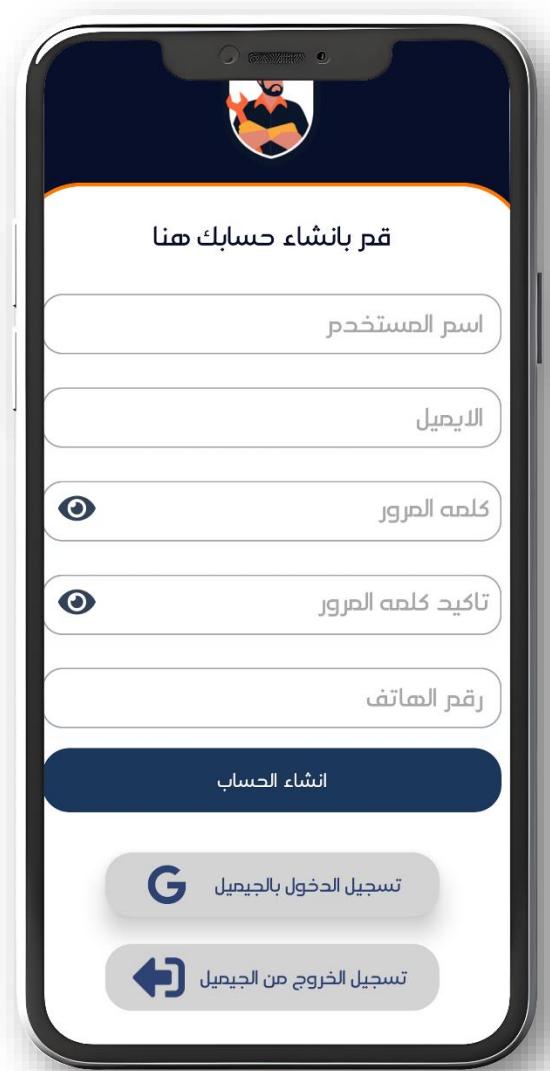


Figure 4-10 Sign up page

4.2.11 Services Page

Services page to choose the service required by the user or by searching for it from the search bar.



Figure 4-11 Services Page

4.2.12 Map | Choose your provider.

After choosing the service user gets the near service providers on map .

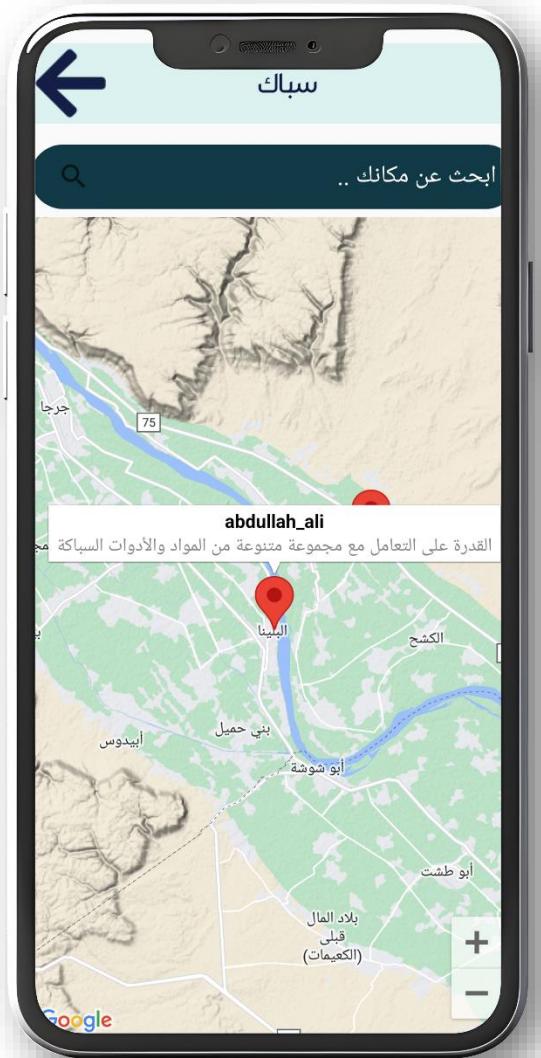


Figure 4-12 Map to choose any provider.

4.2.13 Information of the provider

After click on the suitable provider window will open to see provider details including his(rate , average salary , location)

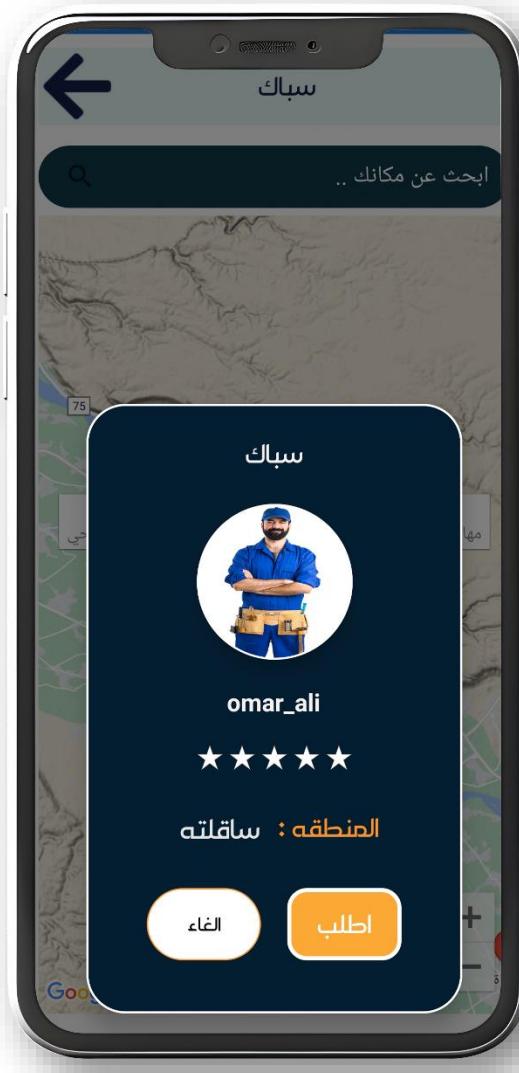


Figure 4-13 Info of provider

4.2.14 Provider request

After the service provider requests the user enter his phone number or email, he can upload a picture of the problem and then confirm the request.

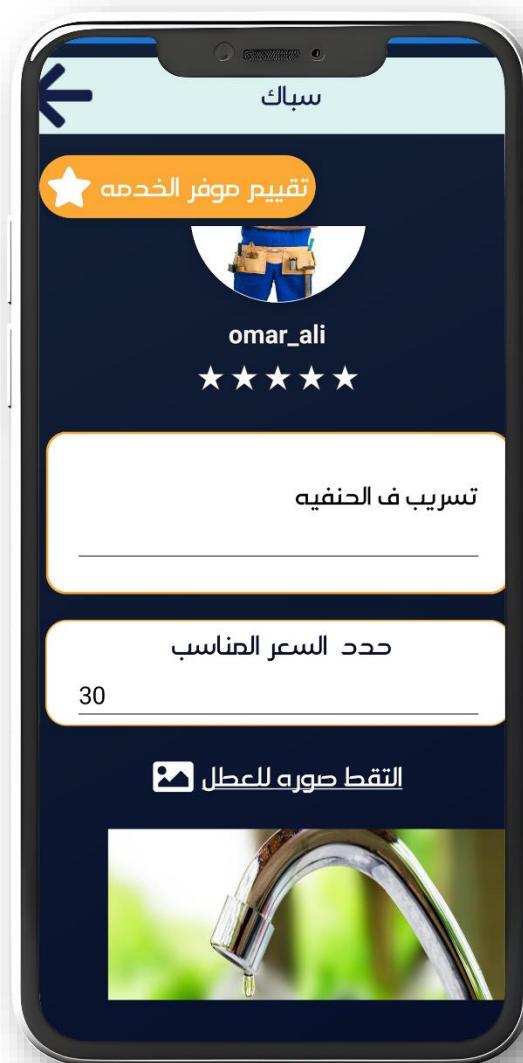


Figure 4-14 Provider Request

4.2.15 Successful Page

When your request saved in database this page will appear.



Figure 4-15 Successful Page

4.2.16 Profile Page

user can change his profile picture and the Gmail and go to requests page.

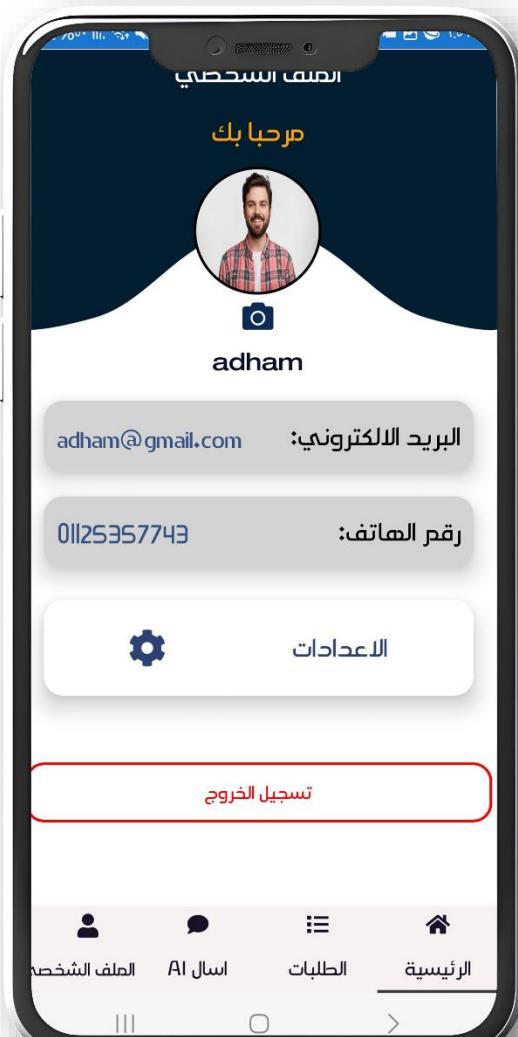


Figure 4-16 Profile Page

4.2.17 Settings Page

In setting page, user will write his phone number , password and location, then save it.



Figure 4-17 Setting Page

4.2.18 Change Password

User can change his password in this page



Figure 4-18 Change Password

4.2.19 Ask Ai Page

Page to Ask the Ai about Your problem



Figure 4-19 Ask Ai Page

4.2.20 Requests Page

page to display the orders that have been placed



Figure 4-20 Requests page

4.2.21 Provider Sign in

Here Provider can sign in to the App

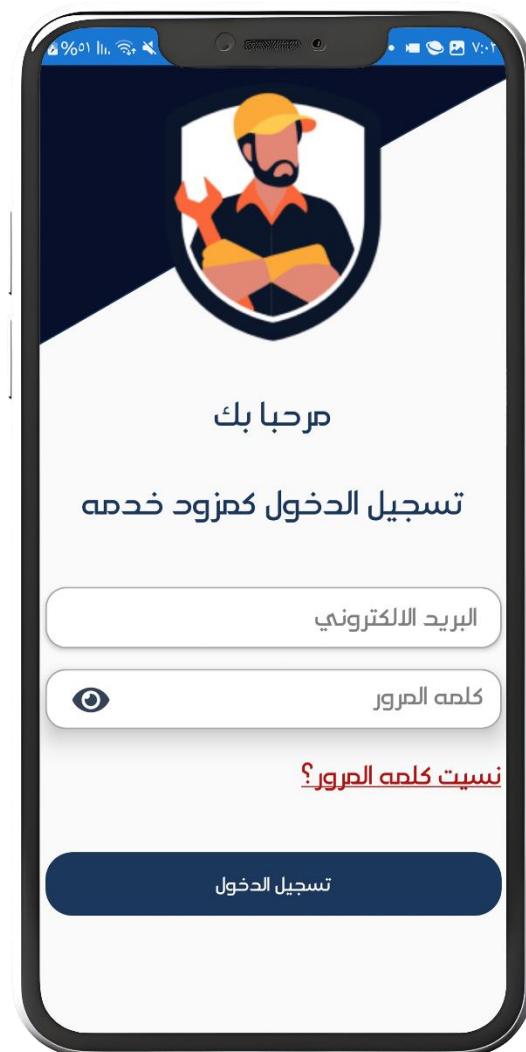


Figure 4-21 Provider sign in page

4.2.22 Provider Requests

Here provider can see all of his requests and interact with it (update , reject , accept)



Figure 4-22 Provider Requests page

4.2.23 Provider Profile

In this page provider can see his profile information



Figure 4-23 Provider Profile page

4.3 Website Dashboard Ui:

4.3.1 Admin Sign in

This page admin only will sign in

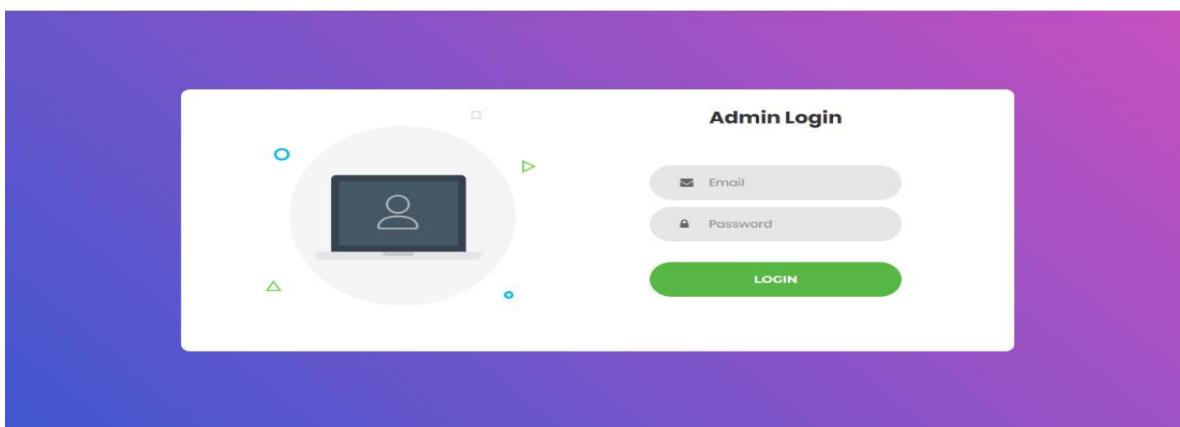


Figure 4-24 Admin Login

4.3.2 Main Page:

This page includes system analysis

A screenshot of the main dashboard. The left sidebar shows navigation links: Dashboard, Providers, Services, Users, Add Service, and Add Providers. The main area has a header "Dashboard" and a "Welcome Admin" message: "Hello Admin! 🌟 Welcome to the Dashboard. Your insights and expertise are invaluable. Let's dive into the data and make informed decisions together. If you need any assistance, feel free to reach out!" Below this is a section titled "Here are some analyzes and statistics:" with five cards: Total Orders (20), Total Users (15), Total Services (10), Total Providers (20), and Total Sales (1400L.E.).

Figure 4-25 Main Page

4.3.3 Show Services:

This page includes all services exists in the system



ID	Name	Name Ar	Image
1	Electrician	كهربائي	
2	Plumber	سباك	
3	Carpenter	نجار	
4	painter	نقاش	

Figure 4-26 Show service

4.3.4 Show Providers:

this page includes all service-providers in the system



ID	Name	Service Name	Service Name Ar	Area	Remove
1	ali_maher	Electrician	كهربائي	أحمديم	Remove
2	oraby_sayed	Electrician	كهربائي	طما	Remove
3	samer_ahmed	Electrician	كهربائي	مركز سوهاج	Remove
4	islam_mahmoud	Electrician	كهربائي	المراعي	Remove
5	amer_ahmed	Electrician	كهربائي	طهطا	Remove

Num of Providers in the Electrician job = 5

ID	Name	Service Name	Name Ar	Area	Remove
8	omar_ali	Plumber	سباك	ساقله	Remove
10	ahmed_ali	Plumber	سباك	دار السلام	Remove
12	mahmoud_alaa	Plumber	سباك	المنشأ	Remove
13	abdullah_ali	Plumber	سباك	البلينا	Remove

Num of Providers in the Plumber job = 4

Figure 4-27 Show Providers Page

4.3.5 Show Users:

this page includes all users in the system

Id	Phone	Name	Email
3068	01125357743	sara	sara@gmail.com
3069	01585884852	nd.mostafa5	nd.mostafa5@gmail.com
3070	01125357743	Nada	nada555@gmail.com
3071	01125357743	adham	adham@gmail.com

Figure 4-28 Show Users Page

4.3.6 Add Service:

In this page admin can add new service to system

Figure 4-29 Add Service Page

4.3.7 Add Provider:

In this page admin can add or reject the new providers

ID	Name	Address	Email	Mobile Phone	Service	Personal Photo	Operation
1	احمد حلف	اخيم	kakd@gmail.com	01010039239932	Electricain		<button>confirm</button> <button>reject</button>
2	محمد ابراهيم	طهطا	d87ty@gmail.com	01010039666	Electricain		<button>confirm</button> <button>reject</button>
3	يعقوب اشرف	المراعنة	yehia33@gmail.com	01197839666	Electricain		<button>confirm</button> <button>reject</button>
4	يوسف حاتم	دار السلام	youssef@gmail.com	0159039666	Electricain		<button>confirm</button> <button>reject</button>
5	محمود ملاحت	الصوامعة	medhat12@gmail.com	012009666	Electricain		<button>confirm</button> <button>reject</button>

Figure 4-30 Add Providers Page

Chapter 5

System

implementation

5 Chapter 5 || System implementation

5.1 Mobile App Code (Xamarin):

5.1.1 Page 1

```

<?xml version="1.0" encoding="utf-8" ?>
<controls:Custompage xmlns="http://xamarin.com/schemas/2014/forms"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    xmlns:controls="clr-namespace:otp.Controls"
    xmlns:xct="http://xamarin.com/schemas/2020/toolkit"
    x:Class="otp.Pages.IntroPage">

    <ContentPage.Content>
        <Grid>
            <Image Grid.Row="0" Margin="0,0,0,0" VerticalOptions="StartAndExpand"
                HorizontalOptions="FillAndExpand" Opacity="1" Source="pr.png"
            ></Image>
            <BoxView>
                <BoxView.Background>
                    <LinearGradientBrush StartPoint="1,0" EndPoint="1,1">
                        <GradientStop Offset="0" Color="Transparent"></GradientStop>
                        <GradientStop Offset="0.9" Color="#192655"></GradientStop>
                        <GradientStop Offset="1" Color="#2e4159"></GradientStop>
                    </LinearGradientBrush>
                </BoxView.Background>
            </BoxView>
            <Grid RowDefinitions="auto,auto,auto,auto,auto" RowSpacing="10"
Margin="30,0,30,30"
                HeightRequest="300" VerticalOptions="EndAndExpand">

                <CarouselView IndicatorView="indicator" FlowDirection="RightToLeft"
x:Name="caView" ItemsSource="{Binding ListSliders }" HorizontalOptions="EndAndExpand">
                    <CarouselView.ItemTemplate>
                        <DataTemplate >
                            <StackLayout Spacing="20" >
                                <Label Text="{Binding Description1}" FontFamily="DG"
TextColor="#E8AA42"
                                    FontAttributes="Bold" FontSize="20"
HorizontalOptions="Start"></Label>
                                <Label Text="{Binding Description2}" FontFamily="DG"
TextColor="White"
                                    Grid.Row="1"
                                    FontAttributes="None" FontSize="18"
HorizontalOptions="Start"></Label>
                            </StackLayout>
                        </DataTemplate>
                    </CarouselView.ItemTemplate>
                </CarouselView>
            </Grid>
        </Grid>
    </ContentPage.Content>
</controls:Custompage>

```

```

        </StackLayout>
    </DataTemplate>
</CarouselView.ItemTemplate>
</CarouselView>

<IndicatorView x:Name="indicator" Grid.Row="1"
    Padding="0,0,0,-10"
    IndicatorsShape="Square"
    HorizontalOptions="Center"
SelectedIndicatorColor="Transparent"
    IndicatorColor="Transparent">
<IndicatorView.IndicatorTemplate>
    <DataTemplate>
        <ImageButton
        >
            <ImageButton.Source>
                <FontImageSource Glyph="" Color="#E8AA42"
FontFamily="FontIconSolid"></FontImageSource>
                </ImageButton.Source>
            </ImageButton>
        </DataTemplate>
    </IndicatorView.IndicatorTemplate>
</IndicatorView>

<StackLayout Orientation="Horizontal" Padding="0,0,0,30" Spacing="20"
VerticalOptions="EndAndExpand" HorizontalOptions="CenterAndExpand">
    <Button Command="{Binding GoRegister}"
        x:Name="btnIntro"
        xct:TouchEffect.PressedBackgroundColor="White"
        xct:TouchEffect.PressedScale=".9"
        xct:TouchEffect.PressedOpacity=".6"
        xct:TouchEffect.PressedRotation="10"
        xct:TouchEffect.PressedAnimationDuration="1000"
        Text="ابداً الان" FontSize="16" HeightRequest="50 "
        BorderColor="White" BorderWidth="2" FontFamily="DG"
        TextColor="White"
        CornerRadius="25"
        BackgroundColor="#2e4159 " WidthRequest="154"
        FontAttributes="Bold" ></Button>

    </StackLayout>
</Grid>
</Grid>
</ContentPage.Content>
</controls:Custompage>

```

```

using FreshMvvm;
using System;

```

```

using System.Collections.Generic;
using System.Collections.ObjectModel;
using System.Text;
using otp.Models;
using Xamarin.Forms;

namespace otp.ViewModels
{
    public class IntroViewModel : FreshBasePageModel
    {

        public Command GoRgister { get; set; }

        public IntroViewModel()
        {
            ListSliders = LoadItems();

            GoRgister = new Command(OnGoRgister);

        }

        ObservableCollection<IntroClass> _ListSliders;
        public ObservableCollection<IntroClass> ListSliders
        {
            get
            {
                return _ListSliders;
            }
            set
            {
                _ListSliders = value;
                if (_ListSliders != null)
                    RaisePropertyChanged("ListSliders");
            }
        }

        public ObservableCollection<IntroClass> LoadItems()
        {
            ObservableCollection<IntroClass> lstItems = new
            ObservableCollection<IntroClass>();

            IntroClass oItems = new IntroClass();
            oItems.Description1 = "ابحث عن أقرب الخدمات إليك هنا";
            oItems.Description2 = "استكشف الخدمات القريبة المتاحة في المنطقة التي تتوارد فيها";
            lstItems.Add(oItems);

            oItems = new IntroClass();
            oItems.Description1 = "مرحباً بك مرة أخرى";
            lstItems.Add(oItems);

            return lstItems;
        }
    }
}

```

```

        async void OnGoRgister()
    {
        await CoreMethods.PushPageModel<ChoseViewModel>(null,true);

    }
}

using System.Threading.Tasks;
using Xamarin.Forms;
using Xamarin.Forms.Xaml;

namespace otp.Pages
{
    [XamlCompilation(XamlCompilationOptions.Compile)]
    public partial class IntroPage : Controls.Custompage
    {
        public IntroPage()
        {
            InitializeComponent();

            AnimationCarsoual();
        }

        System.Timers.Timer timer;
        private async Task StartPulseAnimation(Button button)
        {
            uint duration = 1000;
            double scale = 1.2;

            int pulseCount = 0;

            while (pulseCount < 4)
            {
                await button.ScaleTo(scale, duration / 2);
                await button.ScaleTo(1, duration / 2);
                pulseCount++;
            }
        }

        protected override async void OnAppearing()
        {
            base.OnAppearing();
            await StartPulseAnimation(btnIntro);
        }

        private void AnimationCarsoual()
        {
            timer = new System.Timers.Timer(5000) { AutoReset = true, Enabled = true };
            timer.Elapsed += (s, e) =>
            {
                Device.BeginInvokeOnMainThread(() =>
                {
                    if (caView.Position == 1)
                    {
                        caView.Position = 0;
                        return;
                    }
                    caView.Position += 1;
                });
            };
        }
    }
}

```

```

        );
    }

}
-----  

using System;
using System.Collections.Generic;
using System.Text;

namespace otp.Models
{
    public class IntroClass
    {
        public string Description1 { get; set; }
        public string Description2 { get; set; }

    }
}

```

5.1.1.1 Page 2

```

<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    xmlns:yummy="clr-
namespace:Xamarin.Forms.PancakeView;assembly=Xamarin.Forms.PancakeView"

    NavigationPage.HasNavigationBar="False"
    BackgroundColor="#070F2B"
    x:Class="otp.Pages.ChosePage">
<ContentPage.Content>
    <StackLayout>

        <Grid

            VerticalOptions="CenterAndExpand"
            RowSpacing="20"
        >

            <Grid.RowDefinitions>
                <RowDefinition Height="auto" />
                <RowDefinition Height="120" />
                <RowDefinition Height="150" />
                <RowDefinition Height="150" />
                <RowDefinition Height="auto" />
            </Grid.RowDefinitions>
            <Path Data="m559.96688,125.56515c-21.53924,0 -38.99997,10.96781 -
38.99997,24.49914c0,4.62315 2.06424,8.94371 5.61595,12.63227l-5.61595,10.31451115.63613,-
3.36103c6.51575,3.07082 14.58886,4.9151 23.36384,4.9151c21.53929,0 38.99999,-10.9696
38.99999,-24.50086s-17.4607,-24.49914 -38.99999,-24.49914zm2.73584,38.352621-5.4855,010,-
4.4077315.4855,010,4.40773zm0,-9.1146110,1.457981-5.4855,010,-1.79444c0,-5.41536
7.35742,-6.2734 7.35742,-10.11864c0,-1.75531 -1.87192,-3.10113 -4.32614,-3.10113c-

```

```

2.54085,0 -4.77192,1.56839 -4.77192,1.56839-3.12054,-3.25067c0,0 3.0759,-2.68987
8.38235,-2.68987c5.04235,0 9.72234,2.6151 9.72234,7.02111c0.00267,6.16663 -
7.75802,6.87513 -7.75802,10.90728l0,0z"
    Aspect="Uniform" Margin="20" Fill="#FF7700"
HorizontalOptions="CenterAndExpand" Grid.Row="1"/>
    <Label

        Margin="0,0,0,0"
        Grid.Row="0"
        HorizontalOptions="CenterAndExpand"

        Text="حدد دورك"
        FontSize="40"
        fontFamily="DG"
        TextColor="White" />
<yummy:PancakeView BackgroundColor="White" Grid.Row="2"
BackgroundGradientEndPoint="1,1" BackgroundGradientStartPoint="0,0" Padding="20"
CornerRadius="30" HorizontalOptions="CenterAndExpand">
    <yummy:PancakeView.Border>
        <yummy:Border Color="#FF7700" Thickness="5"></yummy:Border>

    </yummy:PancakeView.Border>
    <yummy:PancakeView.BackgroundGradientStops>
        <yummy:GradientStopCollection>
            <yummy:GradientStop Color="White"
Offset="0"></yummy:GradientStop>
            <yummy:GradientStop Color="White"
Offset="1"></yummy:GradientStop>
        </yummy:GradientStopCollection>
    </yummy:PancakeView.BackgroundGradientStops>
    <StackLayout Orientation="Horizontal" >

        <Image Source="provider.jpeg" HeightRequest="100">

    </Image>
        <Label Text="التسجيل كمقدم خدمة" FontSize="20" FontAttributes="Bold"
VerticalTextAlignment="Center" HorizontalTextAlignment="Center" fontFamily="DG"></Label>

    </StackLayout>
    <yummy:PancakeView.GestureRecognizers>
        <TapGestureRecognizer Command="{Binding
GoLoginProvider}"></TapGestureRecognizer>
    </yummy:PancakeView.GestureRecognizers>
</yummy:PancakeView>
<yummy:PancakeView Grid.Row="3" WidthRequest="300"
BackgroundColor="White" BackgroundGradientEndPoint="1,1"
BackgroundGradientStartPoint="0,0" Padding="20" CornerRadius="30"
HorizontalOptions="CenterAndExpand">
    <yummy:PancakeView.Border>
        <yummy:Border Color="#FF7700" Thickness="5"></yummy:Border>

    </yummy:PancakeView.Border>
    <yummy:PancakeView.BackgroundGradientStops>
        <yummy:GradientStopCollection>
            <yummy:GradientStop Color="White"
Offset="0"></yummy:GradientStop>

```

```

                <yummy:GradientStop Color="White"
Offset="1"></yummy:GradientStop>
                    </yummy:GradientStopCollection>
                </yummy:PancakeView.BackgroundGradientStops>
                <StackLayout Orientation="Horizontal"
HorizontalOptions="CenterAndExpand" Spacing="20" >

                    <Image >
                        <Image.Source>
                            <FontImageSource Glyph="" Color="#070F2B"
FontFamily="FontIconSolid"></FontImageSource>
                        </Image.Source>
                    </Image>
                    <Label Text="التسجيل مستخدم" FontSize="20" FontAttributes="Bold"
VerticalTextAlignment="Center" HorizontalTextAlignment="Center" FontFamily="DG"></Label>
                </StackLayout>
                <yummy:PancakeView.GestureRecognizers>
                    <TapGestureRecognizer Command="{Binding
GoLogin}"></TapGestureRecognizer>
                    </yummy:PancakeView.GestureRecognizers>
                </yummy:PancakeView>
            </Grid>
        </StackLayout>
    </ContentPage.Content>
</ContentPage>

```

```

using FreshMvvm;
using System;
using System.Collections.Generic;
using System.Text;
using Xamarin.Forms;

namespace otp.ViewModels
{
    public class ChoseViewModel: FreshBasePageModel
    {

        public Command GoLogin { get; set; }
        public Command GoLoginProvider { get; set; }

        public ChoseViewModel()
        {
            GoLogin = new Command(OnGoLogin);
            GoLoginProvider = new Command(OnGoLoginProvider);
        }

        async void OnGoLogin()
        {
            await CoreMethods.PushPageModel<LoginViewModel>(null, true);
        }
        async void OnGoLoginProvider()
        {
            await CoreMethods.PushPageModel<IntroductionViewModel>(null, modal: true);
        }
    }
}

```

```

    }
}
```

5.1.2 Page 2

```

<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    xmlns:xct="http://xamarin.com/schemas/2020/toolkit"
    NavigationPage.HasNavigationBar="False"
    x:Class="otp.Pages.IntroductionPage">
    <Grid>
        <xct:MediaElement x:Name="mediaElement" Source="ms-appx:///Pipeline.mp4"
AutoPlay="True"
            Aspect="AspectFill" VerticalOptions="FillAndExpand"
            BackgroundColor="Black" HorizontalOptions="FillAndExpand"
ShowsPlaybackControls="False"></xct:MediaElement>
        <BoxView>
            <BoxView.Background>
                <LinearGradientBrush StartPoint="1,0" EndPoint="1,1">
                    <GradientStop Color="Transparent" Offset="0"/>
                    <GradientStop Color="#2E4159" Offset="0.9"/>
                    <GradientStop Color="#2E4159" Offset="1"/>
                </LinearGradientBrush>
            </BoxView.Background>
        </BoxView>
        <Grid RowDefinitions="Auto, Auto, Auto" HeightRequest="280" Margin="30,30,30,100"
RowSpacing="40" VerticalOptions="End">
            <Label>
                <Label.FormattedText>
                    <FormattedString>
                        <FormattedString.Spans>
                            <Span Text="{Binding Description1}" FontFamily="DG"
TextColor="White"
FontAttributes="Bold" FontSize="18" />
                            <Span Text="زور موقعاً" TextColor="#E8AA42" FontFamily="DG"
TextDecorations="Underline"
FontSize="20" >
                                <Span.GestureRecognizers>
                                    <TapGestureRecognizer Command="{Binding
OpenLinkCommand}" />
                                </Span.GestureRecognizers>
                            </Span>
                            <Span Text="{Binding Description2}" FontFamily="DG"
TextColor="White"
FontAttributes="Bold" FontSize="18" />
                        </FormattedString.Spans>
                    </FormattedString>
                </Label.FormattedText>
            </Label>
            <Button Grid.Row="1"

```

```

        VerticalOptions="EndAndExpand"
HorizontalOptions="CenterAndExpand"
        Text="تسجيل الدخول" Command="{Binding GoRgister}" HeightRequest="50 "
        BackgroundColor="#E8AA42" TextColor="White" FontSize="16"
        BorderColor="White" BorderWidth="2" FontFamily="DG"
        WidthRequest="154" CornerRadius="25"
        FontAttributes="Bold" ></Button>

    </Grid>
</Grid>
</ContentPage>

```

```

using FreshMvvm;
using System;
using System.Collections.Generic;
using System.Text;
using System.Windows.Input;
using Xamarin.Essentials;
using Xamarin.Forms;

namespace otp.ViewModels
{
    public class IntroductionViewModel : FreshBasePageModel
    {
        public ICommand OpenLinkCommand { get; private set; }
        public Command GoRgister { get; set; }
        private string description1;
        public string Description1
        {
            get { return description1; }
            set
            {
                description1 = value;
                RaisePropertyChanged(nameof(Description1));
            }
        }
        private string description2;
        public string Description2
        {
            get { return description2; }
            set
            {
                description2 = value;
                RaisePropertyChanged(nameof(Description2));
            }
        }

        public IntroductionViewModel()
        {
            Description1 = "\n . تعد شركتنا الرائدة في تقديم فرص عمل للفنيين ذوي الخبرة " +
            + "\n: إذا كنت تبحث عن فرصة للانضمام او التعرف اكثر علينا";
            Description2 = "\n: إذا كنت عضواً في شركتنا، يُرجى ";
            OpenLinkCommand = new Command(OpenLink);
            GoRgister = new Command(OnGoRgister);
        }
    }
}

```

```

    private void OpenLink()
    {
        var url = "https://6633cac352829c12f1cab22b--admirable-halva-6e7f2f.netlify.app/";

        Launcher.OpenAsync(new Uri(url));
    }

    async void OnGoRegister()
    {
        await CoreMethods.PushPageModel<LoginProviderViewModel>(null, true);
    }
}

```

5.1.3 Page 3

```

<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    xmlns:Res="clr-namespace:otp.Resources"
    xmlns:controls="clr-namespace:otp.Controls"
    xmlns:xct="http://xamarin.com/schemas/2020/toolkit"

    NavigationPage.HasNavigationBar="False"
    xmlns:yummy="clr-
namespace:Xamarin.Forms.PancakeView;assembly=Xamarin.Forms.PancakeView"
    xmlns:inputLayout="clr-
namespace:Syncfusion.XForms.TextInputLayout;assembly=Syncfusion.Core.XForms"
    x:Class="otp.Pages.LoginProviderPage">
    <ContentPage.Resources>
        <ResourceDictionary>
            <Style TargetType="Entry" x:Key="InvalidS">
                <Setter Property="TextColor" Value="Red"></Setter>
            </Style>

            <Style TargetType="Entry" x:Key="ValidS">
                <Setter Property="TextColor" Value="Black"></Setter>
            </Style>
        </ResourceDictionary>
    </ContentPage.Resources>
    <ContentPage.Content>
        <StackLayout>
            <Grid HorizontalOptions="FillAndExpand" VerticalOptions="FillAndExpand"
RowSpacing="20" Padding="20,0,20,0">
                <Grid.RowDefinitions>
                    <RowDefinition Height="Auto"></RowDefinition>
                    <RowDefinition Height="Auto"></RowDefinition>
                    <RowDefinition Height="Auto"></RowDefinition>

```

```

<RowDefinition Height="Auto"></RowDefinition>
<RowDefinition Height="Auto"></RowDefinition>

</Grid.RowDefinitions>

<Path Grid.Row="0" Margin="-20,0"
Data="m132.34,154.23002l209.00002,0l0,0.66666c-66.50001,28 -156.50001,72.87853 -
209.00002,96.76573l0,-97.43239z"
Stroke="#1C375C" Fill="#070F2B" HeightRequest="220" Aspect="Fill"
></Path>
<Image Grid.Row="0" Margin="0,0,0,0" Source="Logo.png" Aspect="AspectFit"
HeightRequest="180" WidthRequest="80"></Image>
<Label Grid.Row="1" Text="مرحبا بك" FontFamily="DG" FontSize="24"
FontAttributes="Bold" TextColor="#1C375C" HorizontalOptions="CenterAndExpand"></Label>
<Label Grid.Row="2" Text="تسجيل الدخول كمزود خدمة" TextColor="#1C375C"
FontFamily="DG" FontSize="24" HorizontalOptions="CenterAndExpand"></Label>

<StackLayout Grid.Row="3" VerticalOptions="CenterAndExpand">
<Frame CornerRadius="15" WidthRequest="350"
Margin="0,20,0,0"
HorizontalOptions="CenterAndExpand"
BorderColor="DarkGray"
HasShadow="True"
HeightRequest="6 " Grid.Row="2" >

<Entry FlowDirection="RightToLeft" x:Name="txtEmail"
Text="{Binding txtemail}" Keyboard="Email"
TextColor="Black"
ClearButtonVisibility="WhileEditing"
PlaceholderColor="Gray"
Placeholder="البريد الإلكتروني"
Margin="0,-30,-5,-30"
FontFamily="DG"
xct:TouchEffect.HoveredBackgroundColor="White"
xct:TouchEffect.NormalBackgroundColor="Transparent"
VerticalOptions="FillAndExpand">

<Entry.Behaviors >
<xct:EmailValidationBehavior

DecorationFlags="Trim"
InvalidStyle="{StaticResource InvalidS}"
ValidStyle="{StaticResource ValidS}"
Flags="ValidateOnValueChanging"
IsValid="True"

>
</xct:EmailValidationBehavior>
</Entry.Behaviors>

</Entry>

```

```

        </Frame>

        <yummy:PancakeView CornerRadius="15" Grid.Row="4"
BackgroundColor="White"
WidthRequest="350"
HeightRequest="45"

Margin="0,10,0,0"
HorizontalOptions="FillAndExpand"
>
<yummy:PancakeView.Border>
<yummy:Border Color="DarkGray"
Thickness="2"></yummy:Border>
</yummy:PancakeView.Border>
<yummy:PancakeView.Shadow>
<yummy:DropShadow></yummy:DropShadow>
</yummy:PancakeView.Shadow>
<StackLayout Grid.Row="3" Orientation="Horizontal"
Margin="20,5,10,10"    >

        <Image   x:Name="imgpass" HeightRequest="25"
VerticalOptions="EndAndExpand" HorizontalOptions="StartAndExpand"    >
<Image.Source >
<FontImageSource Glyph="" Color="#334257"
FontFamily="FontIconSolid"></FontImageSource>
</Image.Source>
<Image.GestureRecognizers>
<TapGestureRecognizer
Tapped="TapGestureRecognizer_Tapped"></TapGestureRecognizer>
</Image.GestureRecognizers>
</Image>

        <Entry   Text="{Binding txtPass}"
Margin="0,-30,0,-30"
TextColor="Black"
HeightRequest="12"
Grid.ColumnSpan="2"

PlaceholderColor="Gray"
FontFamily="DG"

xct:TouchEffect.HoveredBackgroundColor="White"

xct:TouchEffect.NormalBackgroundColor="Transparent"
xct:ShadowEffect.Radius="23"

x:Name="PassProvider" Placeholder="كلمة المرور"
ClearButtonVisibility="WhileEditing" ></Entry>
</StackLayout>
</yummy:PancakeView>

```

```

<Label Grid.Row="4" TextColor="#B31312" FontAttributes="Bold"
       TextDecorations="Underline" FontFamily="DG" Text="نسیت کلمہ المرور؟"
       Padding="0,10,0,0" VerticalOptions="EndAndExpand"
       HorizontalOptions="EndAndExpand">
    <Label.GestureRecognizers>
        <TapGestureRecognizer Command="{Binding GoToForget}"></TapGestureRecognizer>
    </Label.GestureRecognizers>
    </Label>
</StackLayout>
<StackLayout Grid.Row="4" VerticalOptions="EndAndExpand"
Padding="0,20,0,0">
    <Button Text="تسجيل الدخول" Command="{Binding NavigateToHome}"
            xct:TouchEffect.PressedScale=".9"
            xct:TouchEffect.PressedOpacity=".7"

            xct:TouchEffect.PressedAnimationDuration="100"
            FontFamily="DG"
            BackgroundColor="#1C375C" TextColor="White" CornerRadius="20">
        <VisualStateManager.VisualStateGroups>
            <VisualStateGroup x:Name="CommonState">
                <VisualState x:Name="Normal">
                    <VisualState.Setters>
                        <Setter Property="Scale" Value="1">

                        </Setter>
                    </VisualState.Setters>

                </VisualState>
                <VisualState x:Name="Pressed">
                    <VisualState.Setters>
                        <Setter Property="Scale" Value=".9">

                        </Setter>
                    </VisualState.Setters>

                </VisualState>
            </VisualStateGroup>
        </VisualStateManager.VisualStateGroups>
    </Button>
</StackLayout>

</Grid>
</StackLayout>
</ContentPage.Content>
</ContentPage>

```

```

using Rg.Plugins.Popup.Extensions;
using Syncfusion.XForms.TextInputLayout;
using System;
using System.Collections.Generic;

```

```

using System.Linq;
using System.Text;
using System.Threading.Tasks;
using Xamarin.Forms;
using Xamarin.Forms.Xaml;

namespace otp.Pages
{
    [XamlCompilation(XamlCompilationOptions.Compile)]
    public partial class LoginProviderPage : ContentPage
    {
        public LoginProviderPage()
        {
            InitializeComponent();
        }

        public bool isPressed = true;

        private void TapGestureRecognizer_Tapped(object sender, EventArgs e)
        {
            if (isPressed == false)
            {
                imgpass.Source = new FontImageSource
                {
                    Glyph = "☒",
                    FontFamily = "FontIconSolid",
                    Color = Color.FromHex("#334257")
                };
                PassProvider.IsPassword = true;
                isPressed = true;
            }
            else
            {
                imgpass.Source = new FontImageSource
                {
                    Glyph = "☒",
                    FontFamily = "FontIconSolid",
                    Color = Color.FromHex("#334257")
                };
                PassProvider.IsPassword = false;
                isPressed = false;
            }
        }
    }
}

```

```

using Acr.UserDialogs;
using FreshMvvm;
using Newtonsoft.Json;
using otp.Models;
using Rg.Plugins.Popup.Extensions;
using System;
using System.Collections.Generic;

```

```

using System.Collections.ObjectModel;
using System.Linq;
using System.Text;
using Xamarin.Forms;

namespace otp.ViewModels
{
    public class LoginProviderViewModel : FreshBasePageModel
    {
        public LoginProviderViewModel()
        {
            NavigateToHome = new Command(OnNavigateToHome);
            auth = DependencyService.Get<IAuthenticationService>();
            GoToForget = new Command(OnGoToForget);
        }

        string _txtemail;
        public string txtemail
        {
            get
            {
                return _txtemail;
            }
            set
            {
                _txtemail = value;
                RaisePropertyChanged("txtemail");
            }
        }

        string _txtPass;
        public string txtPass
        {
            get
            {
                return _txtPass;
            }
            set
            {
                _txtPass = value;
                RaisePropertyChanged("txtPass");
            }
        }

        private INavigation _navigation;
        public ProviderModel CurrentProvider { get; set; }

        public Command NavigateToHome { get; set; }
        public Command GoToForget { get; set; }
        public Command GoRegistet { get; set; }
        IAuthenticationService auth;
        Provider oProvider = new Provider();

        public async void OnNavigateToHome(object obj)
    }
}

```

```

{
    var pop2 = new Helpers.MsgxPass();
    if (string.IsNullOrEmpty(txtPass) || string.IsNullOrEmpty(txtemail))
    {
        await App.Current.MainPage.Navigation.PushPopupAsync(pop2, true);
        return;
    }
    try
    {

        Helpers.Settings.ProviderEmail = txtemail;
        bool token = await auth.SignIn(txtemail, txtPass);
        if (!token)
        {

            await App.Current.MainPage.Navigation.PushPopupAsync(pop2, true);
            return;
        }
        else
        {
            UserDialogs.Instance.ShowLoading("يتم التحميل");
            string json = await Helpers.Utility.CallWebApi($""/api/Provider");
            List<Provider> providerList =
JsonConvert.DeserializeObject<List<Provider>>(json);

            string[] emailParts = txtemail.Split('@');
            string emailName = emailParts[0];

            Provider matchedProvider = providerList.FirstOrDefault(provider =>
provider.name == emailName);

            if (matchedProvider != null)
            {
                int providerId = matchedProvider.providerid;
            }

            await
CoreMethods.PushPageModel<RequsetProviderViewModel>(matchedProvider, true);

            UserDialogs.Instance.HideLoading();
        }
    }
    catch (Exception ex)
    {
        await App.Current.MainPage.Navigation.PushPopupAsync(pop2, true);
        return;
    }
}

async void OnGoToForget()
{
    await CoreMethods.PushPageModel<ForgotPasswordViewModel>(null, true);
}

```

```
        }  
    }  
}
```

5.1.4 Page 4

```
<?xml version="1.0" encoding="utf-8" ?>  
  
<control:Custompage xmlns="http://xamarin.com/schemas/2014/forms"  
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"  
    xmlns:control="clr-namespace:otp.Controls"  
    xmlns:Res="clr-namespace:otp.Resources"  
    xmlns:local="clr-namespace:otp.Helpers"  
    xmlns:xct="http://xamarin.com/schemas/2020/toolkit"  
    xmlns:syncfusion="clr-  
        namespace:Syncfusion.ListView.XForms;assembly=Syncfusion.SfListView.XFor  
        ms"  
    xmlns:yummy="clr-  
        namespace:Xamarin.Forms.PancakeView;assembly=Xamarin.Forms.PancakeView  
        "  
    xmlns:lottie="clr-namespace:Lottie.Forms;assembly=Lottie.Forms"  
    Title="{Binding Title}"  
    x:Name="homepage"  
    x:Class="otp.Pages.HomePage">
```

```
<xct:TabView
    IsSwipeEnabled="False"
    TabStripPlacement="Bottom"
    Style="{StaticResource TavView1}" >
    <xct:TabViewItem
        Style="{StaticResource TavView2}"
        Text="{Res:Translate Text=lblHome}" >
        <xct:TabViewItem.Icon>
            <FontImageSource Glyph="□" Color="{DynamicResource BgColor}"
                FontFamily="FontIconSolid"></FontImageSource>
        </xct:TabViewItem.Icon>
        <Grid >
            <Grid.RowDefinitions>
                <RowDefinition Height="auto"></RowDefinition>
                <RowDefinition Height="auto"></RowDefinition>
                <RowDefinition Height="*"></RowDefinition>
            </Grid.RowDefinitions>
            <control:ToolBar HasBackButton="False" HasCartButton="False"
                Title="{Binding Title,Source={x:Reference homepage} }"
                VerticalOptions="StartAndExpand"></control:ToolBar>
```

```
<SearchBar Placeholder="ابحث عن خدمتك"  
PlaceholderColor="Gray"  
Margin="10,0,10,10"  
xct:CornerRadiusEffect.CornerRadius="16"  
Grid.Row="1"  
Text="{Binding SearchTerm}"  
SearchCommand="{Binding SearchWithText}"  
CancelButtonColor="Black"  
BackgroundColor="White"  
TextColor="Black"  
TextTransform="Lowercase"  
HorizontalTextAlignment="Center"  
FontSize="Medium"  
FontAttributes="Bold" >  
<SearchBar.Behaviors>  
<xct:EventToCommandBehavior EventName="TextChanged"  
Command="{ Binding SearchWithText }">  
</xct:EventToCommandBehavior>  
</SearchBar.Behaviors>  
</SearchBar>
```

```
<syncfusion:SfListView x:Name="listView" Grid.Row="2"
    SelectionBackgroundColor="Transparent"
    ItemsSource="{Binding ListFilterServices}"
    ItemSpacing="10"
    ItemSize="180">
    <syncfusion:SfListView.LayoutManager>
        <syncfusion:GridLayout SpanCount="2" />
    </syncfusion:SfListView.LayoutManager>
    <syncfusion:SfListView.ItemTemplate>
        <DataTemplate>
            <ViewCell>
                <yummy:PancakeView Margin="5" Style="{StaticResource ServiceBox}"
                    xct:TouchEffect.PressedAnimationDuration="100"
                    xct:TouchEffect.PressedScale=".75"
                    xct:TouchEffect.PressedOpacity=".6"
                    xct:TouchEffect.PressedAnimationEasing="{x:Static Easing.BounceIn}">
                    <yummy:PancakeView.Border>
```

```
<yummy:Border Color="#18122B" Thickness="10"
></yummy:Border>

</yummy:PancakeView.Border>

<yummy:PancakeView.Shadow >

<yummy:DropShadow Color="White" Opacity="1"
></yummy:DropShadow>

</yummy:PancakeView.Shadow>

<StackLayout Grid.Row="2" Padding="0,0,0,5">

<Grid>

<lottie:AnimationView AutoPlay="True"
Animation="Animation - 1708864593593.json"
HeightRequest="90"
IsVisible="{ Binding IsLoading }"
Grid.Row="0"
Grid.Column="0" />

<StackLayout IsVisible="{ Binding
IsLoadingServices }"

Spacing="5" Grid.Row="0"
Grid.Column="0">
```

```
<Image Source="{Binding ServiceIcon}"  
HeightRequest="110"  
Margin="0,5,0,0"  
Aspect="AspectFill"></Image>  
  
<Label Text="{Binding ServiceName}"  
FontFamily="DG" HorizontalOptions="CenterAndExpand"></Label>  
  
</StackLayout>  
  
</Grid>  
  
</StackLayout>  
  
<yummy:PancakeView.GestureRecognizers>  
  
<TapGestureRecognizer Command="{Binding  
BindingContext.SelectService, Source={x:Reference homepage}}"  
CommandParameter="{Binding}" />  
  
</yummy:PancakeView.GestureRecognizers>  
  
</yummy:PancakeView>  
  
</ViewCell>  
  
</DataTemplate>  
  
</syncfusion:SfListView.ItemTemplate>  
  
</syncfusion:SfListView>  
  
</Grid>
```

```
</xct:TabViewItem>

<xct:TabViewItem
    Text="{Res:Translate Text=lblOrders}"
    Style="{StaticResource TavView2}"
    TapCommand="{Binding GoToRequest}"
    >

    <xct:TabViewItem.Icon>
        <FontImageSource Glyph="□" Color="{DynamicResource BgColor}"
        FontFamily="FontIconSolid"></FontImageSource>
    </xct:TabViewItem.Icon>

    <Grid>
        <RefreshView x:Name="rrefreshview" IsRefreshing="{Binding
        IsRefreshing}">
            <Command>{Binding OnRefreshRequested}</Command>
        </RefreshView>
        <CollectionView ItemsSource="{Binding lstRequests}" >
            <CollectionView.Header>
                <StackLayout>
                    <control:ToolBar HeightRequest="80"
                        HasCartButton="False"
                        Title="{Res:Translate Text=lblOrders}"></control:ToolBar>
```

```
</StackLayout>

</CollectionView.Header>

<CollectionView.ItemsLayout>

    <LinearItemsLayout Orientation="Vertical"
ItemSpacing="10"></LinearItemsLayout>

</CollectionView.ItemsLayout>

<CollectionView.ItemTemplate>

    <DataTemplate>

        <Grid ColumnSpacing="10" Padding="0,0,0,0">

            <Grid.RowDefinitions>

                <RowDefinition Height="auto"></RowDefinition>

                <RowDefinition Height="auto"></RowDefinition>

            </Grid.RowDefinitions>

            <Grid.ColumnDefinitions>

                <ColumnDefinition Width="auto"></ColumnDefinition>

                <ColumnDefinition Width="auto"></ColumnDefinition>

                <ColumnDefinition Width="auto"></ColumnDefinition>

                <ColumnDefinition Width="*"></ColumnDefinition>

            </Grid.ColumnDefinitions>

        </DataTemplate>
    </CollectionView.ItemTemplate>
</CollectionView>
```

```
</Grid.ColumnDefinitions>

<yummy:PancakeView BackgroundColor="White"
x:Name="mainView" Grid.ColumnSpan="5" CornerRadius="30" Margin="10" >

    <yummy:PancakeView.Border>
        <yummy:Border Color="#E36414"
Thickness="5"></yummy:Border>

    </yummy:PancakeView.Border>

    <StackLayout Padding="20,10,10,10" >

        <Image HorizontalOptions="EndAndExpand"
IsVisible="{Binding isVisibleButton}" VerticalOptions="StartAndExpand"
x:Name="dellIcon"

    >

<Image.Source>

    <FontImageSource Glyph="□" Color="#952323"
FontFamily="FontIconSolid"></FontImageSource>

</Image.Source>

<Image.GestureRecognizers>

    <TapGestureRecognizer
Tapped="TapGestureRecognizer_Tapped" Command="{Binding
BindingContext.RemoveFromCart, Source={x:Reference homepage}}"
CommandParameter="{Binding}" ></TapGestureRecognizer>
```

```
</Image.GestureRecognizers>

</Image>

<Label TextColor="Black" Grid.ColumnSpan="4"
Text="{Binding ServiceName}" FontFamily="DG"
HorizontalTextAlignment="Center" VerticalTextAlignment="Center"></Label>

<StackLayout HorizontalOptions="FillAndExpand"
>

    <Label Text="{Res:Translate cuImage}"
FontFamily="DG" TextColor="#FF6C22"
HorizontalOptions="CenterAndExpand" HorizontalTextAlignment="Center"
VerticalTextAlignment="Center"></Label>

    <Image Source="{Binding ImageNameProblem}"
BackgroundColor="Aqua" HeightRequest="130"
HorizontalOptions="CenterAndExpand" ></Image>

    </StackLayout>

    <StackLayout Orientation="Horizontal"
HorizontalOptions="FillAndExpand">

        <Label Text="{Res:Translate providerName}"
TextColor="#FF6C22" FontFamily="DG" HorizontalOptions="StartAndExpand"
VerticalTextAlignment="Center"></Label>

        <Label Text="{Binding ProviderName}"
HorizontalOptions="EndAndExpand" VerticalTextAlignment="Center"></Label>

    </StackLayout>
```

```
<StackLayout Orientation="Horizontal"
HorizontalOptions="FillAndExpand">

    <Label Text="{Res:Translate cuCost2}"
TextColor="#FF6C22" FontFamily="DG" HorizontalOptions="StartAndExpand"
VerticalTextAlignment="Center"></Label>

    <Label Text="{Binding Cost, StringFormat='
{0:F}₪'}" HorizontalOptions="EndAndExpand"
VerticalTextAlignment="Center"></Label>

</StackLayout>

<StackLayout Orientation="Horizontal"
HorizontalOptions="FillAndExpand">

    <Label Text="{Res:Translate cuDate}"
FontFamily="DG" TextColor="#FF6C22" HorizontalTextAlignment="Center"
VerticalTextAlignment="End"></Label>

    <Label Text="{Binding Date}"
HorizontalOptions="EndAndExpand" VerticalTextAlignment="End"></Label>

</StackLayout>

<StackLayout Orientation="Horizontal"
VerticalOptions="EndAndExpand" HorizontalOptions="CenterAndExpand">
```

```
<Label Text="{Binding statusLabel}"  
FontSize="20" FontFamily="DG" TextColor="{Binding StatusColor}"  
HorizontalOptions="CenterAndExpand" />  
  
<Image HorizontalOptions="EndAndExpand"  
IsVisible="{Binding IsConfirmed}">  
  
    <Image.Source>  
        <FontImageSource Glyph="□" Color="#65B741"  
FontFamily="FontIconSolid"></FontImageSource>  
  
    </Image.Source>  
  
    </Image>  
  
<Image HorizontalOptions="EndAndExpand"  
IsVisible="{Binding IsWaiting}">  
  
    <Image.Source>  
        <FontImageSource Glyph="□"  
Color="#0766AD" FontFamily="FontIconSolid"></FontImageSource>  
  
    </Image.Source>  
  
    </Image>  
  
</StackLayout>
```

```
<StackLayout x:Name="confirmationButtons"
HorizontalOptions="CenterAndExpand" Padding="0,0,0,20">

    <Button Text="{Res:Translate Confirmed}"
IsVisible="{Binding IsButtonEnabled}" Command="{Binding
BindingContext.Confirmed, Source={x:Reference homepage}}"
CommandParameter="{Binding}" BackgroundColor="#21BF73"
TextColor="White" BorderWidth="3" BorderColor="#E8F9FD"
CornerRadius="15" FontFamily="DG" ></Button>

</StackLayout>

</StackLayout>

</yummy:PancakeView>

</Grid>

</DataTemplate>

</CollectionView.ItemTemplate>

</CollectionView>

</RefreshView>

</Grid>

</xct:TabViewItem>

<xct:TabViewItem
Text="{Res:Translate Text=lblChat}"
Style="{StaticResource TavView2}">
```

```
TapCommand="{Binding GoChat}">  
<xct:TabViewItem.Icon>  
    <FontImageSource Glyph="□" Color="{DynamicResource BgColor}"  
        FontFamily="FontIconSolid"></FontImageSource>  
</xct:TabViewItem.Icon>  
  
</xct:TabViewItem>  
  
<xct:TabViewItem Style="{StaticResource TavView2}"  
    TapCommand="{Binding GotToProfile}"  
    Text="{Res:Translate Text=lblProfile}"  
    FontSize="12">  
  
<xct:TabViewItem.Icon>  
    <FontImageSource Glyph="□" Color="{DynamicResource BgColor}"  
        FontFamily="FontIconSolid"></FontImageSource>  
</xct:TabViewItem.Icon>  
  
<Grid BackgroundColor="White" RowDefinitions="auto,120">  
    >
```

```
<Grid RowDefinitions="100,120,auto,auto,2*,auto,auto" >

    <Path Data="M0,0H375V262.033s-9.025,4.813-55.426-10.71-70.141-
83.047-130.3-83.22-83.544,67.7-130.893,83.22S0,262.033,0,262.033Z"
Grid.RowSpan="2"

        Aspect="Fill" Fill="#041E31"/>

        <StackLayout HorizontalOptions="CenterAndExpand" Grid.Row="0"
Grid.RowSpan="2" >

            <Label Text="{Res:Translate lblProfile}" FontFamily="DG"
FontSize="20" TextTransform="None"

                HorizontalOptions="CenterAndExpand"

                TextColor="White"></Label>

            <StackLayout VerticalOptions="CenterAndExpand"

                HorizontalOptions="CenterAndExpand"

                Grid.Row="0" Orientation="Vertical" >

                <Label TextTransform="None" Text="{Res:Translate
Text=lblWelcome}"

                    HorizontalOptions="CenterAndExpand"

                    FontFamily="DG"
                    FontSize="20"

                    TextColor="Orange"
```

```
></Label>

<!--<Label
    TextTransform="None"
    HorizontalOptions="CenterAndExpand"
    x:Name="txtemail"

    FontFamily="Helvetica"
    FontSize="20"
    TextColor="Black"></Label>-->

</StackLayout>

<StackLayout Orientation="Horizontal"
    HorizontalOptions="CenterAndExpand">

    <StackLayout HorizontalOptions="CenterAndExpand"
        Orientation="Horizontal" >

        <yummy:PancakeView CornerRadius="60"
            HorizontalOptions="CenterAndExpand" Padding="0" >

            <yummy:PancakeView.Border>
                <yummy:Border Color="Black"
                    Thickness="2"></yummy:Border>

            </yummy:PancakeView.Border>

        <StackLayout Orientation="Horizontal"
            HorizontalOptions="CenterAndExpand">
```

```
<Image Source="{Binding SelectedImage}"  
Aspect="Fill" WidthRequest="100" HeightRequest="100" ></Image>  
  
</StackLayout>  
  
</yummy:PancakeView>  
  
</StackLayout>  
  
</StackLayout>  
  
<ImageButton BackgroundColor="Transparent"  
HeightRequest="25"  
HorizontalOptions="CenterAndExpand"  
Command="{Binding imgPic}" >  
  
<ImageButton.Source>  
  
<FontImageSource Glyph="□" Color="#0F2C59"  
FontFamily="FontIconSolid"></FontImageSource>  
  
</ImageButton.Source>  
  
</ImageButton>  
  
</StackLayout>  
  
<StackLayout Spacing="20"  
HorizontalOptions="FillAndExpand"  
Grid.Row="3" Orientation="Vertical" >  
  
<StackLayout HorizontalOptions="CenterAndExpand"  
Grid.Row="3">
```

```
<Label TextColor="#070F2B"
    Grid.ColumnSpan="2"
    VerticalTextAlignment="Center"
    Text="{Binding Username}"
    HorizontalOptions="CenterAndExpand"
    FontFamily="Helvetica" ></Label>

<!--<Label BackgroundColor="White"
    Grid.ColumnSpan="2"
    HorizontalTextAlignment="Center"
    TextDecorations="Underline"
    TextColor="Red" Text="{Res:Translate
lblChangePassword}">

    FontFamily="DG">
    <Label.GestureRecognizers >
        <TapGestureRecognizer Command="{ Binding
ChangePassword }"></TapGestureRecognizer>
    </Label.GestureRecognizers>
    </Label>-->
</StackLayout >
```

```
<StackLayout HorizontalOptions="FillAndExpand"
Grid.Row="4" Padding="20,0,20,0" >

    <yummy:PancakeView BackgroundColor="LightGray"
Grid.Row="4" Padding="10" CornerRadius="15"

        BackgroundGradientStartPoint="0,0"
        BackgroundGradientEndPoint="1,1"
        HorizontalOptions="FillAndExpand">

        <yummy:PancakeView.Shadow>

            <yummy:DropShadow Color="White" Opacity="1"
></yummy:DropShadow>

        </yummy:PancakeView.Shadow>

        <StackLayout Orientation="Horizontal"
HorizontalOptions="FillAndExpand" >

            <Label Text="{Res:Translate lblEmail}"
VerticalTextAlignment="Start" TextColor="Black"
HorizontalOptions="StartAndExpand" FontFamily="DG"></Label>

            <Label VerticalTextAlignment="Center" Text="{Binding
UserEmail}" TextColor="#365486" HorizontalOptions="EndAndExpand"
FontFamily="DG"></Label>

        </StackLayout>

    </yummy:PancakeView>
```

```
</StackLayout>

<StackLayout HorizontalOptions="FillAndExpand"
Grid.Row="5" Padding="20,0,20,0" >

    <yummy:PancakeView BackgroundColor="LightGray"
Grid.Row="4" Padding="10" CornerRadius="15"
        BackgroundGradientStartPoint="0,0"
        BackgroundGradientEndPoint="1,1"
        HorizontalOptions="FillAndExpand"
    >

    <yummy:PancakeView.Shadow>
        <yummy:DropShadow Color="White" Opacity="1"
    ></yummy:DropShadow>

    </yummy:PancakeView.Shadow>

    <StackLayout Orientation="Horizontal"
HorizontalOptions="FillAndExpand" >

        <Label Text="{Res:Translate lblPhoneNumber}"
VerticalTextAlignment="Start" TextColor="Black"
HorizontalOptions="StartAndExpand" FontFamily="DG"></Label>
```

```
<Label VerticalTextAlignment="Center" Text="{ Binding  
PhoneNumber }" TextColor="#365486" HorizontalOptions="EndAndExpand"  
FontFamily="DG"></Label>  
  
</StackLayout>  
  
  
  
</yummy:PancakeView>  
  
</StackLayout>  
  
</StackLayout>  
  
<StackLayout HorizontalOptions="FillAndExpand" Grid.Row="4"  
Padding="20,20,20,0" >  
  
    <yummy:PancakeView BackgroundColor="White" Padding="20"  
CornerRadius="15"  
  
    BackgroundGradientStartPoint="0,0"  
  
    BackgroundGradientEndPoint="1,1"  
  
    HorizontalOptions="FillAndExpand"  
  
>  
  
  
  
  
<yummy:PancakeView.Shadow>  
  
    <yummy:DropShadow Color="White" Opacity="1"  
></yummy:DropShadow>  
  
</yummy:PancakeView.Shadow>
```

```
<StackLayout Orientation="Horizontal"
    HorizontalOptions="FillAndExpand"
    >

<Label BackgroundColor="White"
    HorizontalOptions="CenterAndExpand"
    TextColor="#2E4374"
    VerticalOptions="CenterAndExpand"
    TextTransform="None"
    FontFamily="DG"
    Text="{Res:Translate lblSetting}" >
    <Label.GestureRecognizers>
        <TapGestureRecognizer Command="{Binding
GoSettings}"></TapGestureRecognizer>
    </Label.GestureRecognizers>
</Label>
<Image
    Margin="0,0,0,0"
    HorizontalOptions="CenterAndExpand" >
    <Image.Source>
```

```
<FontImageSource Glyph="□" Color="#2E4374"
FontFamily="FontIconSolid">

</FontImageSource>

</Image.Source>

</Image>

<StackLayout.GestureRecognizers>

<TapGestureRecognizer Command="{Binding GoGmail}">

</TapGestureRecognizer>

</StackLayout.GestureRecognizers>

</StackLayout>

</yummy:PancakeView>

</StackLayout>

<!--<StackLayout HorizontalOptions="CenterAndExpand"
Padding="10,0,10,0" Grid.Row="4" >

<Label Text="{Res:Translate lblChooseLanguage}"
FontSize="20" TextTransform="None"
VerticalOptions="StartAndExpand"

FontFamily="DG"
HorizontalOptions="CenterAndExpand"
```

```
FontAttributes="Bold"

TextColor="#0F2C59">></Label>

<Button Text="English" Grid.Row="0" CornerRadius="30"
BorderColor="White" BorderWidth="3" FontFamily="DG"
BackgroundColor="Black" TextColor="White" Command="{Binding
NavigateEnglish}"></Button>

<Button Text="عربي" Grid.Row="1" BorderColor="Black"
BorderWidth="3" FontFamily="DG" BackgroundColor="White"
TextColor="Black" Command="{Binding NavigateArabic}" ></Button>

</StackLayout>

</Grid >

<StackLayout Grid.Row="1" Padding="5,0,5,0"
VerticalOptions="CenterAndExpand" >

<Button Text="{Res:Translate lblLogOut}" BorderColor="Red"
BorderWidth="2"
FontFamily="DG" BackgroundColor="White"
TextColor="Red"
CornerRadius="15"
xct:TouchEffect.PressedOpacity=".7"
xct:TouchEffect.NormalBackgroundColor="White"
xct:TouchEffect.PressedBackgroundColor="Red"
xct:TouchEffect.PressedAnimationDuration="100"
```

```
    HorizontalOptions="FillAndExpand"  
  
    Command="{Binding logout}" ></Button>  
  
  </StackLayout>  
  
</Grid>  
  
</xct:TabViewItem>  
  
</xct:TabView>  
  
</control>
```

```
using System;  
using System.Collections.Generic;  
using System.Globalization;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
  
using Xamarin.Forms;  
using Xamarin.Forms.Xaml;  
using Newtonsoft.Json;  
using Plugin.StoreReview;  
using Xamarin.Forms.PancakeView;  
  
namespace otp.Pages  
{  
    [XamlCompilation(XamlCompilationOptions.Compile)]  
    public partial class HomePage : Controls.Custompage  
    {  
        public HomePage()  
        {  
            InitializeComponent();  
        }  
  
        private void TapGestureRecognizer_Tapped(object sender, EventArgs e)  
        {  
            Image delIcon = (Image)sender;
```

```
        if (delIcon.Source is FontImageSource iconsource)
        {
            delIcon.ScaleTo(0, 250);
        }
    }
}
```

using System;

using System.Collections.Generic;

using System.Collections.ObjectModel;

using System.Globalization;

using System.Linq;

using System.Text;

using otp.Models;

using FreshMvvm;

using Xamarin.Forms;

using Xamarin.Essentials;

using System.IO;

using System.Threading.Tasks;

using Newtonsoft.Json;

using Plugin.StoreReview;

using otp.Helpers;

using Rg.Plugins.Popup.Extensions;

```
namespace otp.ViewModels

{

    public class HomeViewModel:FreshBasePageModel

    {

        public string Title { get; set; } = "في الخدمة";

        ObservableCollection<ServiceModel> _ListServices;

        public ObservableCollection<ServiceModel> ListServices {

            get

            {

                return _ListServices;

            }

            set

            {

                _ListServices = value;

                RaisePropertyChanged("ListServices");

            }

        }

        bool isRefreshing;

        public bool IsRefreshing

        {
```

```
get { return isRefreshing; }

set

{

    if (isRefreshing != value)

    {

        isRefreshing = value;

        RaisePropertyChanged(nameof(IsRefreshing));

    }

}

}
```

```
ObservableCollection<ServiceModel> _ListFilterServicesl;

public ObservableCollection<ServiceModel> ListFilterServicesl

{

    get

    {

        return _ListFilterServicesl;

    }

    set
```

```
{  
    _ListFilterServices1 = value;  
    RaisePropertyChanged("ListFilterServices1");  
}  
  
}  
  
private bool _isLoading;  
  
public bool IsLoading  
{  
    get { return _isLoading; }  
  
    set  
    {  
        if (_isLoading != value)  
        {  
            _isLoading = value;  
            RaisePropertyChanged("IsLoading");  
        }  
    }  
}  
  
}  
  
private bool _IsLoadingServices;
```

```
public bool IsLoadingServices
{
    get { return _IsLoadingServices; }

    set
    {
        if (_IsLoadingServices != value)
        {
            _IsLoadingServices = value;
            RaisePropertyChanged("IsLoadingServices");
        }
    }
}
```

```
ObservableCollection<Models.RequestModel> _lstRequests;

public ObservableCollection<Models.RequestModel> lstRequests
{
    get
    {
        return _lstRequests;
    }
}
```

```
set

{

    _lstRequests = value;

    RaisePropertyChanged("lstRequests");

}

public Command NavigateEnglish { get; set; }

public Command NavigateArabic { get; set; }

public Command SelectService { get; set; }

public Command SearchWithText { get; set; }

public Command OnRefreshRequested { get; }

public Command logout { get; set; }

public Command ChangePassword { get; set; }

public Command imgPic { get; set; }

public Command GoChat { get; set; }

public Command GoToRequest { get; set; }

public Command RemoveFromCart { get; set; }

public Command Confirmed { get; set; }

public Command GoSettings { get; set; }
```

```
public HomeViewModel()
{
    ChangePassword = new Command(OnChangePassword);
    imgPic = new Command(OnimgPic);
    RemoveFromCart = new
Command<RequestModel>(OnRemoveFromCart);
    Listusers = new ObservableCollection<users>();
    SearchWithText = new Command(OnSearchWithText);
    GoToRequest = new Command(OnGoToRequest);
    GoChat = new Command(OnGoChat);
    GoSettings = new Command(OnGoSettings);
    SelectService = new Command<ServiceModel>(OnSelectService);
    ListServices = new ObservableCollection<ServiceModel>();
    foreach (var service in ListServices)
    {
        service.ServiceName = (Helpers.Settings.Language == "ar") ?
service.ServiceNameAr : service.ServiceName;
    }
    OnRefreshRequested = new Command(async () => await
RefreshCategories());
}
```

```
lstRequests = new ObservableCollection<RequestModel>();  
  
logout = new Command(Onlogout);  
  
Confirmed = new Command<RequestModel>(OnConfirmed);  
  
}  
  
string _SearchTerm;  
  
public string SearchTerm  
  
{  
  
    get  
  
    {  
  
        return _SearchTerm;  
  
    }  
  
    set  
  
    {  
  
        _SearchTerm = value;  
  
        RaisePropertyChanged("SearchTerm");  
  
    }  
  
}  
  
private ImageSource _selectedImage;
```

```
public ImageSource SelectedImage
{
    get => _selectedImage;
    set
    {
        _selectedImage = value;
        RaisePropertyChanged("SelectedImage");
    }
}

async void OnSearchWithText()
{
    if (string.IsNullOrEmpty(SearchTerm))
    {
        ListFilterServices1 = new
ObservableCollection<ServiceModel>(ListServices);
        // lstFiltermodel2 = new
ObservableCollection<itemPhoneModel>(lstmodel2);
        return;
    }
}
```

```
if (Helpers.Settings.Language == "ar")  
{  
    ListFilterServices1 = new  
    ObservableCollection<ServiceModel>(ListServices.Where(a =>  
        a.ServiceNameAr.Contains(SearchTerm)));  
  
}  
  
else  
  
{  
    ListFilterServices1 = new  
    ObservableCollection<ServiceModel>(ListServices.Where(a =>  
        a.ServiceName.Contains(SearchTerm)));  
  
}  
  
//lstFiltermodel2 = new  
ObservableCollection<itemPhoneModel>(lstFiltermodel2.Where(a =>  
    a.ItemName.Contains(SearchTerm)));  
  
}  
  
async void OnSelectService(ServiceModel service)  
{  
    string json1 = await Helpers.Utility.CallWebApi("/api/user");
```

```
List<users> Listusers =  
JsonConvert.DeserializeObject<List<users>>(json1);  
  
string userinfoString = Helpers.Settings.userinfo;  
  
var userList =  
JsonConvert.DeserializeObject<List<users>>(userinfoString);  
  
Listusers.Reverse();  
  
  
  
if (!string.IsNullOrEmpty(Username))  
{  
  
    // Check if the customer name exists in the userList  
  
    var user = Listusers.FirstOrDefault(u => u.CustomerName ==  
Username);  
  
    if (user != null)  
  
    {  
  
        service.userId = user.CustomerId;  
  
    }  
  
}  
  
await  
CoreMethods.PushPageModel<ServiceMapViewModel>(service,modal:true);  
}  
  
private string _phoneNumber;
```

```
public string PhoneNumber
{
    get => _phoneNumber;
    set
    {
        if (_phoneNumber != value)
        {
            _phoneNumber = value;
            RaisePropertyChanged(nameof(PhoneNumber));
        }
    }
}

public string _UserEmail;

public string UserEmail
{
    get
    {
        return _UserEmail;
    }
}
```

```
set  
{  
    _UserEmail = value;  
    RaisePropertyChanged("UserEmail");  
}  
  
}  
  
private string _Username;  
  
public string Username  
{  
    get => _Username;  
    set  
    {  
        if (_Username != value)  
        {  
            _Username = value;  
            RaisePropertyChanged(nameof(Username));  
        }  
    }  
}
```

```
public async override void Init(object initData)
{
    IsLoading = true;
    IsLoadingServices = false;

    try
    {
        var _service = new Services.ServicesService();
        ListServices = await _service.GetServices();

        ListFilterServices1 = new
ObservableCollection<ServiceModel>(ListServices);

        foreach (var service in ListServices)

        {
            // service.ServiceName = (Helpers.Settings.Language == "ar") ?
service.ServiceNameAr : service.ServiceName;

            if (Helpers.Settings.Language == "en")

            {
                service.ServiceName = service.ServiceName;

                CultureInfo cal;
                cal = new CultureInfo("en");

```

```
Resources.AppResources.Culture = cal;

}

else

{

    service.ServiceName = service.ServiceNameAr;

    CultureInfo cal;

    cal = new CultureInfo("ar");

    Resources.AppResources.Culture = cal;

}

}

finally

{

    IsLoadingServices = true;

    IsLoading = false;

}

string json1 = await Helpers.Utility.CallWebApi("/api/user");

List<users> Listusers = JsonConvert.DeserializeObject<List<users>>(json1);
```

```
// Deserialize the userinfo string back to a list of users

string customerName = initData.ToString();

if (!string.IsNullOrEmpty(customerName))

{

    // Check if the customer name exists in the userList

    var user = Listusers.FirstOrDefault(u => u.CustomerName ==

customerName);

    if (user != null)

    {

        // Found user with the matching customer name

        Username = user.CustomerName;

        PhoneNumber = user.CustomerPhone;

        UserEmail = user.CustomerEmail;

        int customerId = user.CustomerId;

    }

}

string userinfoString = Helpers.Settings.userinfo;

var userList =

JsonConvert.DeserializeObject<List<users>>(userinfoString);

if (userList != null && userList.Any())
```



```
string _imgname;

public string imgname

{

    get

    {

        return _imgname;

    }

    set

    {

        _imgname = value;

        RaisePropertyChanged("imgname");

    }

}

async void OnChangePassword()

{

    await

CoreMethods.PushPageModel<ResetPasswordViewModel>(null,true);

}

async void OnimgPic()

{
```

```
string action = await Application.Current.MainPage.DisplayActionSheet("Pick Image", "Cancel", null, "Camera", "Gallery");

if (action == "Camera" && !MediaPicker.IsCaptureSupported)
{
    await Application.Current.MainPage.DisplayAlert("Error", "Camera is not supported on this device.", "OK");
    return;
}

try
{
    MediaPickerOptions options = new MediaPickerOptions
    {
        Title = "تغيير الصوره الشخصية"
    };
}

FileResult result = null;

if (action == "Camera")
```

```
{  
    result = await MediaPicker.CapturePhotoAsync(options);  
}  
  
else if (action == "Gallery")  
  
{  
    result = await MediaPicker.PickPhotoAsync(options);  
}  
  
else  
  
{  
    return;  
}  
  
if (result != null)  
  
{  
    string imageUrl = result.FullPath;  
  
    imgname = imageUrl;  
  
    string userinfoString = Helpers.Settings.userInfo;  
  
    var userList =  
        JsonConvert.DeserializeObject<List<users>>(userinfoString);  
}
```

```
if (userList != null && userList.Any())
{
    // Find the user to update
    users userToUpdate = userList.FirstOrDefault(u =>
        u.CustomerName == Username);
    if (userToUpdate != null)
    {
        userToUpdate.SelectedImageUrl = imgname;
        SelectedImage = ImageSource.FromUri(new Uri(imageUrl));
        string imagePath = result.FullPath;
        Stream imageStream = await result.OpenReadAsync();
        SelectedImage = ImageSource.FromStream(() => imageStream);
    }
    string updatedUserInfo = JsonConvert.SerializeObject(userList);
    Helpers.Settings.userinfo = updatedUserInfo;
}

}

catch (Exception ex)
```

```
{  
    await Application.Current.MainPage.DisplayAlert("Error", $"Error:  
{ex.Message}", "OK");  
  
}  
}
```

```
protected override void ViewIsAppearing(object sender, EventArgs e)
```

```
{  
    base.ViewIsAppearing(sender, e);  
  
    string userinfoString = Helpers.Settings.userInfo;  
  
    var userList =  
JsonConvert.DeserializeObject<List<users>>(userinfoString);
```

```
if (userList != null && userList.Any())  
{  
    var currentUser = userList.FirstOrDefault(u => u.CustomerName ==  
Username);
```

```
if (currentUser != null)  
{  
    SelectedImage = currentUser.SelectedImageUrl;  
}
```

```
    }

}

ObservableCollection<users> _Listusers;

public ObservableCollection<users> Listusers

{

    get

    {

        return _Listusers;

    }

    set

    {

        _Listusers = value;

        RaisePropertyChanged("Listusers");

    }

}

bool _isConfirmed;

public bool IsConfirmed

{

    get { return _isConfirmed; }
```

```
set

{
    if (_isConfirmed != value)
        {
            _isConfirmed = value;
            RaisePropertyChanged(nameof(IsConfirmed));
        }
}

}

string _StatusLabel;

public string StatusLabel
{
    get { return _StatusLabel; }

    set
    {
        if (_StatusLabel != value)
            {
                _StatusLabel = value;
                RaisePropertyChanged("StatusLabel");
            }
    }
}
```

```
    }

}

async Task<ObservableCollection<RequestModel>> GetCategories()

{

    try

    {

        string json1 = await Helpers.Utility.CallWebApi("/api/user");

        List<users> Listusers =

JsonConvert.DeserializeObject<List<users>>(json1);

        string userinfoString = Helpers.Settings.userinfo;

        var userList =

JsonConvert.DeserializeObject<List<users>>(userinfoString);

        Listusers.Reverse();

        users userFromApi = null;

        int customerId;

        var user = Listusers.FirstOrDefault(u => u.CustomerName ==

Username);

        if (user != null)

        {

            Username = user.CustomerName;

            PhoneNumber = user.CustomerPhone;
```

```

userFromApi = Listusers.FirstOrDefault(u => u.CustomerName ==
Username && u.CustomerPhone == PhoneNumber);

if (userFromApi != null)

{

    customerId = userFromApi.CustomerId;

}

customerId = user.CustomerId;

}

if (userinfoString != null)

{

    string json = await

Helpers.Utility.CallWebApi($""/api/Request/{userFromApi.CustomerId}");



ObservableCollection<RequestModel> lstRequests =

JsonConvert.DeserializeObject<ObservableCollection<RequestModel>>(json);

if (lstRequests.Count == 0)

    await

CoreMethods.PushPageModel<EmptyListViewModel>(null,modal:true);

foreach (var request in lstRequests)

```

```
{  
    Image image = new Image();  
  
    if (Uri.TryCreate(request.ImageNameProblem, UriKind.Absolute,  
out Uri uri) && uri.IsFile)  
  
    {  
        image.Source = ImageSource.FromFile(uri.LocalPath);  
  
    }  
  
    if (request.StatusOfRequest == true)  
  
    {  
        request.statusLabel = "تم قبول طلبك";  
  
        request.IsConfirmed = true;  
  
        request.StatusColor = Color.FromHex("#65B741");  
  
        request.IsEnabled = false;  
  
        request.isVisableButton = false;  
  
        request.IsWaiting = false;  
  
    }  
  
    else if (request.StatusOfRequest == null)  
  
    {  
        request.statusLabel = "...طلبك قيد الانتظار";  
  
        request.StatusColor = Color.FromHex("#0766AD");  
    }  
}
```

```
request.IsConfirmed = false;  
  
request.isVisableButton = true;  
  
request.IsWaiting = true;  
  
request.IsEnabled = false;  
  
}  
  
else if (request.StatusOfRequest == false)  
  
{  
  
    request.statusLabel = "تم تعديل طلبك ..";  
  
    request.StatusColor = Color.FromHex("#952323");  
  
    request.IsConfirmed = false;  
  
    request.isVisableButton = true;  
  
    request.IsEnabled = true;  
  
  
    request.IsWaiting = false;  
  
}  
  
}  
  
IsLoading = true;
```

```
    IsRefreshing = false;  
  
    return lstRequests;  
  
}  
  
}  
  
catch (Exception ex)  
  
{  
  
    await Application.Current.MainPage.DisplayAlert("Error", $"Error:  
{ex.Message}", "OK");  
  
}  
  
  
  
return null;  
  
}  
  
async Task RefreshCategories()  
  
{  
  
    IsRefreshing = true;  
  
    await Task.Delay(2000);  
  
    lstRequests.Clear();  
  
    lstRequests = await GetCategories();  
  
    foreach (var request in lstRequests)  
  
    {
```

```
Image image = new Image();

if (request.IsResponse == true)

{

    request.statusLabel = "سيتم وصول مزود خدمتك ف الميعاد";

    request.StatusColor = Color.FromHex("#65B741");

    request.IsConfirmed = true;

    request.isVisableButton = false;

    request.IsEnabled = false;

    request.IsWaiting = false;

}

IsLoading = true;

IsRefreshing = false;

}

async public void Onlogout()

{

    DependencyService.Get<IAuthenticationService>().SignOut();
```

```
    await CoreMethods.PushPageModel < LoginViewModel  
>(Username,modal:true);  
}
```

```
async void OnConfirmed(RequestModel oRequestModel)  
{  
    int id = oRequestModel.ID;  
    oRequestModel.IsResponse = true;  
  
    string result = await  
Utility.UpdateStatusOfResponse(id,oRequestModel.StatusOfRequest,oRequestMo  
del.IsResponse,oRequestModel.Cost,oRequestModel.Date);  
  
    foreach (var request in lstRequests)  
    {  
        if (request.StatusOfRequest == false)  
        {  
            request.statusLabel = "...تم تعديل طلبك ..";  
            request.StatusColor = Color.FromHex("#952323");  
            request.IsConfirmed = true;  
            request.isVisableButton = false;  
            request.IsEnabled = false;
```

```
    request.IsWaiting = false;

}

}

oRequestModel.IsEnabled = false;
oRequestModel.isVisableButton = false;

await App.Current.MainPage.Navigation.PushPopupAsync(new
Helpers.MsgSendToProvider(), true);

}

async void OnRemoveFromCart(RequestModel ORequestModel)
{
    string result = await
Helpers.Utility.DeleteData($"{Helpers.Utility.ServerUrDatabase}/api/Request",
ORequestModel.ID);

    await App.Current.MainPage.Navigation.PushPopupAsync(new
Helpers.MsgDelete(), true);

}

async void OnGoToRequest()
```

```
{  
    await GetCategories();  
}  
  
async void OnGoSettings()  
{  
    await  
CoreMethods.PushPageModel<SettingsViewModel>(Username,modal:true);  
}  
  
}  
  
async void OnGoChat()  
{  
    await CoreMethods.PushPageModel<ChatViewModel>(null,true);  
}  
}  
}
```

5.2 Website Dashboard Code:

5.2.1 Main Page:

```
<!doctype html>

<html lang="en" class="h-100">

    <head>
        <meta charset="utf-8">
        <meta name="viewport"
            content="width=device-width, initial-scale=1, shrink-to-fit=no">
        <meta http-equiv="X-UA-Compatible" content="IE=edge">
        <link rel="shortcut icon" type="image/x-icon" href="assets/img/Logo (1).png">
        <title>NS-dashboard</title>
        <link href="assets/css/bootstrap.css" rel="stylesheet">
        <link href="assets/css/main.css" rel="stylesheet">
    </head>

    <body class="d-flex flex-column h-100">
        <div id="page">
            <div class="wrapper">
                <nav id="sidebar" class="active">
                    <div class="sidebar-header text-center">
                        
                        <h4 class="sidebar-title theme-item">NS</h4>
                    </div>
                    <ul class="list-group sidebar-list">
                        <li class="list-item"><a href="#">Dashboard</a></li>
                        <li class="list-item"><a href="#">Users</a></li>
                        <li class="list-item"><a href="#">Products</a></li>
                        <li class="list-item"><a href="#">Orders</a></li>
                        <li class="list-item"><a href="#">Reports</a></li>
                    </ul>
                </nav>
                <div class="main-content">
                    <div class="content-header">
                        <h1>NS Dashboard</h1>
                        <p>Welcome to the NS Dashboard. Here you can manage users, products, orders, and reports. Stay organized and efficient!</p>
                    </div>
                    <div class="content-body">
                        <div class="card">
                            <div class="card-header">Recent Orders</div>
                            <div class="card-body">
                                <table border="1">
                                    <thead>
                                        <tr>
                                            <th>Order ID</th>
                                            <th>Product Name</th>
                                            <th>Quantity</th>
                                            <th>Status</th>
                                        </tr>
                                    </thead>
                                    <tbody>
                                        <tr>
                                            <td>12345678901234567890</td>
                                            <td>Product A</td>
                                            <td>10</td>
                                            <td>Completed</td>
                                        </tr>
                                        <tr>
                                            <td>12345678901234567891</td>
                                            <td>Product B</td>
                                            <td>5</td>
                                            <td>Pending</td>
                                        </tr>
                                        <tr>
                                            <td>12345678901234567892</td>
                                            <td>Product C</td>
                                            <td>20</td>
                                            <td>Completed</td>
                                        </tr>
                                        <tr>
                                            <td>12345678901234567893</td>
                                            <td>Product D</td>
                                            <td>15</td>
                                            <td>Pending</td>
                                        </tr>
                                    </tbody>
                                </table>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </body>

```

```
</div>

<ul class="list-unstyled components text-secondary">
    <li><a href="index.html"><i
        class="data-feather theme-item" data-
feather="home"></i> <span
        class="theme-item"> Dashboard</span></a></li>

    <li><a href="providers.html"><i
        class="data-feather theme-item" data-
feather="users"></i> <span
        class="theme-item"> Providers</span></a></li>

    <li><a href="Services.html"><i
        class="data-feather theme-item" data-
feather="grid"></i> <span
        class="theme-item"> Services</span></a></li>

    <li><a href="users.html"><i
        class="data-feather theme-item" data-
feather="user"></i> <span
        class="theme-item"> Users</span></a></li>

    <li><a href="AddService.html"><i class="data-feather theme-
item"
        data-feather="settings"></i> <span class="theme-
item">Add Service</span></a></li>

    <li><a href="addproviders.html"><i
        class="data-feather theme-item" data-
feather="plus"></i> <span
        class="data-feather theme-item"> Add
    Providers</span></a></li>

    <!-- <li><a href="loginregister.html"><i
        class="data-feather theme-item" data-
feather="plus"></i> <span
```

```
        class="data-feather theme-item"> Add  
Providers</span></a></li> -->  
  
    </ul>  
  
</nav>  
  
<div id="bodywrapper" class="container-fluid showhidetoggle">  
  
    <nav class="navbar navbar-expand-md navbar-white bg-white py-0"  
        aria-label="navbarexample" id="navbar">  
        <div class="container-fluid">  
            <button type="button" id="sidebarCollapse"  
                class="btn btn-light py-0">  
                <i data-feather="menu"></i> <span></span>  
            </button>  
              
            <h4 class="sidebar-title theme-item mt-2  
navbrandarea2">NS</h4>  
            <button class="navbar-toggler py-0" type="button"  
                data-bs-toggle="collapse" data-bs-  
target="#navbarsExample04"  
                aria-controls="navbarsExample04" aria-  
expanded="false"  
                aria-label="Toggle navigation">  
                <span class="navbar-toggler-icon"><i data-  
feather="menu"></i></span>  
            </button>  
  
            <div class="collapse navbar-collapse mx-1"  
id="navbarsExample04">
```

```

<ul class="navbar-nav me-auto mb-2 mb-lg-0">

    <li class="nav-item"><div class="nav-dropdown">
        <a class="nav-item nav-link active text-secondary py-0"
            href="index.html"><i
            class="data-feather theme-item" data-
            feather="home"></i> <span
            class="theme-item">Home </span></a>
        </div></li>

    </ul>

    <div class="usermenu">
        <div class="nav-dropdown py-0">
            <a href="#" class="nav-item nav-link dropdown-toggle
            text-secondary py-0"
                id="navbarDropdown3" role="button" data-
                bs-toggle="dropdown"
                aria-expanded="false">  <!--<i class="theme-item" -->
                <!--data-feather="user"--></i> --> <span
                class="theme-item">Admin
                </span><i class="theme-item" data-
                feather="chevron-down"></i></a>
        <ul class="dropdown-menu dropdown-menu-end"
            aria-labelledby="navbarDropdown3">
    
```

```

        <!-- <li><a href="profile.html"
class="dropdown-item mt-2"><i
                                class="data-feather" data-
feather="user"></i> Profile</a></li> -->

                <li><a href="#" class="dropdown-item mt-
2"
                                data-bs-toggle="modal" data-bs-
target="#settingsModal"><i
                                class="data-feather" data-
feather="settings"></i> Settings</a></li>

                <!-- <li><a href="#" class="dropdown-item
mt-2"><i
                                class="data-feather" data-
feather="log-out"></i> Logout</a></li> -->

            </ul>
        </div>
    </div>

    </div>
</div>
</div>

<div class="settings">
    <div class="modal fade" id="settingsModal"
        aria-labelledby="settingsModalTitle" aria-hidden="true"
        tabindex="-1">
        <!--
            data-bs-backdrop="static" data-bs-
keyboard="false" -->
            <div class="modal-dialog modal-dialog-settings">
                <div class="modal-content">
                    <div class="modal-header text-center">

```

```
<h5 class="modal-title"
id="exampleModalLabel">Settings</h5>

    <button type="button" class="btn-close" data-
bs-dismiss="modal"

        aria-label="Close"></button>

    </div>
    <div class="modal-body">

        <section id="logincontent" class="shiftdown">

            <div class="row g-3 mb-3 mt-3">

                <div class="col-md-6">

                    <h6 class="text-muted">Select
Color</h6>

<span onclick="changeColor('0')"
class="btn btn-sm btn-primary
rounded-circle"><span
class="me-2"></span></span>

<span onclick="changeColor('1')"
class="btn btn-sm btn-success
rounded-circle"><span
class="me-2"></span></span>

<span onclick="changeColor('2')"
class="btn btn-sm btn-danger
rounded-circle"><span
class="me-2"></span></span>

<span onclick="changeColor('3')"
class="btn btn-sm btn-warning
rounded-circle"><span
class="me-2"></span></span>

<span onclick="changeColor('4')"
class="btn btn-sm btn-light
rounded-circle"><span
class="me-2"></span></span>
```

```
        class="btn btn-sm btn-
secondary rounded-circle"><span>
            class="me-2"></span></span>
<div class="d-flex justify-
content-between">
    <button
        onclick="removeColorCookie()">Reset to
            Default</button>
</div>
</div>
<div class="col-md-6">
    <h6 class="text-
muted">Preferences</h6>
    <div class="form-check form-
switch">
        <input class="form-check-
input" type="checkbox"
            id="flexSwitchCheckDefault"> <label
            class="form-check-label"
            for="flexSwitchCheckDefault">Switch
            option 1</label>
</div>
    <div class="form-check form-
switch">
        <input class="form-check-
input" type="checkbox"
            id="flexSwitchCheckChecked"
            checked> <label
            class="form-check-label"
            for="flexSwitchCheckChecked">Switch
            option 2</label>
</div>
```

```
</div>

</div>

<div class="row g-3 mb-3 mt-3">
  <div class="col-md-6">
    <div class="form-check">
      <input class="form-check-input" type="checkbox" value="" checked> <label
        for="defaultCheck1" class="form-check-label"> Checkbox 1 </label>
    </div>
    <div class="form-check">
      <input class="form-check-input" type="checkbox" value="" checked> <label
        for="defaultCheck2" class="form-check-label"> Checkbox 2 </label>
    </div>
  </div>
  <div class="col-md-6">
    <h6 class="text-muted">View Size</h6>
    <div class="form-check">
      <input class="form-check-input" type="radio" name="flexRadioDefault" id="radioCompactView"> <label
        for="radioCompactView" class="form-check-label"> Compact View </label>
    </div>
    <div class="form-check">
      <input class="form-check-input" type="radio" name="flexRadioDefault" id="radioLargeView"> <label
        for="radioLargeView" class="form-check-label"> Large View </label>
    </div>
  </div>
</div>
```

```

    class="form-check-label"
for="radioCompactView">
    Compact</label>
</div>
<div class="form-check">
<input class="form-check-
input" type="radio"
name="flexRadioDefault"
id="radioFullView"> <label
class="form-check-label"
for="radioFullView">
    Full-screen </label>
</div>
<div class="d-flex justify-
content-between">
    <button
onclick="removeViewSizeCookie()">Reset to
        Default</button>
    </div>
</div>
</div>
<hr />
<button class="btn btn-sm btn-danger"
data-bs-dismiss="modal"
type="button">
    <i data-feather="check-circle"
class="mr-1"></i> Ok
    </button>
</section>
</div>
</div>

```

```
</div>

</div>

</div>

<div id="sidebarOverlayNav" class="screen-overlay float-end">
    <a href="javascript:void(0)" class="closebtn"
        onclick="closeOverlayNav()">&times;</a>
    <div class="screen-overlay-content text-secondary">
        <a href="#" class="active">About</a> <a
        href="#">Services</a> <a
        href="#">Clients</a> <a href="#">Contact</a>
    </div>
</div>

<div class="content">
    <div class="container-fluid">
        <div class="row mt-2">
            <div class="col-md-6 float-start">
                <h4 class="m-0 text-dark text-
muted">Dashboard</h4>
            </div>
            <!-- <div class="col-md-6">
                <ol class="breadcrumb float-end">
                    <li class="breadcrumb-item"><a href="#">
Home</a></li>
                    <li class="breadcrumb-item
active">Dashboard</li>
                </ol>
            </div> -->
        </div>
    </div>
</div>
```

```

<div class="containerAdmin">
    <h1>Welcome Admin</h1>
    <p>“Hello Admin! 🌟 Welcome to the Dashboard. Your insights and expertise are invaluable.

    Let’s dive into the data and make informed decisions together. If you need any assistance, feel free to reach out!”</p>
</div>
<br>
<div class="analysis">Here are some analyzes and statistics:</div>
<div class="row">
    <div class="col-sm-6 col-md-6 col-lg-3">
        <div class="card card-rounded">
            <div class="content">
                <div class="row">
                    <div class="col-sm-4">
                        <div class="icon-big text-center">
                            <i class="teal data-feather-big" stroke-width="3" data-feather="shopping-cart"></i>
                        </div>
                    </div>
                    <div class="col-sm-8">
                        <div class="detail">
                            <p class="detail-subtitle">Total Orders</p>
                            <span class="number requests"></span>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>

```

```
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
<div class="col-sm-6 col-md-6 col-lg-3">
<div class="card card-rounded">
<div class="content">
<div class="row">
<div class="col-sm-4">
<div class="icon-big text-center">
<i class="olive data-feather-big" stroke-width="3"
data-feather="users"></i>
</div>
</div>
<div class="col-sm-8">
<div class="detail">
<p class="detail-subtitle">Total Users</p>
<span class="number">users</span>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
```

```
<div class="col-sm-6 col-md-6 col-lg-3">
    <div class="card card-rounded">
        <div class="content">
            <div class="row">
                <div class="col-sm-4">
                    <div class="icon-big text-center">
                        <i class="orange data-feather-big" stroke-width="3" data-feather="tool"></i>
                    </div>
                </div>
                <div class="col-sm-8">
                    <div class="detail">
                        <p class="detail-subtitle">Total Services</p>
                        <span class="number services"></span>
                    </div>
                </div>
            </div>
        </div>
    </div>
<br>
<div class="row">
```

```
<div class="col-sm-6 col-md-6 col-lg-3">
    <div class="card card-rounded">
        <div class="content">
            <div class="row">
                <div class="col-sm-4">
                    <div class="icon-big text-center">
                        <i class="teal data-feather-big" stroke-width="3"
                           data-feather="user"></i>
                    </div>
                </div>
                <div class="col-sm-8">
                    <div class="detail">
                        <p class="detail-subtitle">Total providers</p>
                        <span class="number providers"></span>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>
<div class="col-sm-6 col-md-6 col-lg-3">
    <div class="card card-rounded">
        <div class="content">
            <div class="row">
                <div class="col-sm-4">
                    <div class="icon-big text-center">
```



```
<span  
    class="text-muted d-block text-center text-sm-left d-sm-inline-block">  
    Copyright © N-S <script>  
        document.write(new Date().getFullYear())  
    </script>  
</span> <span  
    class="float-none text-muted float-sm-right d-block mt-1 mt-sm-0  
text-center">  
    N-S dashboard  
</span>  
</div>  
</footer>  
<div id="loading" class="spinner-border text-primary align-middle"  
role="status"></div>  
  
<button class="btn btn-sm btn-primary rounded-circle"  
onclick="scrollToTopFunction()" id="scrollToTop" title="Scroll to top">  
    <i data-feather="arrow-up-circle"></i>  
</button>  
<script src="assets/js(feather.min.js)"></script>  
<script src="assets/js/bootstrap.bundle.min.js"></script>  
<script src="assets/js/Chart.min.js"></script>  
<script src="assets/js/script.js"></script>  
  
<script type="text/javascript">  
    document.addEventListener("DOMContentLoaded", function(event) {  
        feather.replace();  
    });
```

```
</script>

<script src="main.js"></script>

</body>

</html>
```

5.2.2 Show Services

```
<!doctype html>

<html lang="en" class="h-100">

<head>
<meta charset="utf-8">
<meta name="viewport"
      content="width=device-width, initial-scale=1, shrink-to-fit=no">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<link rel="shortcut icon" type="image/x-icon" href="assets/img/leaf.svg">
<title>Avni - Bootstrap 5 Admin Template</title>
<link href="assets/css/bootstrap.css" rel="stylesheet">
<link href="assets/css/main.css" rel="stylesheet">
<link href="assets/css/services.css" rel="stylesheet">

</head>

<body class="d-flex flex-column h-100">
  <div id="page">

    <div class="wrapper">
```

```
<nav id="sidebar" class="active">

    <div class="sidebar-header text-center">
        
        <h4 class="sidebar-title theme-item">NS</h4>
    </div>

    <ul class="list-unstyled components text-secondary">
        <li><a href="index.html"><i class="data-feather theme-item" data-feather="home"></i> <span class="theme-item"> Dashboard</span></a></li>

        <li><a href="providers.html"><i class="data-feather theme-item" data-feather="users"></i> <span class="theme-item"> Providers</span></a></li>
        <li><a href="Services.html"><i class="data-feather theme-item" data-feather="grid"></i> <span class="theme-item"> Services</span></a></li>
        <li><a href="users.html"><i class="data-feather theme-item" data-feather="user"></i> <span class="theme-item"> Users</span></a></li>
        <li><a href="AddService.html"><i class="data-feather theme-item" data-feather="settings"></i> <span class="theme-item">Add Service</span></a></li>
    </ul>
</nav>
```

```
<li><a href="addproviders.html"><i  
class="data-feather theme-item" data-  
feather="plus"></i> <span  
class="data-feather theme-item"> Add  
Providers</span></a></li>  
  
</ul>  
  
</nav>  
  
<div id="bodywrapper" class="container-fluid showhidetoggle">  
  
<nav class="navbar navbar-expand-md navbar-white bg-white py-0"  
aria-label="navbarexample" id="navbar">  
  <div class="container-fluid">  
    <button type="button" id="sidebarCollapse"  
           class="btn btn-light py-0">  
      <i data-feather="menu"></i> <span></span>  
    </button>  
      
    <h4 class="sidebar-title theme-item mt-2  
navbrandarea2">NS</h4>  
    <button class="navbar-toggler py-0" type="button"  
           data-bs-toggle="collapse" data-bs-  
target="#navbarsExample04"  
           aria-controls="navbarsExample04" aria-  
expanded="false"  
           aria-label="Toggle navigation">  
      <span class="navbar-toggler-icon"><i data-  
feather="menu"></i></span>  
    </button>
```

```

        <div class="collapse navbar-collapse mx-1"
id="navbarsExample04">

            <ul class="navbar-nav me-auto mb-2 mb-lg-0">

                <li class="nav-item"><div class="nav-dropdown">
                    <a class="nav-item nav-link active text-
secondary py-0"
href="index.html"><i
                    class="data-feather theme-item" data-
feather="home"></i> <span
                    class="theme-item">Home </span></a>
                </div></li>

            </ul>

            <div class="usermenu">
                <div class="nav-dropdown py-0">
                    <a href="#" class="nav-item nav-link dropdown-toggle
text-secondary py-0"
                        id="navbarDropdown3" role="button" data-
bs-toggle="dropdown"
                        aria-expanded="false">  <!--<i class="theme-item" -->
                        <!--data-feather="user"--></i> --> <span
                    class="theme-item">Admin
                        </span><i class="theme-item" data-
feather="chevron-down"></i></a>
                </div>
            </div>
        </div>
    
```

```



    <!-- <li><a href="profile.html"
class="dropdown-item mt-2"><i
         class="data-feather" data-
feather="user"></i> Profile</a></li> -->

    <li><a href="#" class="dropdown-item mt-
2"
         data-bs-toggle="modal" data-bs-
target="#settingsModal"><i
         class="data-feather" data-
feather="settings"></i> Settings</a></li>
    <!-- <li><a href="#" class="dropdown-item
mt-2"><i
         class="data-feather" data-
feather="log-out"></i> Logout</a></li> -->

```



```

    </div>

```

</div>

```

</div>

```

</div>

```

</nav>

<div class="settings">
    <div class="modal fade" id="settingsModal"
         aria-labelledby="settingsModalTitle" aria-hidden="true"
         tabindex="-1">
        <!--
             data-bs-backdrop="static" data-bs-
keyboard="false" -->
        <div class="modal-dialog modal-dialog-settings">

```

```
<div class="modal-content">

    <div class="modal-header text-center">
        <h5 class="modal-title">
id="exampleModalLabel">Settings</h5>

        <button type="button" class="btn-close" data-
bs-dismiss="modal"
aria-label="Close"></button>
    </div>

    <div class="modal-body">

        <section id="logincontent" class="shiftdown">

            <div class="row g-3 mb-3 mt-3">

                <div class="col-md-6">
                    <h6 class="text-muted">Select
Color</h6>

                    <span onclick="changeColor('0')"
class="btn btn-sm btn-primary
rounded-circle"><span
class="me-2"></span></span>
                    <span onclick="changeColor('1')"
class="btn btn-sm btn-success
rounded-circle"><span
class="me-2"></span></span>
                    <span onclick="changeColor('2')"
class="btn btn-sm btn-danger
rounded-circle"><span
class="me-2"></span></span>
                    <span onclick="changeColor('3')"
class="btn btn-sm btn-warning
rounded-circle"><span
```

```

></span></span>

><span
></span></span>
<div class="d-flex justify-content-between">
<button
onclick="removeColorCookie()">Reset to
Default</button>
</div>
</div>
<div class="col-md-6">
<h6 class="text-muted">Preferences</h6>
<div class="form-check form-switch">
<input class="form-check-input" type="checkbox"
> <label
for="flexSwitchCheckDefault">Switch
option 1</label>
</div>
<div class="form-check form-switch">
<input class="form-check-input" type="checkbox"
> <label
for="flexSwitchCheckChecked">Switch

```

```
option 2</label>
      </div>
    </div>
  </div>
<div class="row g-3 mb-3 mt-3">
  <div class="col-md-6">
    <div class="form-check">
      <input class="form-check-input" type="checkbox" value="" id="defaultCheck1" checked> <label class="form-check-label" for="defaultCheck1"> Checkbox 1 </label>
    </div>
    <div class="form-check">
      <input class="form-check-input" type="checkbox" value="" id="defaultCheck2" checked> <label class="form-check-label" for="defaultCheck2"> Checkbox 2 </label>
    </div>
  </div>
<div class="col-md-6">
  <h6 class="text-muted">View Size</h6>
  <div class="form-check">
    <input class="form-check-input" type="radio" checked> <label class="form-check-label" for="defaultRadio1"> View Size 1 </label>
  </div>
  <div class="form-check">
    <input class="form-check-input" type="radio" checked> <label class="form-check-label" for="defaultRadio2"> View Size 2 </label>
  </div>
</div>
```

```
name="flexRadioDefault"
id="radioCompactView"> <label
for="radioCompactView">
    Compact</label>
</div>
<div class="form-check">
    <input class="form-check-
input" type="radio"
name="flexRadioDefault"
id="radioFullView"> <label
for="radioFullView">
    Full-screen </label>
</div>
<div class="d-flex justify-
content-between">
    <button
        onclick="removeViewSizeCookie()">Reset to
        Default</button>
    </div>
    </div>
    </div>
    <hr />
    <button class="btn btn-sm btn-danger"
data-bs-dismiss="modal"
type="button">
        <i data-feather="check-circle"
class="mr-1"></i> Ok
    </button>
</section>
```

```
</div>
</div>
</div>
</div>
</div>

<div id="sidebarOverlayNav" class="screen-overlay float-end">
    <a href="javascript:void(0)" class="closebtn"
        onclick="closeOverlayNav()">&times;</a>
    <div class="screen-overlay-content text-secondary">
        <a href="#" class="active">About</a> <a
        href="#">Services</a> <a
        href="#">Clients</a> <a href="#">Contact</a>
    </div>
</div>

<div class="content">
    <div class="container-fluid">
        <div class="row mt-2">
            <div class="col-md-6 float-start">
                <h4 class="m-0 text-dark text-
muted">Services</h4>
            </div>
            <!-- <div class="col-md-6">
                <ol class="breadcrumb float-end">
                    <li class="breadcrumb-item"><a href="#">
Home</a></li>
```

```
<li class="breadcrumb-item active">Components</li>
</ol>
</div> -->

</div>

<table id="myTable">
<thead>
<tr>
<th>Id</th>
<th>Name</th>
<th>Name Ar</th>
<th>Image</th>
</tr>
</thead>
<tbody>

</tbody>
</table>

</div>

</div>

</div>
</div>
<footer class="footer container mt-auto py-1">
```

```
<div
    class="d-sm-flex justify-content-center justify-content-sm-between">
    <span
        class="text-muted d-block text-center text-sm-left d-sm-inline-block">
        Copyright © N-S <script>
            document.write(new Date().getFullYear())
        </script>
    </span> <span
        class="float-none text-muted float-sm-right d-block mt-1 mt-sm-0
text-center">
        N-S dashboard
    </span>
</div>
</footer>
<div id="loading" class="spinner-border text-primary align-middle"
role="status"></div>

<button class="btn btn-sm btn-primary rounded-circle"
    onclick="scrollToTopFunction()" id="scrollToTop" title="Scroll to top">
    <i data-feather="arrow-up-circle"></i>
</button>

<script src="assets/js/feather.min.js"></script>
<script src="assets/js/bootstrap.bundle.min.js"></script>
<script src="assets/js/script.js"></script>
<script type="text/javascript">
    document.addEventListener("DOMContentLoaded", function(event) {
        feather.replace();
    }
</script>
```

```
});  
</script>  
  
<script type="text/javascript">  
    var tooltipTriggerList = [].slice.call(document  
        .querySelectorAll('[data-toggle="tooltip"]'));  
  
    var tooltipList = tooltipTriggerList.map(function(tooltipTriggerEl) {  
        return new bootstrap.Tooltip(tooltipTriggerEl)  
    });  
</script>  
  
<script src="assets/js/services.js"></script>  
  
</body>  
  
</html>
```

5.2.3 Add Service

```
<!doctype html>  
  
<html lang="en" class="h-100">  
  
<head>  
    <meta charset="utf-8">  
    <meta name="viewport"  
        content="width=device-width, initial-scale=1, shrink-to-fit=no">  
    <meta http-equiv="X-UA-Compatible" content="IE=edge">  
    <link rel="shortcut icon" type="image/x-icon" href="assets/img/leaf.svg">  
    <title>Avni - Bootstrap 5 Admin Template</title>  
    <link href="assets/css/bootstrap.css" rel="stylesheet">  
    <link href="assets/css/main.css" rel="stylesheet">  
</head>
```

```
<body class="d-flex flex-column h-100">
  <div id="page">

    <div class="wrapper">

      <nav id="sidebar" class="active">

        <div class="sidebar-header text-center">
          
          <h4 class="sidebar-title theme-item">NS</h4>
        </div>

        <ul class="list-unstyled components text-secondary">
          <li><a href="index.html"><i class="data-feather theme-item" data-feather="home"></i> <span class="theme-item"> Dashboard</span></a></li>

          <li><a href="providers.html"><i class="data-feather theme-item" data-feather="users"></i> <span class="theme-item"> Providers</span></a></li>

          <li><a href="Services.html"><i class="data-feather theme-item" data-feather="grid"></i> <span class="theme-item"> Services</span></a></li>

          <li><a href="users.html"><i class="data-feather theme-item" data-feather="user"></i> <span
```

```

        class="theme-item"> Users</span></a></li>

    <li><a href="AddService.html"><i class="data-feather theme-
item" data-feather="settings"></i> <span class="theme-
item">Add Service</span></a></li>

    <li><a href="addproviders.html"><i
            class="data-feather theme-item" data-
feather="plus"></i> <span
            class="data-feather theme-item"> Add
        Providers</span></a></li>

    </ul>

</nav>

<div id="bodywrapper" class="container-fluid showhidetoggle">

    <nav class="navbar navbar-expand-md navbar-white bg-white py-0"
        aria-label="navbarexample" id="navbar">
        <div class="container-fluid">
            <button type="button" id="sidebarCollapse"
                    class="btn btn-light py-0">
                <i data-feather="menu"></i> <span></span>
            </button>
            
            <h4 class="sidebar-title theme-item mt-2
navbrandarea2">NS</h4>
            <button class="navbar-toggler py-0" type="button"
                    data-bs-toggle="collapse" data-bs-
target="#navbarsExample04">

```

```
        aria-controls="navbarsExample04" aria-
expanded="false"

        aria-label="Toggle navigation">
        <span class="navbar-toggler-icon"><i data-
feather="menu"></i></span>

    </button>

<div class="collapse navbar-collapse mx-1"
id="navbarsExample04">
    <ul class="navbar-nav me-auto mb-2 mb-lg-0">

        <li class="nav-item"><div class="nav-dropdown">
            <a class="nav-item nav-link active text-
secondary py-0"
href="index.html"><i
            class="data-feather theme-item" data-
feather="home"></i> <span
            class="theme-item">Home </span></a>
        </div></li>

    </ul>

    <div class="usermenu">
        <div class="nav-dropdown py-0">
            <a href="#"
                class="nav-item nav-link dropdown-toggle
text-secondary py-0"
                id="navbarDropdown3" role="button" data-
bs-toggle="dropdown">
```

```
        aria-expanded="false"> <img class="theme-item user-avatar"

image"> <!--<i class="theme-item" -->
<!--data-feather="user"--></i> --> <span
class="theme-item">Admin
</span><i class="theme-item" data-
feather="chevron-down"></i></a>

<ul class="dropdown-menu dropdown-menu-end"
aria-labelledby="navbarDropdown3">
<!-- <li><a href="profile.html"
class="dropdown-item mt-2"><i
class="data-feather" data-
feather="user"></i> Profile</a></li> -->

<li><a href="#" class="dropdown-item mt-
2"
data-bs-toggle="modal" data-bs-
target="#settingsModal"><i
class="data-feather" data-
feather="settings"></i> Settings</a></li>
<!-- <li><a href="#" class="dropdown-item mt-
2"><i
class="data-feather" data-
feather="log-out"></i> Logout</a></li> -->
</ul>
</div>
</div>

</div>
</div>
</nav>
```

```
<div class="settings">
    <div class="modal fade" id="settingsModal"
        aria-labelledby="settingsModalTitle" aria-hidden="true"
        tabindex="-1">
        <!-- data-bs-backdrop="static" data-bs-
keyboard="false" -->

        <div class="modal-dialog modal-dialog-settings">
            <div class="modal-content">
                <div class="modal-header text-center">
                    <h5 class="modal-title"
id="exampleModalLabel">Settings</h5>
                    <button type="button" class="btn-close" data-
bs-dismiss="modal"
                        aria-label="Close"></button>
                </div>
                <div class="modal-body">

                    <section id="logincontent" class="shiftdown">

                        <div class="row g-3 mb-3 mt-3">

                            <div class="col-md-6">
                                <h6 class="text-muted">Select
Color</h6>
                                <span onclick="changeColor('0')"
class="btn btn-sm btn-primary
rounded-circle"><span
class="me-2"></span></span>
                                <span onclick="changeColor('1')"
class="btn btn-sm btn-success
rounded-circle"><span
```

```

></span></span>
<span onclick="changeColor('2')"

rounded-circle"><span

></span></span>
<span onclick="changeColor('3')"

rounded-circle"><span

></span></span>
<span onclick="changeColor('4')"

secondary rounded-circle"><span

></span></span>
<div class="d-flex justify-
content-between">

<button

onclick="removeColorCookie()">Reset to

Default</button>

</div>
</div>
<div class="col-md-6">

<h6 class="text-
muted">Preferences</h6>

<div class="form-check form-
switch">

<input class="form-check-
input" type="checkbox"

id="flexSwitchCheckDefault"class="form-check-label"

for="flexSwitchCheckDefault">Switch

option 1</label>

</div>

```

```
<div class="form-check form-switch">
    <input class="form-check-input" type="checkbox" id="flexSwitchCheckChecked" checked> <label class="form-check-label" for="flexSwitchCheckChecked">Switch option 2</label>
</div>
</div>
</div>
<div class="row g-3 mb-3 mt-3">
    <div class="col-md-6">
        <div class="form-check">
            <input class="form-check-input" type="checkbox" value="" id="defaultCheck1" checked> <label class="form-check-label" for="defaultCheck1"> Checkbox 1 </label>
        </div>
        <div class="form-check">
            <input class="form-check-input" type="checkbox" value="" id="defaultCheck2"> <label class="form-check-label" for="defaultCheck2"> Checkbox 2 </label>
        </div>
    </div>
</div>
```

```
</div>

<div class="col-md-6">
    <h6 class="text-muted">View
Size</h6>

    <div class="form-check">
        <input class="form-check-
input" type="radio"
name="flexRadioDefault"
id="radioCompactView"> <label
for="radioCompactView">
    Compact</label>
</div>

    <div class="form-check">
        <input class="form-check-
input" type="radio"
name="flexRadioDefault"
id="radioFullView"> <label
for="radioFullView">
    Full-screen </label>
</div>

    <div class="d-flex justify-
content-between">
        <button
onclick="removeViewSizeCookie()">Reset to
Default</button>
        </div>
    </div>
</div>
<hr />
```

```
<button class="btn btn-sm btn-danger"
data-bs-dismiss="modal"
type="button">
<i data-feather="check-circle" class="mr-1"></i> Ok
</button>
</section>
</div>
</div>
</div>
</div>
</div>

<div id="sidebarOverlayNav" class="screen-overlay float-end">
<a href="javascript:void(0)" class="closebtn"
onclick="closeOverlayNav()">&times;</a>
<div class="screen-overlay-content text-secondary">
<a href="#" class="active">About</a> <a href="#">Services</a> <a href="#">Clients</a> <a href="#">Contact</a>
</div>
</div>

<div class="content">
<div class="container-fluid">
<div class="row mt-2">
<div class="col-md-6 float-start">
```

```
<h4 class="m-0 text-dark text-muted">Add a  
service</h4>  
  
</div>  
  
</div>  
  
  
<br>  
  
<div class="serviceform">  
  
<form action="">  
  
<h1 class="serviceHeader text-dark text-  
muted">Service Info</h1>  
  
  
  
  
  
  
<div class="serviceforminputs">  
  
<input type="text" class="mr1"  
placeholder="service name in arabic">  
  
<input type="text" placeholder="service name  
in english">  
  
</div>  
  
<div class="serviceimage">  
  
Push an image of the service: &nbsp;  
  
<input type="file">  
  
</div>  
  
<input type="submit" value="Add Service"  
class="addservice">  
  
</form>  
  
</div>  
  
  
</div>  
  
</div>
```

```
</div>

</div>
</div>

<footer class="footer container mt-auto py-1">
  <div
    class="d-sm-flex justify-content-center justify-content-sm-between">
    <span
      class="text-muted d-block text-center text-sm-left d-sm-inline-block">
      Copyright © N-S <script>
        document.write(new Date().getFullYear())
      </script>
    </span> <span
      class="float-none text-muted float-sm-right d-block mt-1 mt-sm-0
text-center">
      N-S dashboard
    </span>
  </div>
</footer>

<div id="loading" class="spinner-border text-primary align-middle"
role="status"></div>

<button class="btn btn-sm btn-primary rounded-circle"
  onclick="scrollToTopFunction()" id="scrollToTop" title="Scroll to top">
  <i data-feather="arrow-up-circle"></i>
</button>
```

```
<script  
    src="https://cdnjs.cloudflare.com/ajax/libs/feather-  
icons/4.28.0/feather.min.js"></script>  
  
<script src="assets/js/bootstrap.bundle.min.js"></script>  
  
<script src="assets/js/script.js"></script>  
  
<script type="text/javascript">  
  
    document.addEventListener("DOMContentLoaded", function(event) {  
  
        feather.replace();  
  
    });  
  
</script>  
  
<script type="text/javascript">  
  
    (function() {  
  
        'use strict'  
  
  
        // Fetch all the forms we want to apply custom Bootstrap validation  
        styles to  
  
        var forms = document.querySelectorAll('.needs-validation')  
  
  
        // Loop over them and prevent submission  
        Array.prototype.slice.call(forms).forEach(function(form) {  
  
            form.addEventListener('submit', function(event) {  
  
                if (!form.checkValidity()) {  
  
                    event.preventDefault()  
  
                    event.stopPropagation()  
  
                }  
  
                form.classList.add('was-validated')  
  
            }, false)  
        })  
    })
```

```

    }))()

</script>

</body>

</html>

```

5.3 Backend Code (.net):

```

UserController.cs
1  // PUT: api/User/
2  [HttpPut]
3  public void Put(int id, [FromBody]string value)
4  {
5  }
6
7  // DELETE: api/User/{id}
8  [HttpDelete]
9  [AllowAnonymous]
10 public void Delete(int id)
11 {
12     var item = ctx.TbUsers.Where(a => a.CustomerId == id).FirstOrDefault();
13     ctx.TbUsers.Remove(item);
14     ctx.SaveChanges();
15 }
16
17 [HttpDelete]
18
19 [Route("api/User/DeleteAll")]
20 [AllowAnonymous]
21 public void DeleteAll()
22 {
23     var allUsers = ctx.TbUsers.ToList();
24     ctx.TbUsers.RemoveRange(allUsers);
25     ctx.SaveChanges();
26 }
27
28 }
29

```

Figure 5-1

```

ServicesController.cs
1  // GET: api/Services
2  [HttpGet]
3  public IEnumerable<Service> Get()
4  {
5      ServiceEntities ctx = new ServiceEntities();
6      var services = ctx.Services.ToList();
7      return services;
8  }
9
10 // GET: api/Services/{id}
11 [HttpGet("{id}")]
12 public Service Get(int id)
13 {
14     ServiceEntities ctx = new ServiceEntities();
15     var service = ctx.Services.Where(a => a.ServiceId == id).FirstOrDefault();
16     return service;
17 }
18
19 // POST: api/Services
20 [HttpPost]
21 public string Post([FromBody]Service value)
22 {
23     try
24     {
25         ServiceEntities ctx = new ServiceEntities();
26
27         ctx.Services.Add(value);
28         ctx.SaveChanges();
29         return "done";
30     }
31     catch (Exception ex)
32     {
33         return ex.Message;
34     }
35 }
36
37
38 // PUT: api/Services/{id}
39 [HttpPut("{id}")]
40 public void Put(int id, [FromBody]string value)
41 {
42 }
43
44 // DELETE: api/Services/{id}
45 [HttpDelete("{id}")]
46 public void Delete(int id)
47 {
48 }
49

```

Figure 5-2

INSERVICE

```

Controller.cs      styles.xml      RequestProviderController.cs      LoginPage.xaml      RequestController.cs      ServiceProvidersController.cs * X
webapp

// GET: api/ServiceProviders
[HttpGet]
public IEnumerable<ServiceProviders> Get(int id)
{
    try
    {
        ServiceEntities ctx = new ServiceEntities();
        return ctx.ServiceProviders.Where(a => a.ProviderId == id).ToList();
    }
    catch
    {
        return new List<ServiceProviders>();
    }
}

[HttpGet]
[Route("api/serviceprovider/{providerid}")]
public IEnumerable<ServiceProviders> Get(int providerid)
{
    try
    {
        ServiceEntities ctx = new ServiceEntities();
        return ctx.ServiceProviders.Where(a => a.ProviderId == providerid).ToList();
    }
    catch
    {
        return new List<ServiceProviders>();
    }
}

// POST: api/ServiceProviders
[HttpPost]
public string Post([FromBody]IServiceProvider value)
{
    try
    {
        ServiceEntities ctx = new ServiceEntities();

        ctx.tbServiceProviders.Add(value);
        ctx.SaveChanges();
        return "Done";
    }
    catch (Exception ex)
    {
        return ex.Message;
    }
}

// PUT: api/ServiceProviders
[HttpPut]
[Route("api/serviceprovider/{providerid}")]
public void Put(int providerid, [FromBody] IServiceProvider value)
{
    try
    {
        ServiceEntities ctx = new ServiceEntities();
        var provider = ctx.tbProviders.FirstOrDefault(p => p.ProviderId == providerid);
    }
    catch (Exception ex)
    {
        return ex.Message;
    }
}

```

Figure 5-3

```

Controller.cs      Splash.cs      styles.xml      RequestProviderController.cs * X      ServicesController.cs      LoginPage.xaml      RequestController.cs
webapp

        catch(Exception ex)
        {
            return ex.Message;
        }
    }

    // PUT: api/Request/S
    // [HttpPut]
    // [Route("api/Request/{id}")]
    [HttpPut("{id}")]
    public void Put(int id, [FromBody] TbOFRRequest value)
    {
        // your existing logic here
        var dbItem = ctx.TbOFRRequests.Where(a => a.ID == id).FirstOrDefault();
        dbItem.StatusOFRRequest = value.StatusOFRRequest;
        dbItem.IsResponse = value.IsResponse;
        dbItem.DateValueCost = value.DateValue;
        dbItem.Date = value.Date;
        ctx.SaveChanges();
    }

    // DELETE: api/Request/S
    [HttpDelete]
    [AllowAnonymous]
    public void Delete(int id)
    {
        var item = ctx.TbOFRRequests.Where(a => a.ID == id).FirstOrDefault();
        ctx.TbOFRRequests.Remove(item);
        ctx.SaveChanges();
    }
}

```

Figure 5-4

```

rateController.cs Splash.cs styles.xml RequestProviderController.cs ServicesController.cs LoginPage.xaml RequestController.cs
webapp RequestController.cs
18     public IEnumerable<TBRequest> Get()
19     {
20         var services = ctx.TBRequests.ToList();
21         return services;
22     }
23
24     // GET: api/Request/{id}
25     public IEnumerable<TBRequest> Get(int id)
26     {
27         var items = ctx.TBRequests.Where(item => item.CustomerId == id).ToList();
28         return items;
29     }
30
31     // POST: api/Request
32     public string Post([FromBody] TBRequest value)
33     {
34         try
35         {
36             // TBRequestProvider value2 = new TBRequestProvider();
37             // value2.ProviderId = value.ProviderId;
38             ctx.TBRequests.Add(value);
39             ctx.TBRequestProviders.Add(value);
40             ctx.SaveChanges();
41             return "done";
42         }
43         catch (Exception ex)
44         {
45             return ex.Message;
46         }
47     }
48
49     // PUT: api/Request/{id}
50     public void Put(int id, [FromBody]string value)
51     {
52     }
53
54     // DELETE: api/Request/{id}
55     [HttpDelete]
56     [AllowAnonymous]
57     public void Delete(int id)
58     {
59         var item = ctx.TBRequests.Where(a => a.ID == id).FirstOrDefault();
60         ctx.TBRequests.Remove(item);
61         ctx.SaveChanges();
62     }
63
64 }
```

Figure 5-5

```

rateController.cs otp.Android Splash.cs styles.xml RequestProviderController.cs ServicesController.cs LoginPage.xaml
webapp rateController.cs
11     ServicesEntities ctx = new ServicesEntities();
12
13     // GET: api/rate/provider{id}
14     [Route("api/rate/{providerId}")]
15     [HttpGet]
16     public IHttpActionResult Get(int providerId)
17     {
18         try
19         {
20             // calculate the average rating for the specified provider
21             double averageRating = ctx.TBRatings.Any(r => r.ProviderId == providerId) ?
22                 ctx.TBRatings.Where(r => r.ProviderId == providerId).Average(r => r.Rating) :
23                 null;
24
25             var provider = ctx.TBProviders.FirstOrDefault(p => p.ProviderId == providerId);
26             if (provider != null)
27             {
28                 provider.Rating = averageRating;
29                 ctx.SaveChanges();
30             }
31
32             // return the average rating
33             return Ok(averageRating);
34         }
35         catch (Exception ex)
36         {
37             // handle exceptions
38             return InternalServerError(ex);
39         }
40     }
41
42
43     // GET: api/rate/
44
45     // POST: api/rate
46     public string Post([FromBody]TBRating value)
47     {
48         try
49         {
50
51             ctx.TBRatings.Add(value);
52             ctx.SaveChanges();
53             return "done";
54         }
55         catch (Exception ex)
56         {
57
58         }
59     }
60 }
```

Figure 5-6

```
otp.Android Splash.cs styles.xml RequestProviderController.cs ServicesController.cs LoginPage.xaml ProviderController.cs Get()

4  using System;
5  using System.Net.Http;
6  using System.Web.Http;
7  using webapp.Models;
8
9  namespace webapp.Controllers
10 {
11     public class ProviderController : ApiController
12     {
13         // GET: api/User
14         ServicesEntities ctx = new ServicesEntities();
15         public IEnumerable<TbProvider> Get()
16         {
17             var users = ctx.TbProviders.ToList();
18             return users;
19         }
20
21         // GET: api/User/{id}
22         public IEnumerable<TableOfProviders> Get(int id)
23         {
24             var items = ctx.TableOfProviders.Where(item => item.providerid == id).ToList();
25             return items;
26         }
27
28         // POST: api/User
29         public string Post([FromBody] TbUser value)
30         {
31             try
32             {
33                 ctx.TbUsers.Add(value);
34                 ctx.SaveChanges();
35                 return "done";
36             }
37             catch (Exception ex)
38             {
39                 return ex.Message;
40             }
41         }
42
43         // PUT: api/User/{id}
44
45
46 }
```

Figure 5-7

Resources

- 1- https://www.tutorialspoint.com/system_analysis_and_design/system_analysis_and_design_overview.htm , Thursday, December 7, 2023, 8:01:56 PM.
- 2- <https://www.geeksforgeeks.org/what-is-dfddata-flow-diagram> , Thursday, December 7, 2023, 8:01:56 PM.
- 3- <https://slideplayer.com/user/account/feedback/> , February 4, 2024, 12:35:6 AM.
- 4- <https://www.visual-paradigm.com/guide/data-modeling/what-is-entity-relationship-diagram/> , December 7, 2023, 8:01:56 PM.
- 5- <https://www.youtube.com/watch?v=kZspaeZ3o7Y>, عظيم حياة, Monday, December 18, 2023, 12:12:22 PM.
- 6- <https://www.youtube.com/watch?v=ARDFI9d5e1k>, عظيم حياة, Monday, December 18, 2023, 12:12:22 PM.
- 7- <https://www.youtube.com/watch?v=-IhKHP-oc8k>, عظيم حياة Saturday, January 27, 2024, 12:57:24 PM.

الملخص

في عصر يتسم بالتحضر وكثرة أنماط الحياة، أصبح الطلب على الخدمات المنزلية التي يمكن الوصول إليها حسب الطلب داخل أحياطنا وأضحاً بشكل متزايد.

يحكي هذا المشروع عن تطبيق يهدف إلى إحداث ثورة في الطريقة التي نتعامل بها مع الخدمات المنزلية الأساسية في مجتمعاتنا المحلية .

حيث يعد هذا المشروع مبادرة رائعة تجمع مجموعة متنوعة من الخدمات المنزلية ضمن منصة واحدة سهلة الاستخدام، تتكون هذه الخدمات من (السباكية، الكهرباء ، الطلاء، تصليح الأجهزة الكهربائية، التنظيف، وغيرها من الخدمات المنزلية الأساسية).

ومن خلال تسخير قوة التكنولوجيا هدفنا هو تمكين الأفراد من تلبية احتياجاتهم المنزلية بكفاءة.

يستكشف هذا التقرير الميزات والوظائف لهذا التطبيق المبتكر، مما يوضح كيفية إعادة تشكيل مشهد خدمات الأحياء وتعزيز الرفاهية العامة للمقيمين.

فريق المشروع

1. أحمد محمد إبراهيم

2 - إسلام محمد علي

3. زياد السيد السمان

4 - محمود عبد اللاه محمد

5 - مريم عبد الراضي حشمت

6. ندى مصطفى أمين

7. هاجر ياسر عبد الجابر