



AN-NAJAH NATIONAL UNIVERSITY  
FACULTY OF ENGINEERING & INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER ENGINEERING

# Growify

Amplifies the Growth for both companies/organizations and  
individuals

**Done by**  
Ahmad Othman, Obaida Aws

**Supervisor**  
Dr.Sufyan Samara

Submitted in partial fulfilment of the requirements for the Bachelor degree  
in Computer Engineering  
January 29, 2024

# Acknowledgements

First of all, we would like to thank our supervisor, Dr. Sufyan Samara, for his continuous efforts throughout the semester and for making everything simple. He always inspired us and encouraged us to get moving. We would also like to thank our parents, families and friends. Those who gave us their full support and believed in our ability. Finally, we would like to express our deep appreciation to all who have contributed in any way and provided invaluable assistance. Your support means a lot to us.



# Disclaimer

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

# Contents

<b>1</b>	<b>Abstract</b>	<b>5</b>
<b>2</b>	<b>Introduction</b>	<b>7</b>
2.1	Problem Statement . . . . .	7
2.2	Objectives . . . . .	7
2.3	Scope of the work . . . . .	7
2.4	Importance . . . . .	7
2.5	Report Organization . . . . .	8
<b>3</b>	<b>Constraints and Earlier Coursework</b>	<b>9</b>
3.1	Constraints & Limitations . . . . .	9
3.2	Standards . . . . .	11
3.3	Earlier Coursework . . . . .	12
<b>4</b>	<b>Literature Review</b>	<b>13</b>
<b>5</b>	<b>Methodology</b>	<b>14</b>
5.1	Tools and services . . . . .	14
5.2	Database Design and Configuration . . . . .	14
5.2.1	MySQL . . . . .	14
5.3	Front-end . . . . .	16
5.3.1	Flutter . . . . .	16
5.4	Back-end Node Js . . . . .	16
5.4.1	Architecture style . . . . .	17
5.4.2	JWT . . . . .	17
5.4.3	Sequelize . . . . .	18
5.4.4	nodemailer . . . . .	18
5.4.5	axios . . . . .	18
5.4.6	Server-Sent Events . . . . .	18
5.4.7	Socket io . . . . .	19
5.4.8	WebRTC . . . . .	20
5.5	Growify as Web application & Mobile application . . . . .	24
5.5.1	Sign Up . . . . .	24
5.5.2	Forgot password . . . . .	26
5.5.3	Login . . . . .	28
5.5.4	Home Screen . . . . .	29
5.5.5	User Settings . . . . .	30
5.5.6	User Calendar . . . . .	34

5.5.7	User Tasks . . . . .	36
5.5.8	User Profile . . . . .	37
5.5.9	Posts . . . . .	40
5.5.10	Other users profiles . . . . .	45
5.5.11	Chat . . . . .	48
5.5.12	User pages . . . . .	53
5.5.13	Page groups . . . . .	63
5.5.14	Group meeting . . . . .	73
5.5.15	Networks . . . . .	78
5.5.16	Jobs . . . . .	82
5.5.17	Search . . . . .	84
5.5.18	Other Pages . . . . .	85
5.5.19	Notifications . . . . .	85
5.5.20	Admin panel . . . . .	86
<b>6</b>	<b>Results &amp; Discussion</b>	<b>105</b>
<b>7</b>	<b>Conclusion &amp; Future Work</b>	<b>107</b>
7.1	Conclusion . . . . .	107
7.2	Future Work . . . . .	107

# List of Figures

3.1	Applied Agile model. . . . .	11
5.1	Growify MySQL Database . . . . .	15
5.2	Our Project Model . . . . .	17
5.3	Server-Sent Events . . . . .	18
5.4	WebSockets . . . . .	19
5.5	WebSockets Vs HTTP . . . . .	20
5.6	WebRTC Signalling Process . . . . .	21
5.7	WebRTC architectures . . . . .	23
5.8	Sign Up . . . . .	24
5.9	code Sent to user email . . . . .	24
5.10	verification code . . . . .	25
5.11	Sign up attempts exceeded . . . . .	25
5.12	Sign up successful . . . . .	26
5.13	Forgot Password . . . . .	26
5.14	Reset password verification code . . . . .	27
5.15	change Password attempts . . . . .	27
5.16	Reset Password . . . . .	28
5.17	Login . . . . .	28
5.18	Home Screen . . . . .	29
5.19	User Settings . . . . .	30
5.20	Profile Settings . . . . .	30
5.21	user Fields . . . . .	31
5.22	Change Password . . . . .	31
5.23	Change Email . . . . .	32
5.24	verifying change email . . . . .	32
5.25	Work Experience . . . . .	33
5.26	Education Level . . . . .	33
5.27	Delete User Account . . . . .	34
5.28	User Calendar . . . . .	34
5.29	Add User Event . . . . .	35
5.30	Set Reminder Date And Time . . . . .	35
5.31	Event Notification . . . . .	35
5.32	User Tasks . . . . .	36
5.33	User Task View . . . . .	36
5.34	Tasks Status . . . . .	36
5.35	Add New Task . . . . .	37
5.36	Tasks Notification . . . . .	37

5.37	Main User Profile Button . . . . .	38
5.38	User Profile . . . . .	38
5.39	User Profile See About Info . . . . .	39
5.40	Add New Post . . . . .	39
5.41	Post in all possible body designs . . . . .	40
5.42	Post multiple images and videos . . . . .	41
5.43	Post Options . . . . .	41
5.44	Edit Post . . . . .	42
5.45	Post Edit History . . . . .	42
5.46	Report Post . . . . .	43
5.47	Post comments . . . . .	43
5.48	Comment Options . . . . .	44
5.49	Comment Report . . . . .	44
5.50	Post Likes . . . . .	45
5.51	Other Users Profile . . . . .	45
5.52	connect . . . . .	46
5.53	Accept / Delete connection request . . . . .	46
5.54	Requested . . . . .	47
5.55	Connected . . . . .	47
5.56	Report User . . . . .	48
5.57	Other users see about info . . . . .	48
5.58	User Chats . . . . .	49
5.59	Users Chat messages . . . . .	49
5.60	Users Audio Call Request . . . . .	50
5.61	Users Call Accepted . . . . .	50
5.62	Users Video Call Request . . . . .	51
5.63	Users Video Call Accepted . . . . .	51
5.64	Video Call Share Screen . . . . .	52
5.65	Call Process . . . . .	52
5.66	Other Users Profile . . . . .	53
5.67	create new page . . . . .	53
5.68	Public Page . . . . .	54
5.69	Private Page . . . . .	55
5.70	Page Admins . . . . .	55
5.71	Add New Admin . . . . .	56
5.72	Admin View . . . . .	56
5.73	Publisher View . . . . .	57
5.74	Page Employees . . . . .	57
5.75	Add New Employee . . . . .	58
5.76	Edit Page Profile . . . . .	58
5.77	Page Jobs . . . . .	59
5.78	Add New Job . . . . .	59
5.79	Job Fields . . . . .	59
5.80	Job Applications . . . . .	60
5.81	Page - User Chat . . . . .	60
5.82	Page - User Call . . . . .	61
5.83	Page Chats . . . . .	61
5.84	Page Calendar . . . . .	62

5.85 Add Event To Page Calendar . . . . .	62
5.86 Page Event Added . . . . .	62
5.87 Main Page Groups . . . . .	63
5.88 Show Child Groups . . . . .	64
5.89 Show Child Groups . . . . .	64
5.90 Admin Group Chat . . . . .	65
5.91 Group Settings . . . . .	65
5.92 Delete Group . . . . .	65
5.93 Edit Group Info . . . . .	66
5.94 Show Group Admins . . . . .	66
5.95 Add New Group Admin . . . . .	67
5.96 Show Group Members . . . . .	67
5.97 Add New Group member . . . . .	68
5.98 Add Member From Page Employees . . . . .	68
5.99 Add External Group Member . . . . .	68
5.100Group Tasks . . . . .	69
5.101Add Group Task . . . . .	69
5.102Select Task Members . . . . .	70
5.103Change Group Task Status . . . . .	70
5.104Group Calendar . . . . .	71
5.105Group Messages . . . . .	71
5.106A proposal to divide groups for a university . . . . .	72
5.107A proposal to divide groups for a Company . . . . .	73
5.108Group Meeting . . . . .	74
5.109Admin Creates Meeting . . . . .	74
5.110Members Join Meeting . . . . .	74
5.111Meeting Members Seeing Each Other . . . . .	75
5.112One Member Share Screen From Mobile . . . . .	75
5.113Two Members Share Screen From Mobile And Web . . . . .	76
5.114Meeting History . . . . .	76
5.115Meeting Process - Admin Starts A Meeting . . . . .	77
5.116Meeting Process - User X Joined . . . . .	77
5.117Meeting Process - Two User Listening To Each Other . . . . .	77
5.118Meeting Process - User Y Joined . . . . .	78
5.119Meeting Process - All User Listening To Each Other . . . . .	78
5.120User Networks . . . . .	79
5.121User Connections . . . . .	79
5.122Requests Received . . . . .	79
5.123Requests Sent . . . . .	80
5.124Pages User Follows . . . . .	80
5.125Pages User Employee In . . . . .	80
5.126User Job Applications . . . . .	81
5.127Groups User Is Admin Or Member In . . . . .	81
5.128If Member Cannot send messages From Group Settings . . . . .	82
5.129Group Member Tasks And Change Status . . . . .	82
5.130User Jobs . . . . .	83
5.131See Job Details And Apply . . . . .	83
5.132The time to apply for job has ended . . . . .	84

5.133Search Filtration . . . . .	84
5.134Search Results . . . . .	85
5.135Others Page . . . . .	85
5.136User Notification . . . . .	86
5.137Notification process . . . . .	86
5.138Admin Login Page . . . . .	87
5.139Dashboard table . . . . .	87
5.140Users table . . . . .	88
5.141Users table . . . . .	88
5.142Users table . . . . .	89
5.143Add New User . . . . .	89
5.144Update User . . . . .	90
5.145User Connection . . . . .	90
5.146Sent Connection table . . . . .	91
5.147Table to display User Posts . . . . .	91
5.148educational level table . . . . .	92
5.149work experience table . . . . .	92
5.150User Applications table . . . . .	93
5.151Post Comments for each post . . . . .	93
5.152Likes for each User Posts . . . . .	94
5.153Get All Posts . . . . .	94
5.154Show All Fields . . . . .	95
5.155Add Field . . . . .	95
5.156Show All Pages . . . . .	96
5.157Show All Pages . . . . .	96
5.158Show All Jobs for page . . . . .	97
5.159Show All Follower for the page . . . . .	97
5.160show page Admins . . . . .	98
5.161show All Employee page . . . . .	98
5.162show All Events for the page . . . . .	98
5.163TempUser table . . . . .	99
5.164Active Users . . . . .	99
5.165Get All Comments . . . . .	100
5.166Get All Likes . . . . .	100
5.167Show All Jobs . . . . .	101
5.168Reported Comment . . . . .	101
5.169Reported Users . . . . .	102
5.170Reported Pages . . . . .	102
5.171Reported Posts . . . . .	103
5.172Show All Groups . . . . .	103
5.173Show All Groups . . . . .	104

# Chapter 1

## Abstract

Our graduation project aims to develop an innovative application that seamlessly combines the main functions of social media applications, organizing personal tasks, structuring companies/organizations, and communicating in one platform. This dynamic hub revolutionizes the way individuals and businesses communicate, collaborate and manage projects, amplifying growth for both companies/organizations and individuals.

Many functionalities are provided through our app/website:

- Professional profiles: Users can create comprehensive profiles, including personal information and a CV, and facilitate access to individuals, view their information, and communicate with them.
- Social networks: The platform allows sharing updates, publications, and communications with peers and professionals, by providing the personal profiles that can be easily accessed by searching, then viewing the user's publications and information, and communicate with the user through a wide number of communication channels we provide and also sending a connection request to add the user to the list of colleagues.
- Real-time communication: Powerful messaging features enable seamless text, audio and video conversations with individuals/pages/group.
- Task management: Integrated tools allow organizing tasks, tracking progress, and adding events to a calendar for individuals/pages/groups, with the ability to be reminded with a notification at a date specified by the user.
- Companies and organizations: Our application allows the creation of pages representing companies and organizations with a complete structure to organize their work, in addition to providing access to these pages, displaying main information, publications, public calendar, available jobs, starting conversations, and following these pages with notifications when a new job is added by the page.
- Structuring companies/organizations after creating the page:
  - Pages can add admins and employees with custom fields
  - After publishing the page for a new job, the page can see the application that



the individuals have submitted, as it contains the user's CV (optional), in addition to a note that the user can write (optional), in addition to access to the applicant's personal profile, which contains all contact information, education, Work, and updateable CV (even after applying for any job) to sync user experiences

- The page can communicate directly with the job applicant to schedule an interview, as the interview takes place on the application also by providing the video calling and screen sharing feature.
- The page can create groups represented as tree (Parent Group can have child groups, etc.) and distribute employees to groups, in addition to the possibility of adding users who are not employees of the company.
- Each group contains a number of features, the most important of which are:
  - \* Adding admins who can control group settings, add new members, determine members permissions, add tasks/events, start a group meeting, and create a child group for the current group.
  - \* A group calendar containing events
  - \* A special page for distributing tasks to group members, with the ability to change the task status (To Do, Doing, Done, Archive, Delete).
  - \* Chat for communication so that texts, pictures, and videos can be sent
  - \* Video/audio meeting with screen sharing feature

# Chapter 2

## Introduction

### 2.1 Problem Statement

Many organizations, companies, and individuals face challenges represented by the need to use a large number of applications/sites to organize their work, as they usually need sites to disseminate their experience and information, search and apply for jobs, identify local and international companies, organize the communication process between individuals and groups, Organizing their calendar and personal tasks, organizing their company's/organization's structure and distributing it into groups that contain multiple communication channels and features.

### 2.2 Objectives

Helping individuals/companies/organizations to publish their work/experiences/events/personal profiles, in addition to structuring and organizing users' work through a large number of features that we provide, such as adding events and tasks and creating pages and groups, which contributes to amplifying the growth of the application's users.

### 2.3 Scope of the work

Our application focuses on providing users interested in organizing their professional work with the access to a large network of other users/pages with the same interests. The application provides a huge number of features that can be used to save time and effort to manage tasks, profiles, connections, and business structuring for people, companies, and organizations.

### 2.4 Importance

Communication plays a major role in our daily lives, as it facilitates the process of exchanging and transferring information, gaining experiences, and getting to know people who contribute to enriching personal capabilities and competencies. Providing channels for optimal communication between people is a major challenge, as you need to identify the special segment of targeted users, know their needs and difficulties that they face

while using current channels, determining their goals for the communication process and trying to facilitate it, saving time and effort by including features that cover their needs.

Here comes the main role of our application, which targets professional people and companies/organizations interested in disseminating their work and reaching the largest number of people interested in the same field. Our application provides a very large number of publishing and communication channels, the most important of which are: adding connections, following pages, creating groups, sharing Profiles, publishing posts/events/jobs, real-time communication between (users, users and pages, company employees, group members) in all possible communication methods (chats, voice calls, video meetings). Our application combines the main features of social networking sites, tools for arranging tasks/events, and tools for structuring and organizing businesses, making it easier for target users to get the greatest possible benefit from using social networking sites to develop their careers and businesses.

## 2.5 Report Organization

This report is organized as follows; Chapter1: The Abstract, Chapter 2: introduction and overview about the objectives, the scope of this project and importance. Chapter 3: problems and constraints we faced in this project. Chapter 4: literature review, Chapter5: methodology and explained in detail about the tools and languages used and how we used them in our application. Chapter6: Result and discussion, Chapter7: Conclusion and Future work.

# Chapter 3

## Constraints and Earlier Coursework

### 3.1 Constraints & Limitations

1. Limited time: We needed to learn:

- New programming languages such as Dart and mobile phone frameworks such as Flutter to implement front-end in addition to back-end frameworks such as Node.js.
- We also needed to learn some important technologies and libraries such as:
  - sequelize: to implement queries on the database.
  - Server-Sent Event (SSE): that we used to send notifications to users.
  - Sockets: that we used to send and receive messages (text/video/audio), and Signaling the WebRTC communications.
  - WebRTC: to fully control the communication process:
    - \* Peer-to-Peer audio/video call using Mesh structure.
    - \* group video meeting using Selective Forwarding Unit (SFU) structure.

It took a very long time to learn all these techniques, apply them, and solve the problems we face. But we were able to implement all the features we mentioned in the abstract perfectly and without using services such as Firebase or ready-made libraries for video/audio call and group meeting.

2. The high complexity of the project: Our project brings together a very large number of features that require, in addition to a large amount of time to implement, deep thinking to transform requirements into technological solutions, as the project contains:
- All the main features of social networking sites such as: Protection using email verification code (a specific number of attempts for the login process/password reset/account confirmation) and storing IP address for each attempt, tokens to authorize any communication process with the server, personal profile, publishing posts (taking into account privacy), adding connections, chat, video and audio call, edit posts while showing editing history, post likes/comments,

search filtering, and a huge number of other secondary features that facilitate the use of the application

- Organizing tasks/events for User in two main interfaces: the calendar to add events with the ability to specify a Reminder with a specific date and time to send a notification to the user, Tasks which enables the user to add tasks he wants to perform and change their status (To Do, Doing, Done, Archive, Delete) in addition to the ability to add a Reminder for each task with a specific date and time so that it sends User notifications
- Organizing and structuring pages that represent companies/organizations: determining the permissions of administrators and employees, and the possibility of dividing them into groups (represented in the form of trees) with all the previously mentioned characteristics of the groups.

All of these main features, in addition to a huge number of secondary features that fall under them, posed a great challenge in thinking about how to apply and represent them logically.

3. Lack of sources: We faced a major problem after determining the technologies that we wanted to use in the project, which is the scarcity of the resources necessary to study and implement them. The most important of these technologies are (SSE and WebRTC), especially when using flutter in applying these technologies, as in addition to the lack of resources, there is a huge scarcity of available libraries and the lack of clarity in their documentation, in addition to many unsolved bugs by the libraries developers, as we were forced to modify one of the libraries to suit our needs.

## 3.2 Standards

1. Agile Model: Every task in our project divided into 6 stages: Plan, design, develop, test, deploy, review.

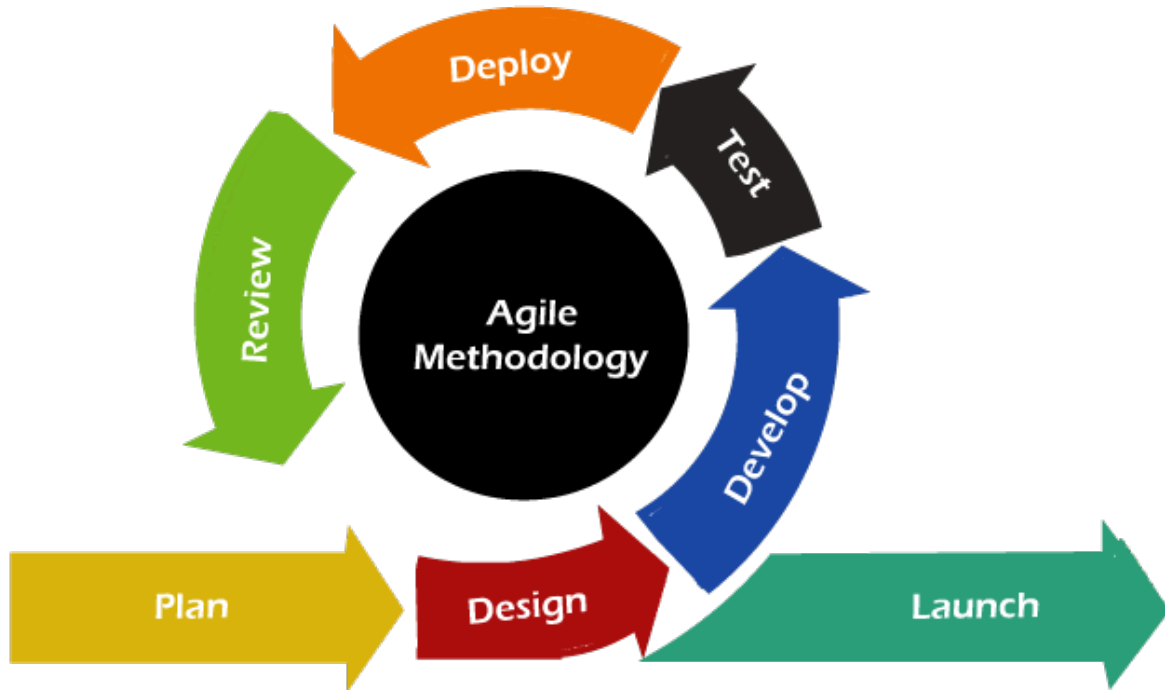


Figure 3.1: Applied Agile model.

### 3.3 Earlier Coursework

1. Database Design Management course:  
This platform's database is entirely made on MySQL. Database design principles and crucial concepts like dependencies, keys, and indexes were also covered in this course.
2. Object-Oriented Programming course:  
Which we used to store the info that coming from the server.
3. Distributed operating system course:  
This course helped us to understand the idea of APIs. So the whole back-end built based on that.
4. Networks courses:  
It helped us learn about protocols for transferring information, which made it easier for us to understand and program WebRTC technologies.
5. Online courses:  
For Dart, Flutter, Nodejs, sequelize, Sockets.
6. Software Engineering and Advanced Software courses:  
Was instrumental in helping us adhere to software engineering standards while writing our code.

## Chapter 4

# Literature Review

There are a large number of applications that include a limited number of features that our application offers, as applications and services are usually divided into social networking sites, task organizing sites, and companies/organizations work organizing sites. Our application is distinguished by its comprehensiveness of the most important features provided by these sites.



# Chapter 5

## Methodology

### 5.1 Tools and services

We used tools and services that facilitated the front-end and back-end programming process, and facilitated the implementation and organization of the agile model stages.

1. Visual Studio Code:

We used it for front-end and back-end programming, as this program provides many features that facilitate the programming process.

2. MySQL Workbench:

It has a simple and easy interface to directly deal with databases and create queries for testing.

3. GitHub:

We hosted the code on it, as it facilitated the programming process and the distribution of tasks among team members through a large number of services and features that it provides.

4. Postman:

We used it mainly to test the APIs that we create on the server, to facilitate the process of displaying response data and using it in Front End.

### 5.2 Database Design and Configuration

#### 5.2.1 MySQL

We used MySQL as a database because of our previous knowledge of it, which makes it easier for us to work with. MySQL is considered one of the very fast database languages, backed by a large number of the benchmark test.

Our database contains 39 tables, divided mainly into two main tables: the user table and the page table. Then the rest of the tables are integrated to serve the features of the user



## 5.3 Front-end

### 5.3.1 Flutter

We used the Flutter Framework to program Front End, as it provides a large number of features, the most important of which is its ability to build applications that work on the web and mobile at the same time and the ease of modifying the design to suit the dimensions of the web and mobile.

We used a number of main libraries to program the most important features in our application, the most important of which are (The mechanism of its operation will be explained in the back-end section):

1. featurehub sse client:

Server-Sent Events opens a continuous connection between client and server when the user opens the app and subscribe to the server events, the server can send new data without the client needing to request it.

2. socket io client:

We used it to implement sockets in conversations and signalling for WebRTC.

3. flutter webrtc:

We used it for peer-to-peer audio/video calling, and for group meeting.

4. flutter background:

We used it to allow the application to run in the background on mobile, we mainly needed this feature to implement the screen sharing feature in video meeting/calling.

5. awesome notifications for mobile and quick notify for web:

We used it to show pop-up notifications on both mobile and web.

## 5.4 Back-end Node Js

We used Node.js Framework to program the backend. Our project contains two servers. The first is specialized in receiving the regular HTTP request from the client, executing the required operation, and returning the response to the client. The second server is specialized in managing sockets and sending notifications and managing WebRTC meetings. Dividing the backend into two servers reduces traffic on the back end, we can later divide the databases into two databases, one containing only the messages and notifications tables, and the other containing all other tables. Thus, we can reduce the traffic on the databases as well (since notifications and messages constitute greater traffic than the rest of the tables on both the server and databases).

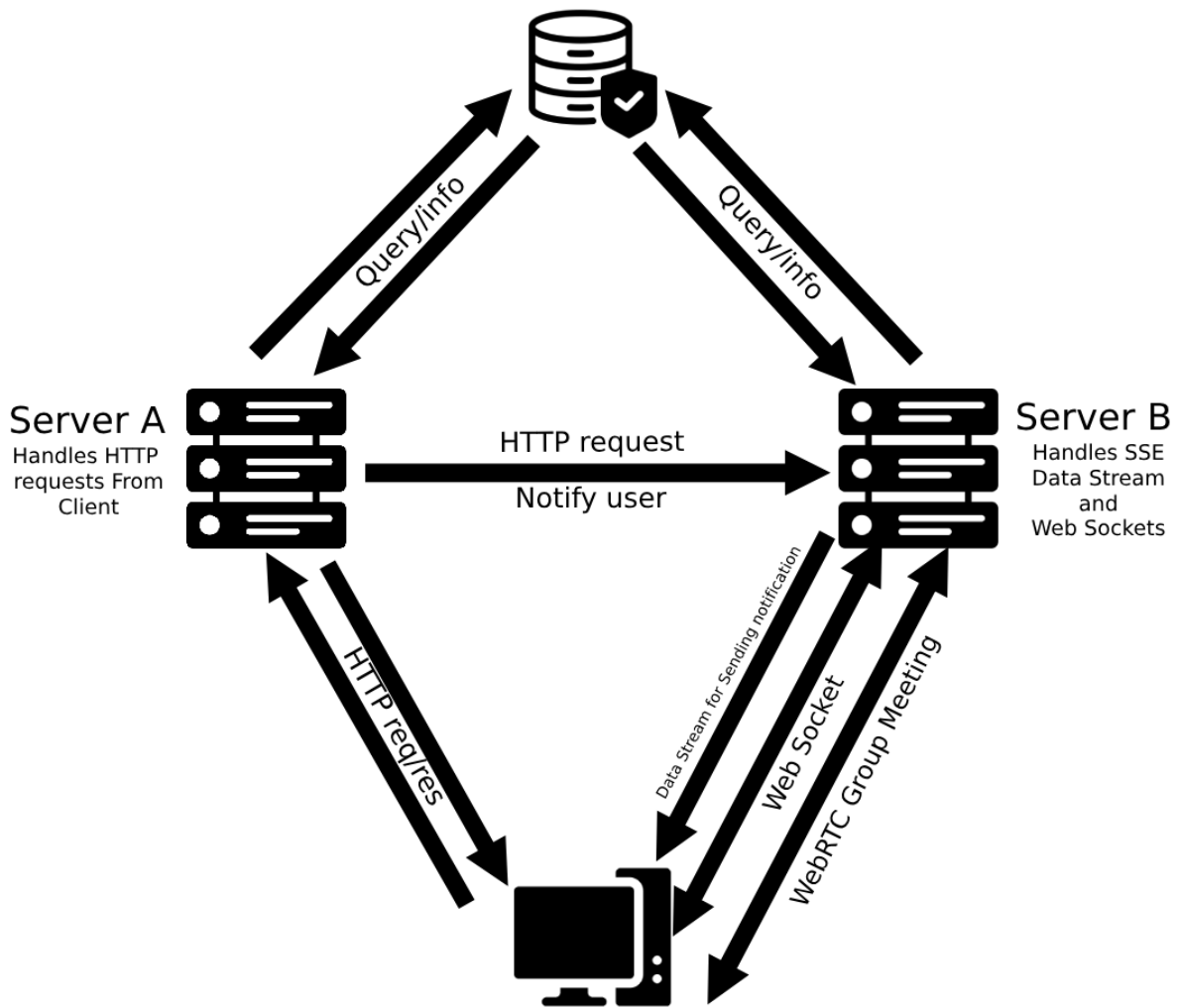


Figure 5.2: Our Project Model

We also implemented a large number of features and used a large number of techniques and architectures, the most important of which are:

#### 5.4.1 Architecture style

We used Restfull APIs to communicate between the client and the server, as the server A has more than 160 endpoint API that controls different functions.

#### 5.4.2 JWT

We used JWT for the authentication process, when the user logs in, he gets two tokens (access token and refresh token). The refresh token is stored in the data base, and in each request, the client attaches the access token to the request's header. The duration of the access token is 10 minutes, and when it expires the application automatically sends a

request to update the Access Token by attaching the Refresh Token to the request header, then the server returns a new Access Token that is valid for another 10 minutes. The Refresh Token is valid unless the user logs out, as it is deleted from the data base and cannot be used again after that.

### 5.4.3 Sequelize

Sequelize is a Nodejs based Object Relational Mapper that makes it easy to work with MySQL databases, Also Sequelize has a Built-in protection against any SQL Injection attacks. Sequelize facilitated the process of organizing the code for queries and the process of organizing responses in objects and controlling them before sending them to the client.

### 5.4.4 nodemailer

We used nodemailer for Sending an email containing the verification code during the process of: Sign Up, Reset password, changing the email.

### 5.4.5 axios

We used it to send requests between server A and B.

### 5.4.6 Server-Sent Events

SSE is a web technology that allows a server to send real time data to the clients over HTTP. It is a unidirectional communication channel where the user need to subscribe to the server events then the server can push data to the client without the need for the client to send a request. we used it to send Real-time notifications to users once its available.

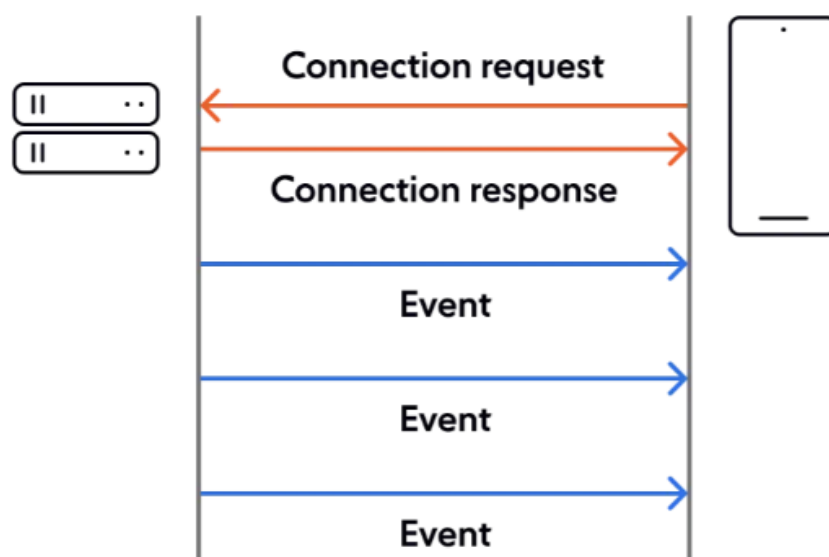


Figure 5.3: Server-Sent Events

## Why not Web Sockets?

- While Server-Sent Events and Web Sockets both enable Real-time communication, Server-Sent Events is more suitable for cases that the server needs to send data to clients "unidirectionally", such as in notification. Web Sockets are more comprehensive and support full-duplex communication "bidirectional data stream".
- No need to rely on the external library JavaScript provides interface to send data stream from server to client.
- SSE sessions can reconnect automatically after disconnect.

### 5.4.7 Socket io

Web Socket operates over a single TCP connection fetching real-time data, providing a lightweight and efficient communication channel "bidirectional". The connection begins with a Web Socket handshake, and once established, both the server and client can send/receive data as needed.

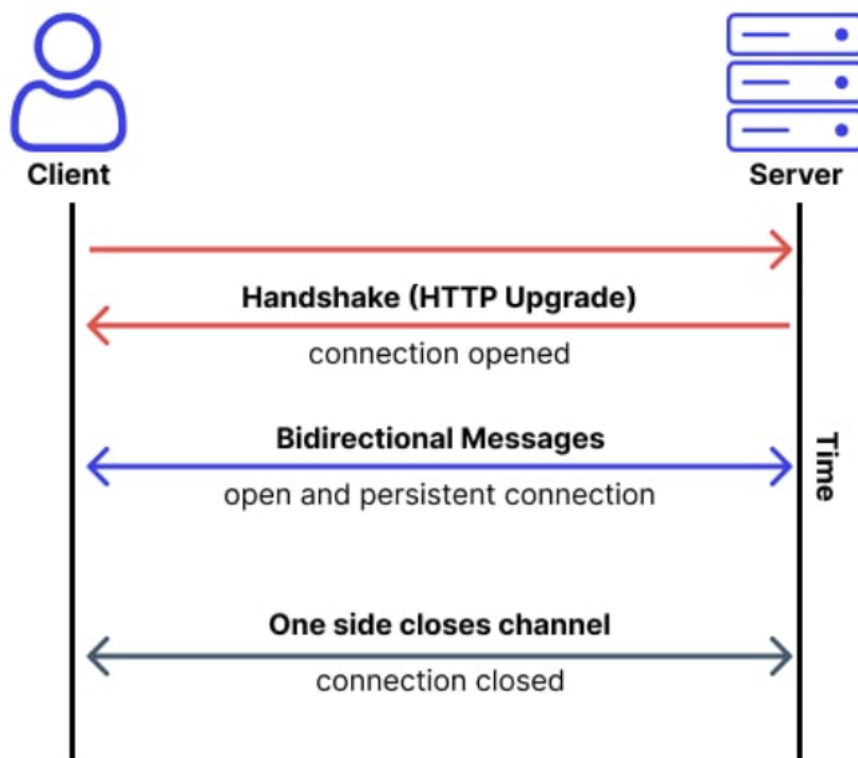


Figure 5.4: WebSockets

## Why not HTTP request?

- The connection isn't open for long as WebSockets.
- In WebSocket connection, the data is sent as soon as it's available. The client doesn't need to keep requesting it, and doesn't require the headers and handshakes for each

request-response cycle. which result a much lower latency with a less overhead and network traffic.

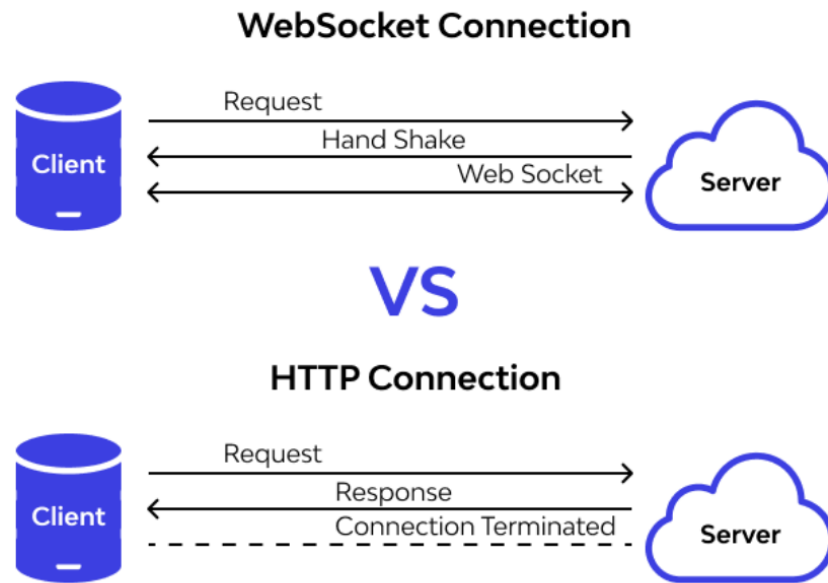


Figure 5.5: WebSockets Vs HTTP

### Where we applied it in our application?

We used websockets for two main functionalities:

- Conversation between two users (sending and receiving text/photos/videos).
- Group chat through rooms (sending and receiving text/photos/videos).
- WebRTC signalling process.

#### 5.4.8 WebRTC

We used WebRTC to make peer-to-peer video/audio calls between user-user or page-user, and also to conduct video meetings between a number of peer users. **We have fully programmed the interface and the server that handles the signaling process without using any services such as Jitsi Meet or ZegoCloud, or any ready-made library that handles the process.**

Web Real-time Communication is made to enhance the web browsing model. It allows Clients to directly exchange real-time media with other Clients in a peer-to-peer connection through secure access to input peripherals like webcams and microphones.

Traditional web architecture build on client-server paradigm, a client needs to send an HTTP request to the server and gets a response. In contrast, WebRTC allows the exchange of data among peers so, peers talk to each other without a server in the middle (after signalling process).

## 1- WebRTC video/audio call

Our application provides peer-to-peer communication between two users/ user-page through the call button on the chat page.

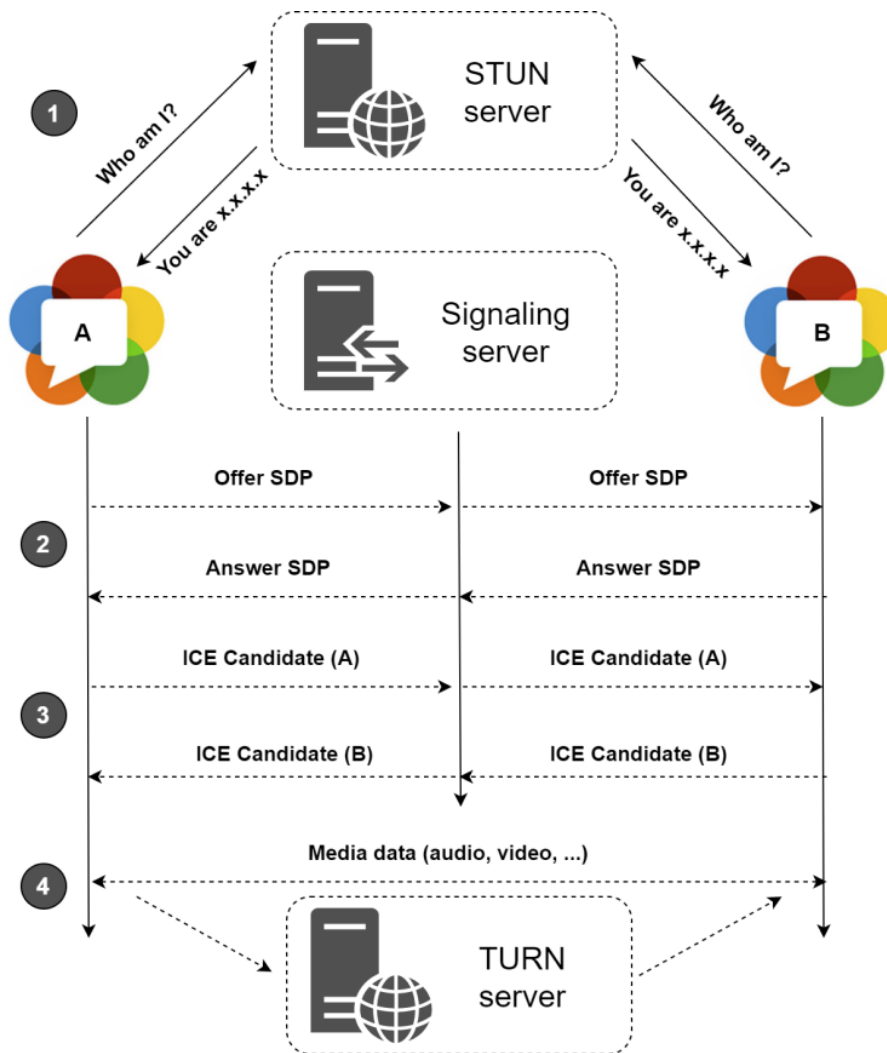


Figure 5.6: WebRTC Signalling Process

In our application, the sockets handles the signaling process, so when the user presses the call button (whether video or just audio), the signaling process starts as follow, three types of information must be exchanged:

- Session description protocol (SDP): describing Peer media capabilities (mainly OfferToReceiveVideo and OfferToReceiveAudio) and further session details:
  - Version: The version of the SDP protocol.
  - Origin: The originator of the SDP message, including the session ID and network address.
  - Connection Data: Information about the network addresses and ports.
  - Encryption: Information about encryption mechanisms being used to secure the connection.



After other peer receive the SDP Offer, then respond with his own SDP message as an answer, describing his capabilities and session details. Then two Peers compare the SDP messages and agree on a set of acceptable media parameters for both clients.

- ICE Candidates: describes the protocols and routing needed for WebRTC peer to be able to communicate with a remote peer, Every ICE contains "a node" of a peer network, until it reaches the outside. By this Peer sends these ICE's to the other peer, so they know through what connection points they can reach each other.
- Media Data: After that, the two peers can send and receive a data stream from each other, which is usually (video and audio, or one of them).

The efficiency of ICE allows it to calculate with mainly two methods:

- Via STUN servers: A STUN server checks the IP address and port of incoming requests, then sends that address back to the WebRTC Peer. The Peer thus uses a STUN server to get his own public IP port address. This allows the Peer to offer a publicly accessible address, which is then passed to another peer via the signaling process.
- Via TURN servers: Sometimes security measures block STUN servers from connecting devices, then the connection switches to the TURN server, TURN relay servers are used to stream audio, video, and other real-time data between peers, it acts as a translator for peers sharing data, It creates encrypted packages of data peers can send to each other in real time without sharing their IP addresses.

## 2- WebRTC group video meeting

As we mentioned earlier, our application provides the ability to create a meeting between a number of users through the meeting button located on the group chat page.

### Which WebRTC architecture is best suited for this situation?

There are mainly three WebRTC architectures:

- Mesh architecture: Which is mainly the direct peer-to-peer connection between two clients which we used in the call process, and is generally used in applications of calls between two users. but when expanding the model to serve multiple peers, the mesh architecture require each peer to send its stream to each user in the meeting, and receive a stream from each user.
- Multipoint Control Unit (MCU): In this architecture the server still be in the middle of all participants connections functioning as a "central gateway" that is decodes each participant's stream and creates a single new stream from each of the streams it has received, then distributes a "single" stream back to every participant, regardless of how many people are in the conference call.
- Selective Forwarding Unit (SFU): This is a more popular approach than MCU when there is a need to scale to more than 6 participants. also this approach involves the server. The difference is that each participant sends their stream to the server and the server sends each stream that it receives back to all of the other participants "works as router".

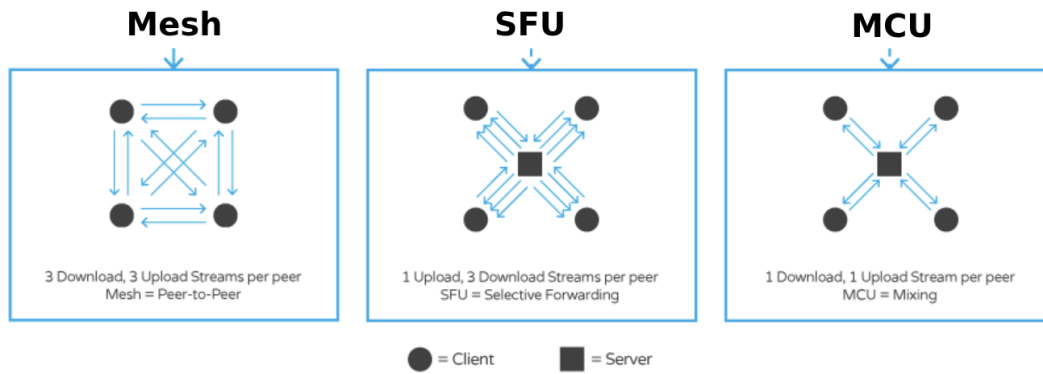


Figure 5.7: WebRTC architectures

In Group Meeting, we built SFU Architecture group meeting, as it is already mainly used in similar services such as Jitsi Meet and ZegoCloud, and is partially used in Zoom Meeting.

We chose this architecture because it distributes the load between the server and the client, as the MCU places the entire load on the server and is not economically feasible. Most services and applications use SFU Structure or Hybrid Structure to implement group meeting.

**Why programming meeting from scratch when there are ready-made services that can be used?**

- Learning new technology, as the practical application of WebRTC using the various architectures that we programmed gave us huge experience in this field.
- Ready-made services provide their technologies for free for trial only, as they are limited to a certain number of minutes per month. During production, you need to pay for these services in exchange for using them in the application.
- Creating WebRTC applications from scratch gives us full control over all the data sent between Peers, including not relying on security that ready-made services provides, which contributes in the future to developing the application to include other features and special data encryption.
- **We do not consider writing two lines of code in our application as a feature.** Using ready-made services requires calling the library and then calling the pre-prepared interface only. All processing also takes place on the servers of the ready-made service.

## 5.5 Growify as Web application & Mobile application

### 5.5.1 Sign Up

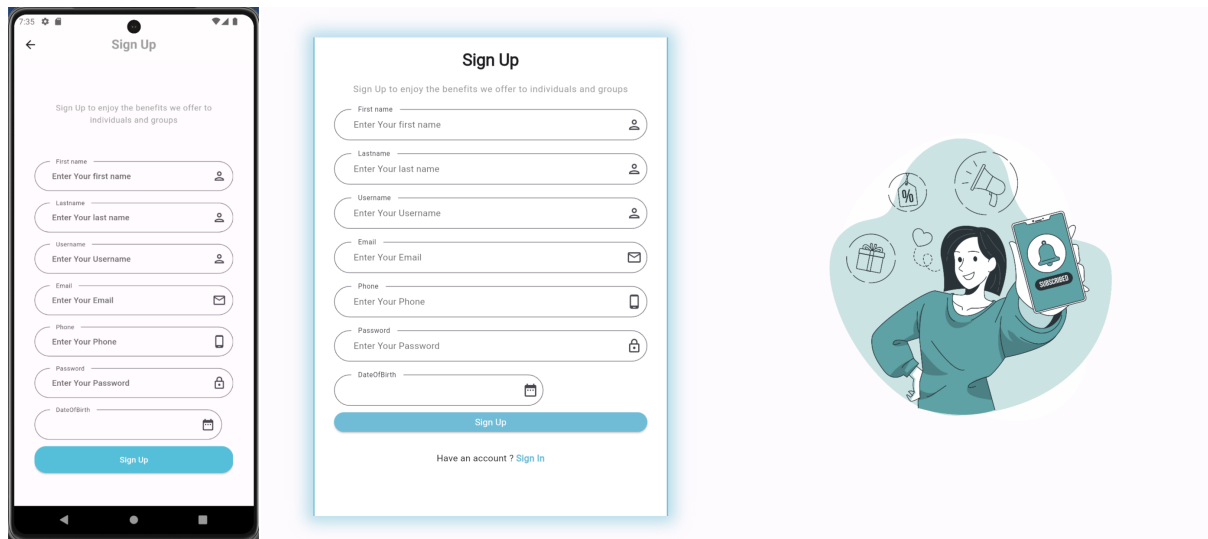


Figure 5.8: Sign Up

In the Sign Up interface, we request the basic information from the user (the user can later add a large number of other information from the settings interface). When user click on the Sign Up button, the information is verified to be correct before sending it to the server (username, email,Phone and password), Its validity is verified again on the server.

After that, the user information is added to the TempUser table and a verification code is sent to the user's email.

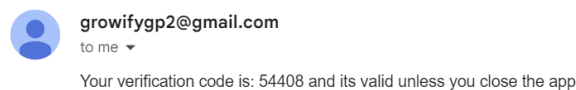


Figure 5.9: code Sent to user email

## verification code

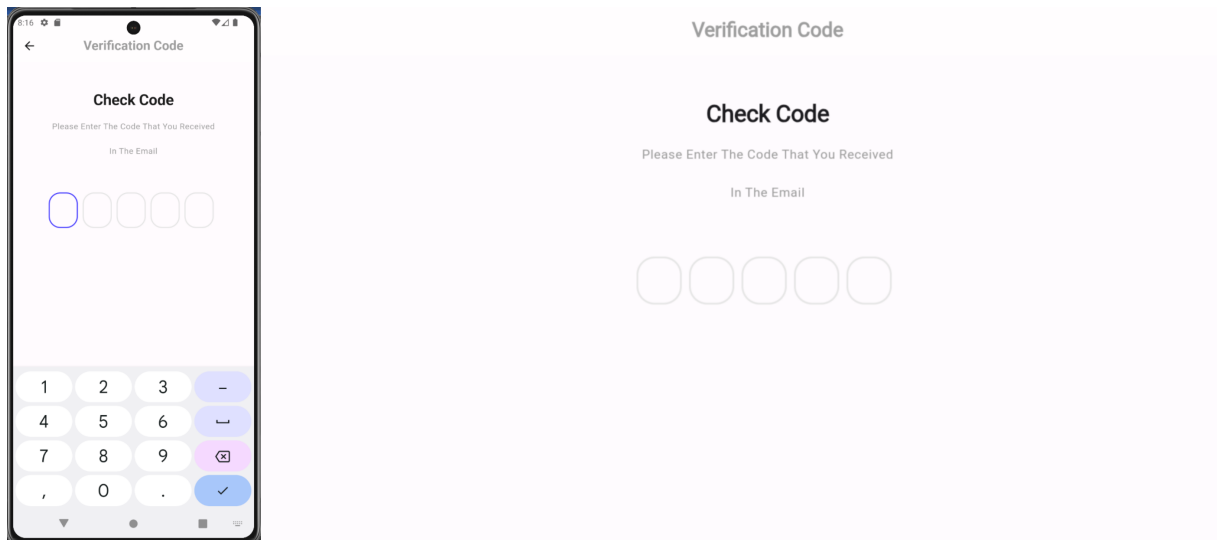


Figure 5.10: verification code

On this page, the user enters the code that was sent to the email. The code get sent to the server. If the code is incorrect, the user has two more attempts to enter the code. If the number of allowed attempts is exceeded, user will be deleted from the TempUsers table and the user need to reSign up to prevent spamming.

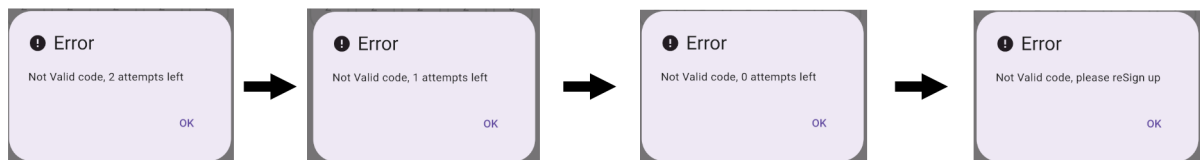


Figure 5.11: Sign up attempts exceeded

If the user enters the correct code, he will be routed to the success sign up screen

## Sign up successful

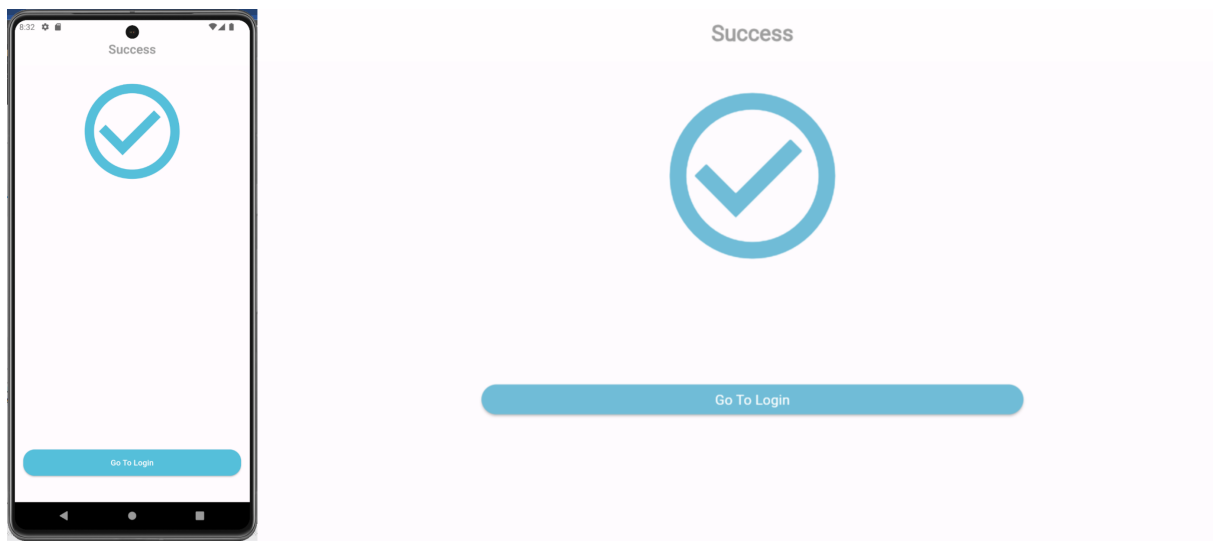


Figure 5.12: Sign up successful

## 5.5.2 Forgot password

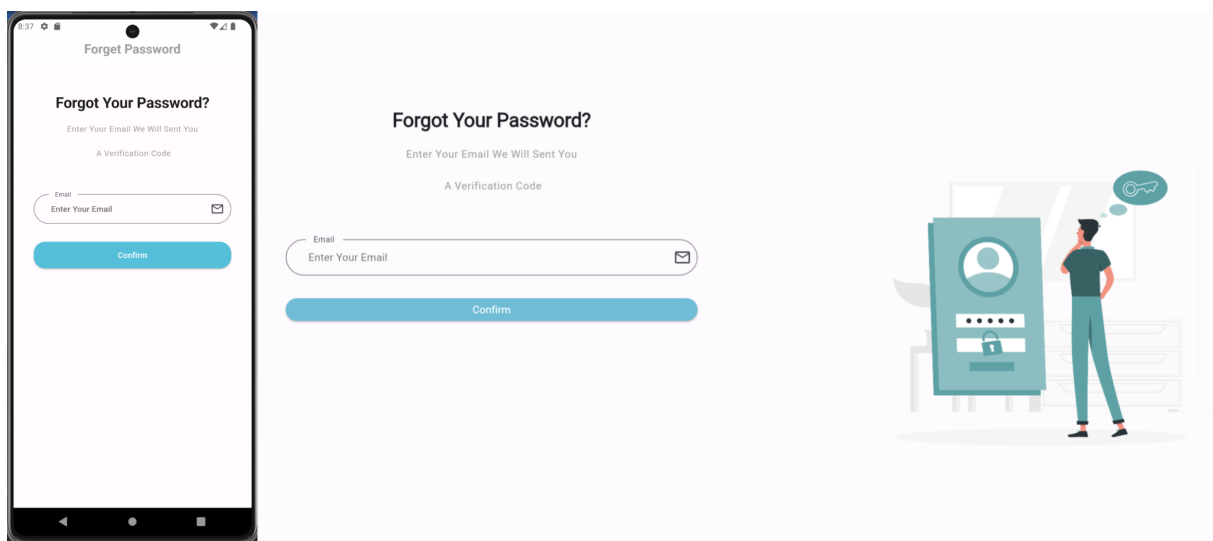


Figure 5.13: Forgot Password

User can reach Forgot password interface from login interface, where the user can enter his email and a verification code will be sent to his email. User's email, code, IP address and attempt counter, will be stored in "forgotPassword-Codes table".

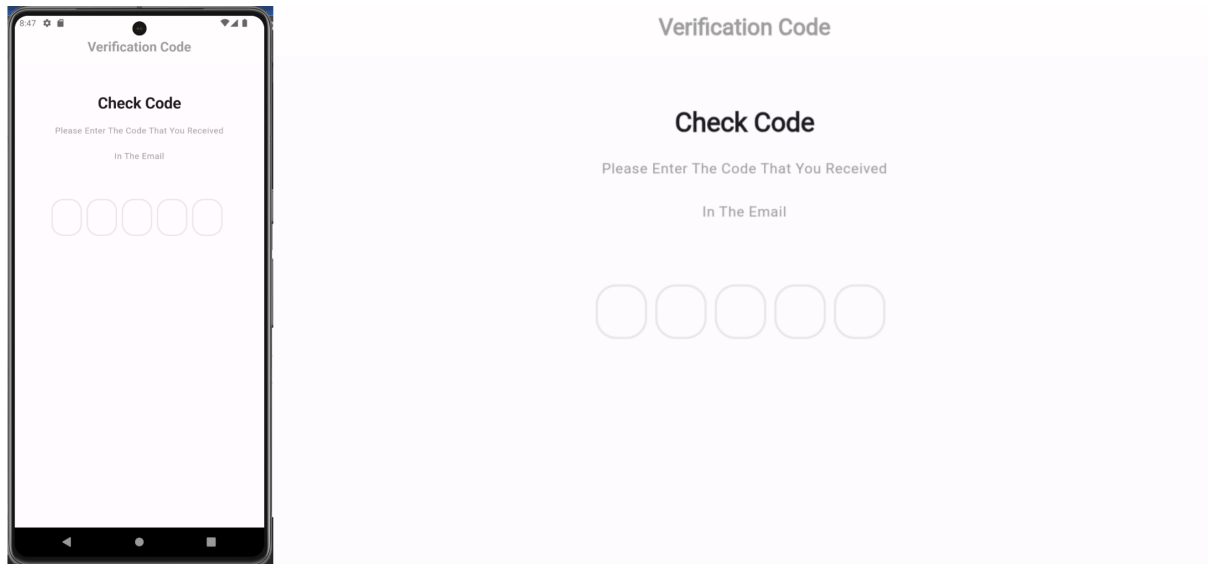


Figure 5.14: Reset password verification code

On this page, the user enters the code that was sent to the email. The code get sent to the server. If the code is incorrect, the user has two more attempts to enter the code. If the number of allowed attempts is exceeded, the IP address of the user will be banned from trying to reset the password for 1 hour.

By taking into account the user's IP address, we can prevent a spamming attempts on the user trying to reset the password through someone else resetting it incorrectly, and thus the ban is not imposed on the account itself, but on the IP address.

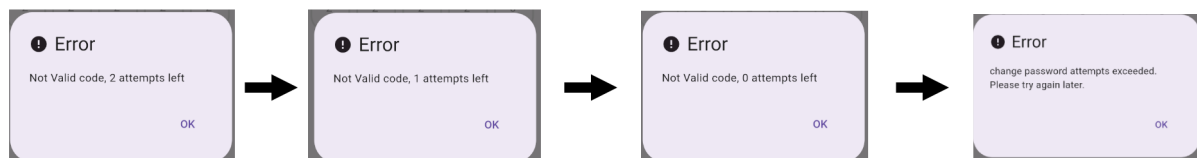


Figure 5.15: change Password attempts

### Why use IP Address and not Mac Address? Isn't this less secure?

Of course, it is less secure, but operating systems prevent access to Mac Address, so using IP Address provides a kind of security within the available capabilities.

Then if the user enters the correct code, he will be routed to the reset password screen, at the same time, the info in the "forgetPasswordCodes table" wont be deleted until user finish the resetting password process. **Why?**

After entering the code correctly, States Code 200 is sent to the client to be routed to the password reset screen. If the user is able to disconnect from the server and send States Code 200 after entering the code incorrectly, he will be able to reset the password directly, the information is preserved in the "forgetPasswordCodes table" and the code is re-sent again with the new password so that the password reset process is completed in a secure way.

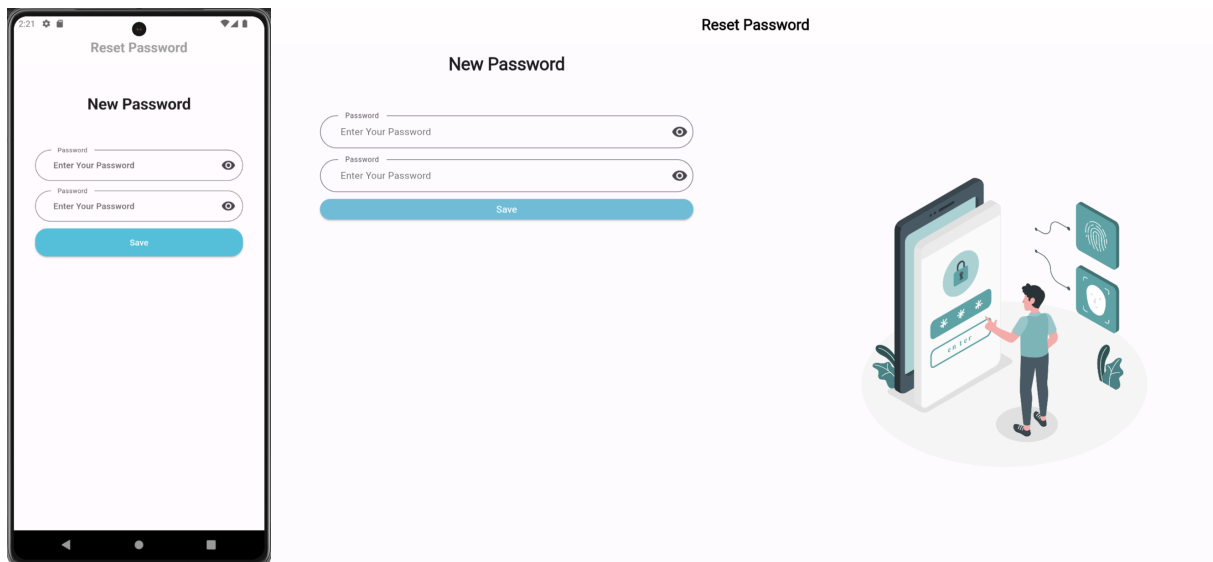


Figure 5.16: Reset Password

After reaching the reset password page, the user can send the password after ensuring that it matches, in addition to the verification code that he entered on the previous page, and the code is re-verified and that the password verified that it matches the conditions (longer than 8 characters), if the code is incorrect, This means that the user has bypassed the code verification screen and thus the process must be repeated again.

When verification of conditions is complete, the password is encrypted and stored in the user table, and the user is routed to the login screen.

### 5.5.3 Login

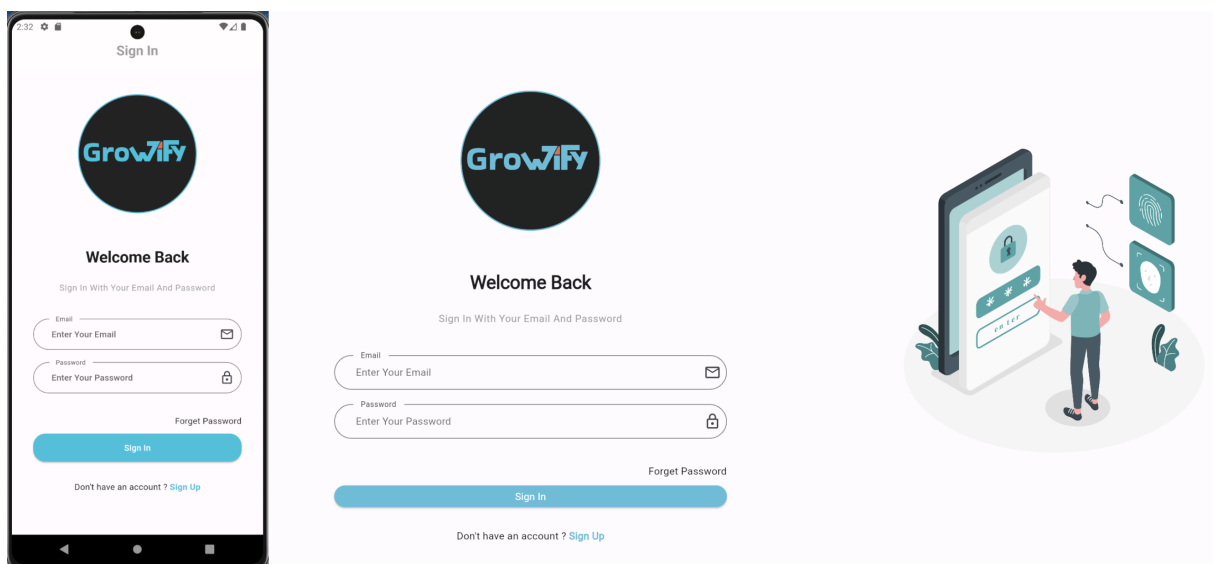


Figure 5.17: Login

In the login interface, the user enters the email and password, and their validity is verified on the server. If the information is correct, an access token and a reference token

are produced (containing the user's email and userName) to be used in all requests sent to the server later, and then the main user information is sent to the user, with the Access Token, and the Refresh Token, in addition the user added to the "activeUsers table".

The user has three attempts to enter the password, which are recorded in the "loginAttempts table". If he enters all of his attempts incorrectly, the user's device's IP will be blocked from logging in for 1 hour, also here. We impose a ban on the device's IP and not on the account itself, to prevent any spamming process that attempts to prevent the user from logging in.

After successfully logging in, the user main info will get stored so user don't have to log in every time he opens the app (unless he loges out)

## 5.5.4 Home Screen

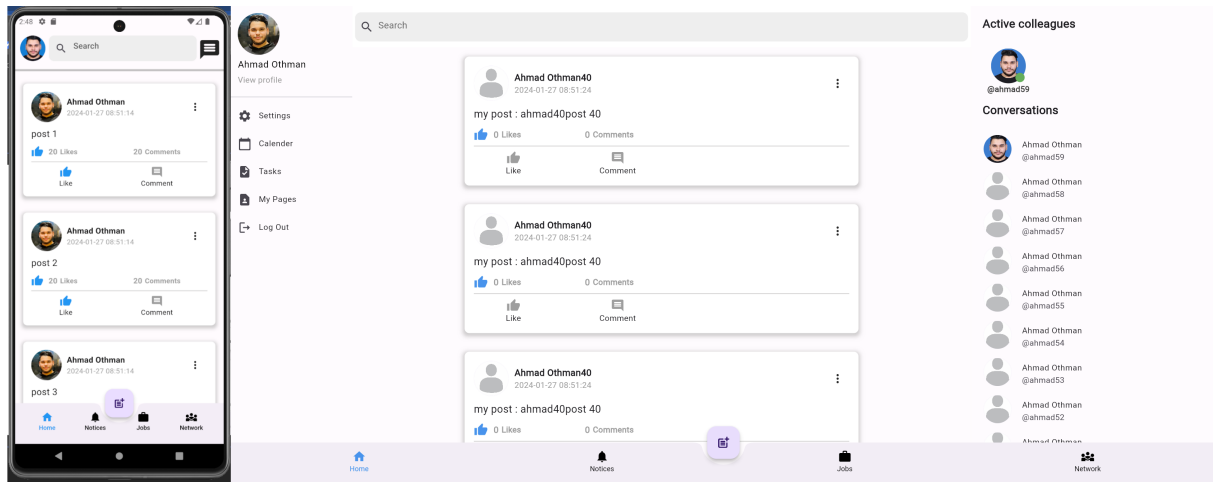


Figure 5.18: Home Screen

First, every time user login or opens the application, he will subscribe to the SSE data stream so he can receive notifications. The Home screen contains a large number of details that will be explained in detail in the upcoming sections

On the left there is a list that contains a number of functionalities that will be explained in the next sections. This list can be accessed from the mobile phone via clicking on the user's picture in the upper left corner. On the right there is a list that contains the user's private chats, above which are the online connections. It can be accessed from the mobile phone from By clicking on the chat icon in the upper right corner, in the middle there is the search, posts, which are the posts of the user Connections in addition to the pages that the user follows, 10 posts are loaded at a time, and when user scroll to the bottom of the page, 10 other posts are loaded, and so on. Below is the bottom bar, which contains a number of functionalities that will be explained in the next sections.



## 5.5.5 User Settings

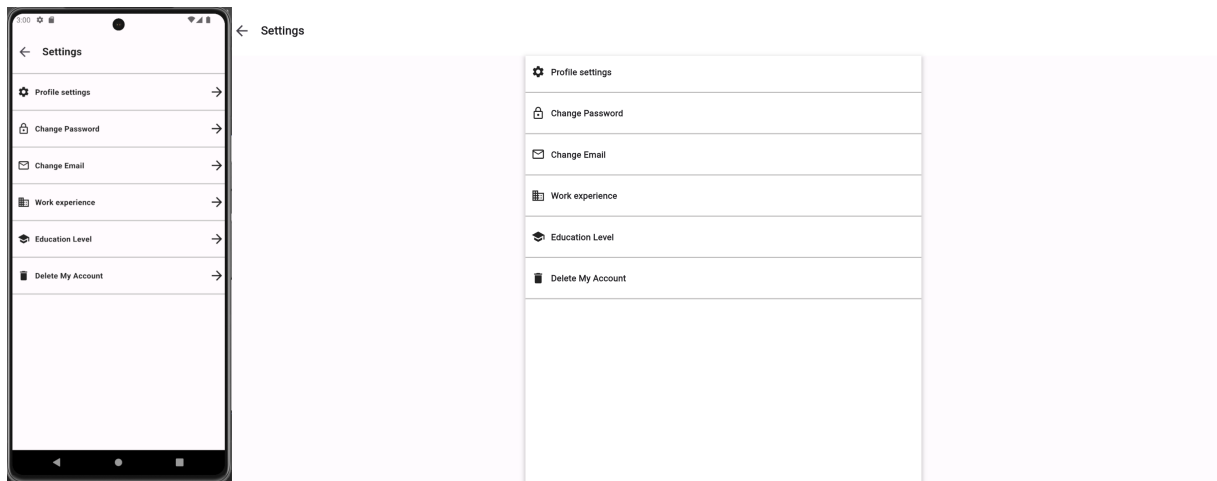


Figure 5.19: User Settings

This page provides full control over the user's account. This interface provides access and control to:

### Profile Settings

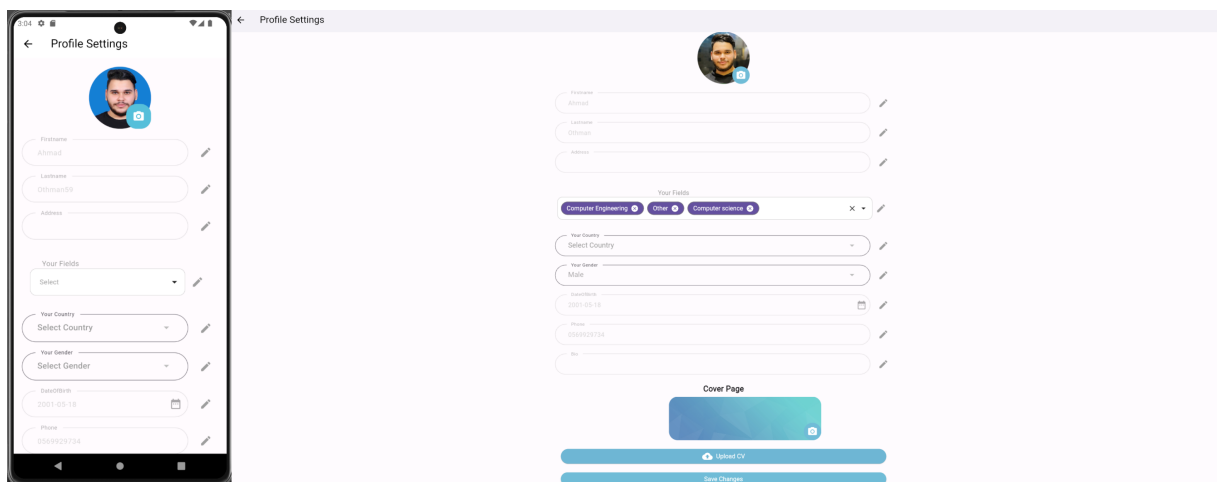


Figure 5.20: Profile Settings

The user can change the basic information that he entered when registering the account, in addition to a number of information that facilitates the filtering process for searching and notifying, main information to be explained:

- User fields: Where the user chooses from a number of options that are already present in the data base for the fields he is interested in, these fields will be used later for search filtration and to send a notification when a page that the user follows adds a new job whose fields are similar to at least one of the user's fields.

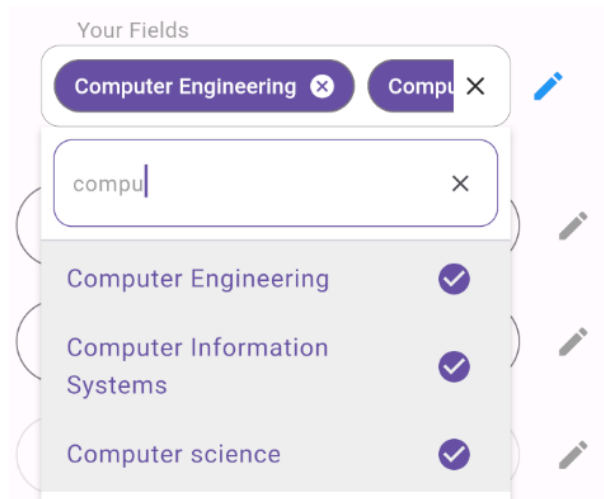


Figure 5.21: user Fields

- Country, Gender: It will later be used in search filtration to improve the user experience.
- bio: It is a short overview of the user that is displayed under his name in the profile page.
- Cover Image: The cover image is displayed in the user's profile. When the account is registered for the first time, the profile image and the cover image are a default image, and the user can change them from the settings.
- CV: The user can upload his CV, where it is displayed in his profile, and other users/pages can download. user can change the CV an infinite number of times.

## Change Password

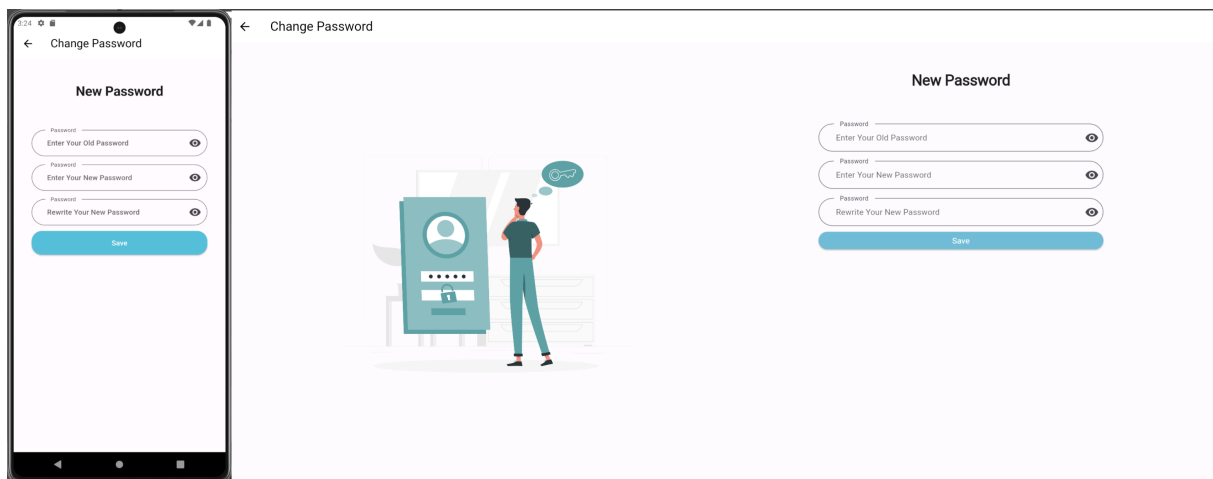


Figure 5.22: Change Password

On this page, the user can enter the old password and then the new password to reset it after verifying that the old password is correct, and verifying that the conditions apply to the new password.

## Change Email

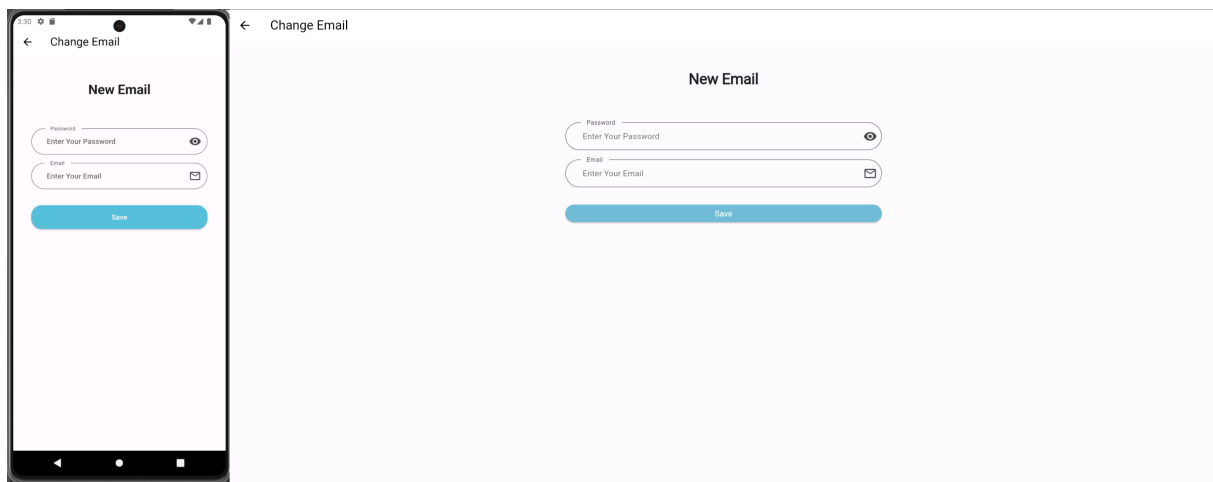


Figure 5.23: Change Email

On this page, the user can enter the password and then the new Email to change it after verifying that the password is correct, and verifying that the conditions apply to the new Email. After that, a verification code will be sent to his new email, then the new email, verification code, IP address and attempts counter will be stored in "changeEmail table", and user will be routed to the verification code screen.

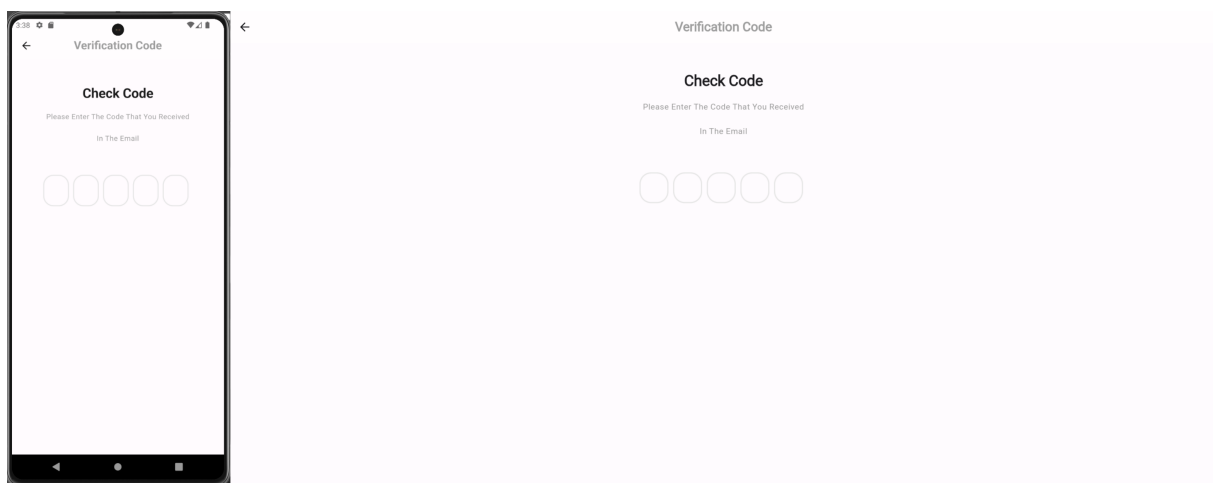


Figure 5.24: verifying change email

Here the user also has three attempts to enter the code, in order to prevent spam on someone else's email, and also provide a layer of protection to prevent the use of an email that the user does not own. After confirming the code, the email in the user's table is changed and then the user is routed to the login screen.

## Work Experience

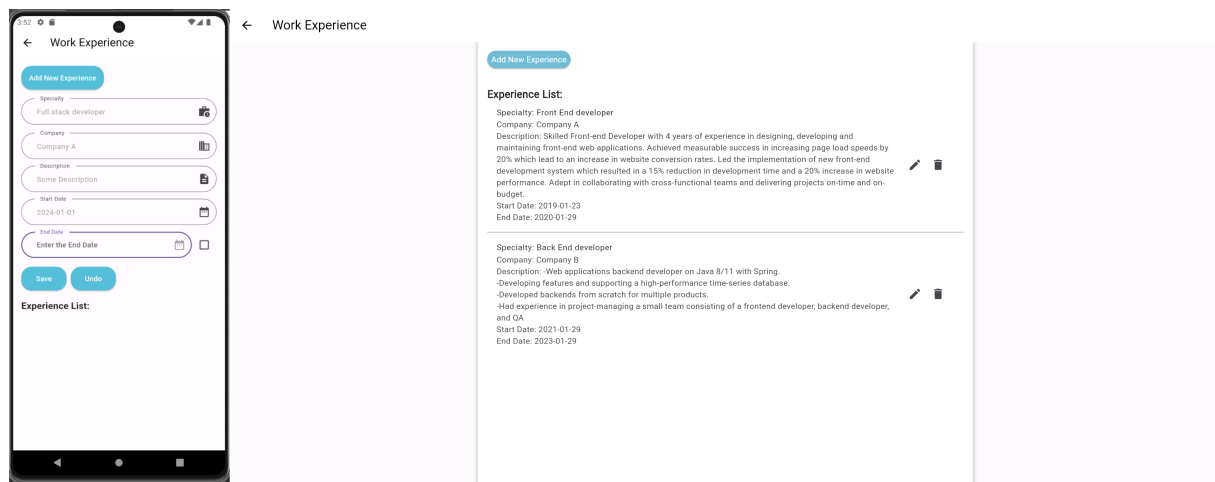


Figure 5.25: Work Experience

Here the user can display the work experience that appears on his profile page. The user can add/delete/edit any work experience. There is a checkbox at the end date. If user does not select it, the end date will appear "present" instead of a date.

When a user adds a page ID on the company field and the page is registered on the application, and the user is registered as an employee on this page with the same speciality that he puts in the speciality field, a verification sign will appear next to the Work Experience on the user's profile (it will be clarified and displayed later in next sections).

## Education Level

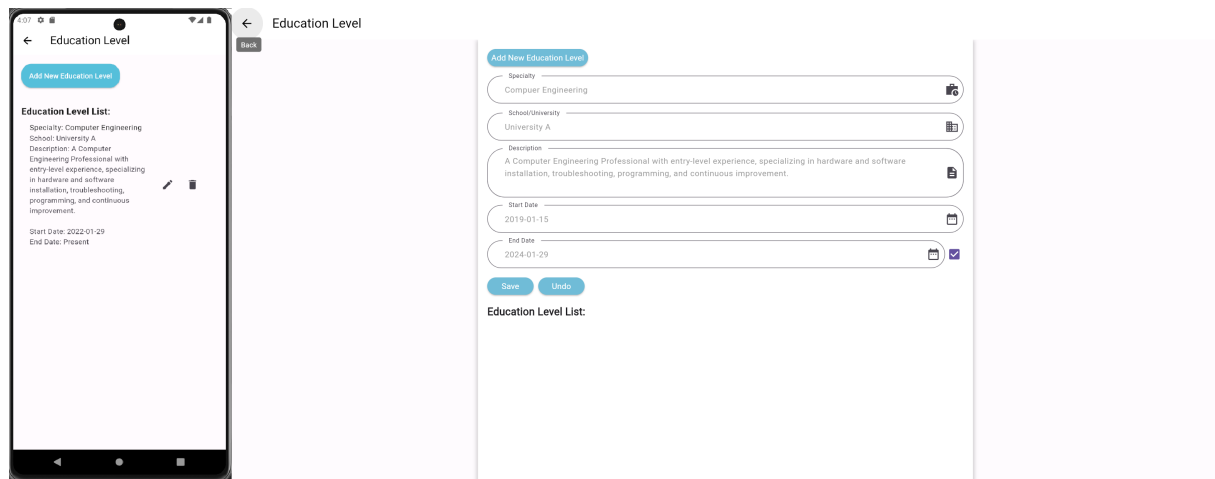


Figure 5.26: Education Level

Here the user can display the Education Level that appears on his profile page. The user can add/delete/edit any Education Level. There is a checkbox at the end date. If user does not select it, the end date will appear "present" instead of a date.

## Delete User Account



Figure 5.27: Delete User Account

When the user presses this button, the user is routed to an interface in which he enters the password to confirm the process of deleting the account. After confirming the process, the account is not deleted from the user's table while its state is converted to false, the photo and cover photo are deleted, and the user's information are deleted from the following tables: activeUsers, notifications, connections, sentConnection, workExperience, educationLevel, userTasks, changeEmail, forgetPasswordCode, userCalender, jobApplication, pageFollower, pageEmployees, pageAdmin, comment, like, post, groupTask, groupAdmin, and groupMember.

We do not completely delete the user from the user table because it is linked to the messages and group messages table, and it is prohibited to delete messages when deleting the user's account for security reasons.

### 5.5.6 User Calendar

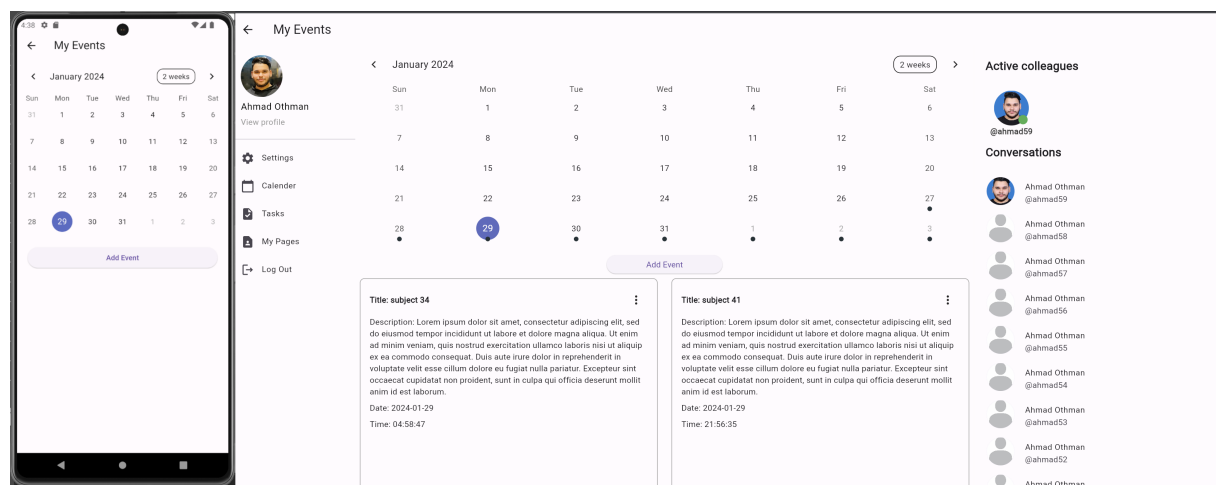


Figure 5.28: User Calendar

On this page, the user can view his (private) calendar. The user can view/add/delete events when pressing on a day in the calendar, so that this feature enables him to organize his work and day.

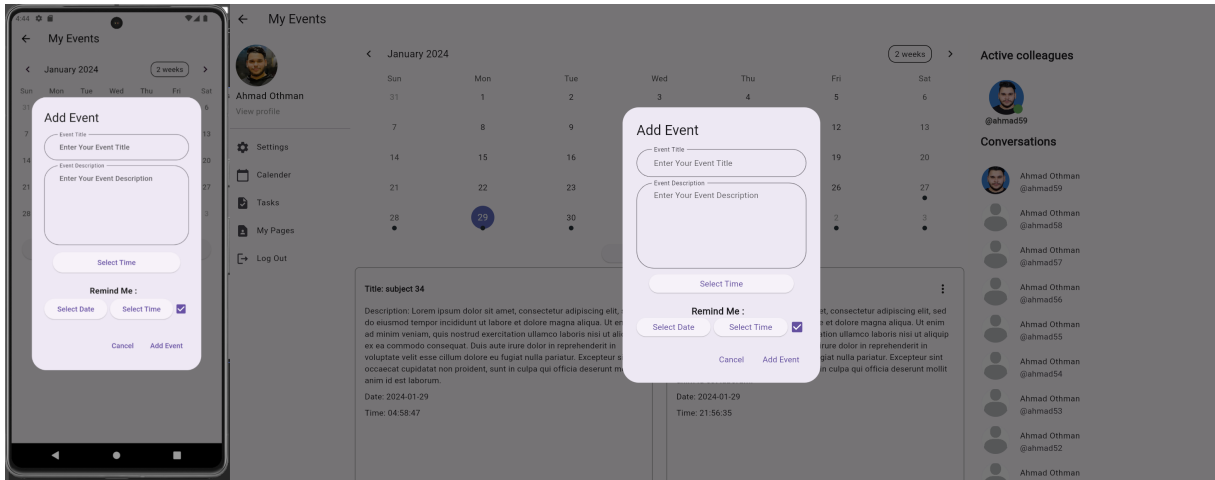


Figure 5.29: Add User Event

The user can add a new event with the following details: title, description, time, and an option if he wants to set a reminder for a specific date and time, as a local notification will be sent to the user when the local time matches the time that the user has specified.

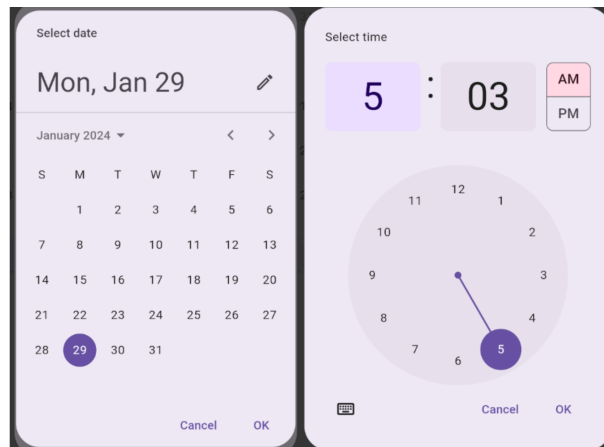


Figure 5.30: Set Reminder Date And Time

When user set a time / date for the reminder, a pop up notification will appear as follows:

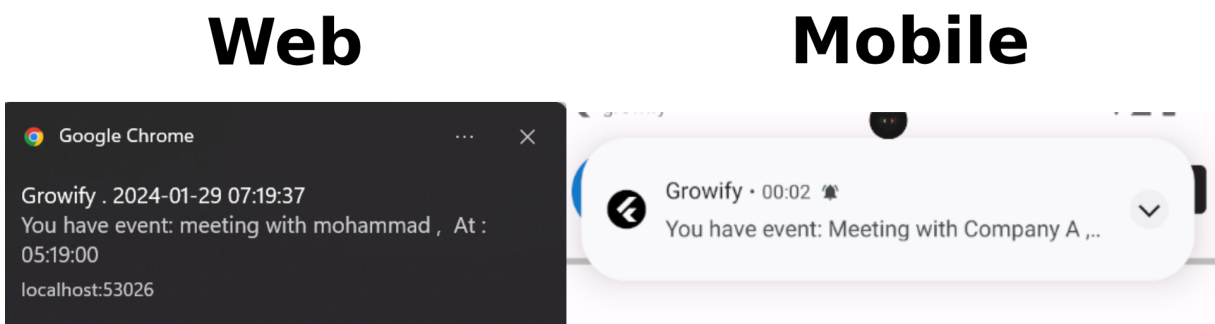


Figure 5.31: Event Notification

## 5.5.7 User Tasks

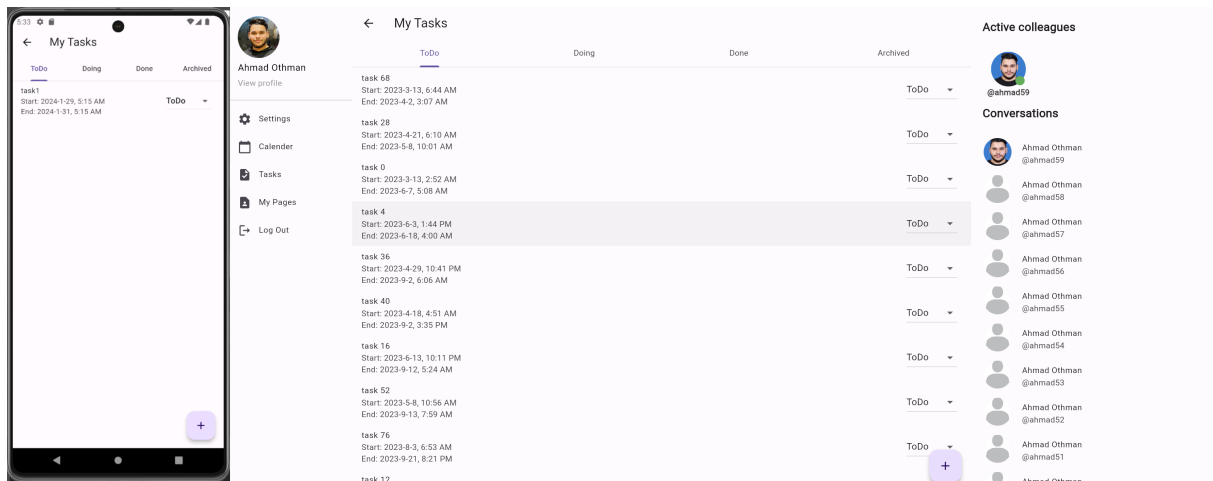


Figure 5.32: User Tasks

On this page, the user can view his (private) tasks. user can view/add/delete/move tasks, so this feature enables him to organize and classify his tasks. The tasks on this page are arranged based on the task end date. When user press on a task he can view the task details.

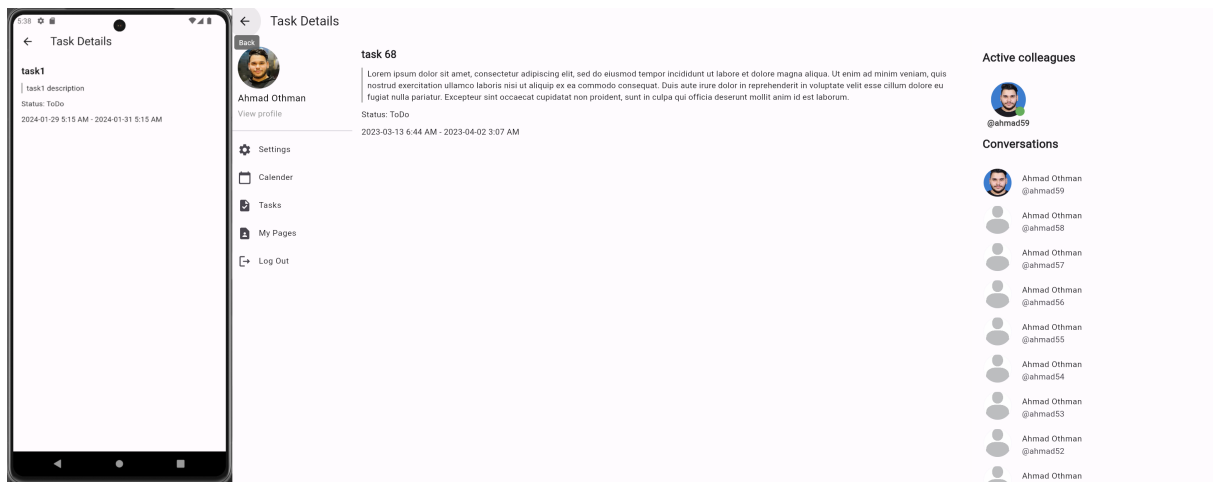


Figure 5.33: User Task View

The user can move the task between four tabs: To Do, Doing, Done, and Archived, in addition to the ability to delete the task.

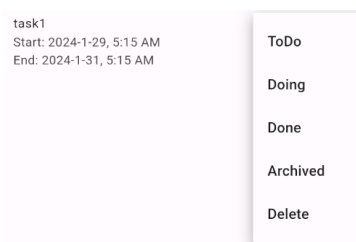


Figure 5.34: Tasks Status

The user can add new task with the following details: title, description, start date / time, end date / time, and an option if he wants to set a reminder for a specific date and time.

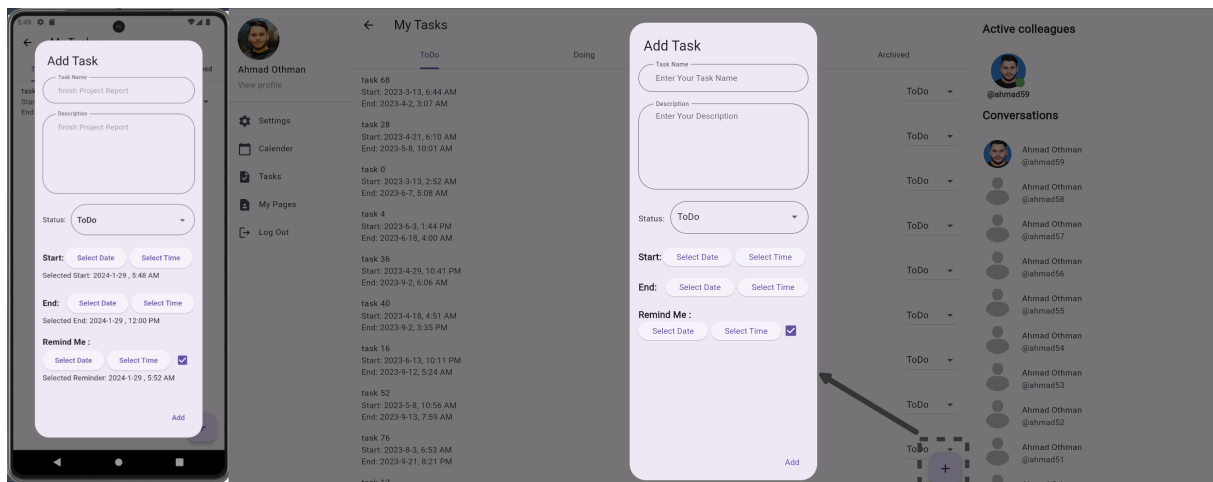


Figure 5.35: Add New Task

The user can set a time for the reminder, as in the calendar, and a pop up notification will appear as follows:

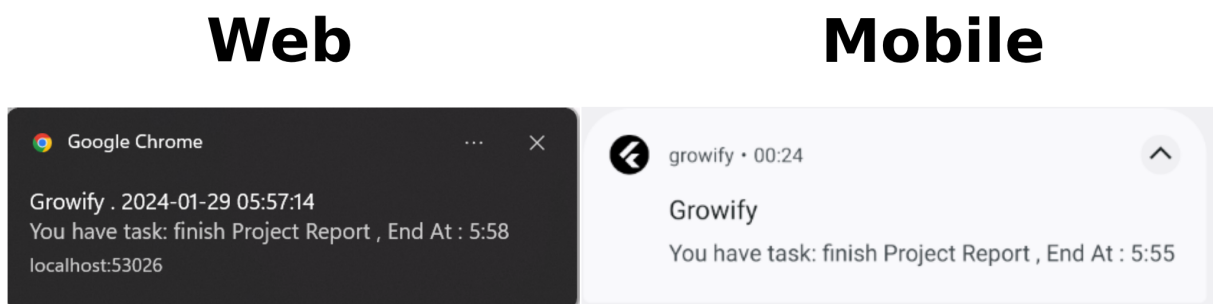


Figure 5.36: Tasks Notification

### 5.5.8 User Profile

The user can access his personal page mainly by clicking on his personal picture on the left side of the screen.



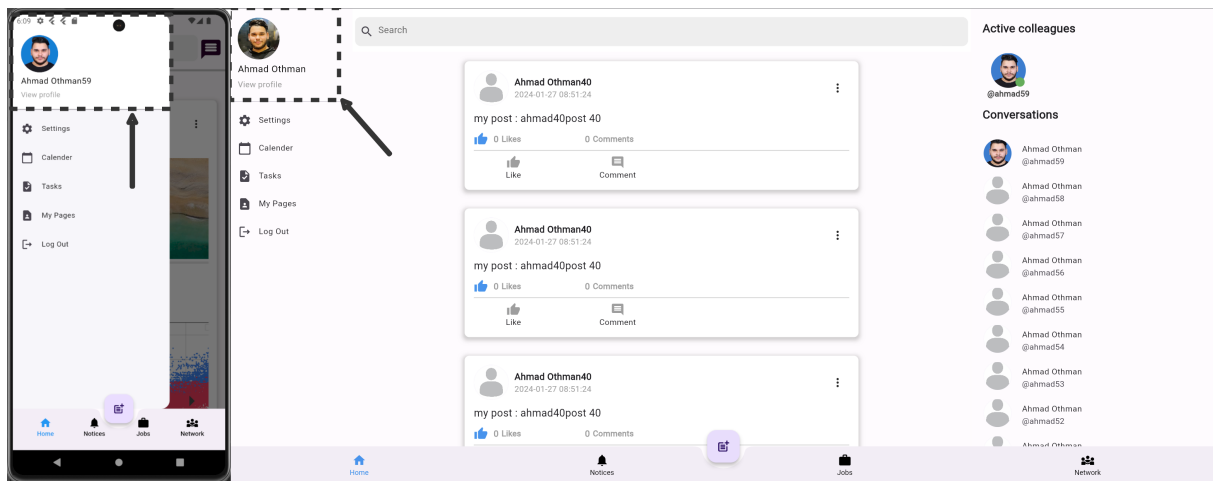


Figure 5.37: Main User Profile Button

After that the user routed to his profile page

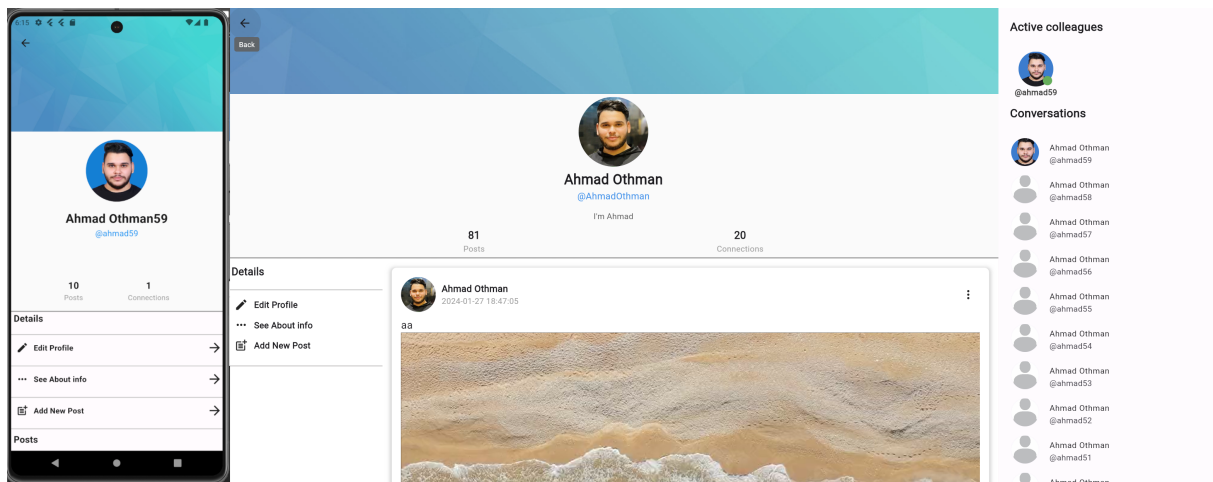


Figure 5.38: User Profile

On this page, the user can see: his cover photo, profile photo, name, username, bio, number of posts and number of connections, a button to edit profile information that routes him to the 5.19 settings page, a button to see user details, a button to publish a new post, and then his posts, where they are displayed by loading 10 posts every time user scrolls to the bottom of the page.

## See About Info

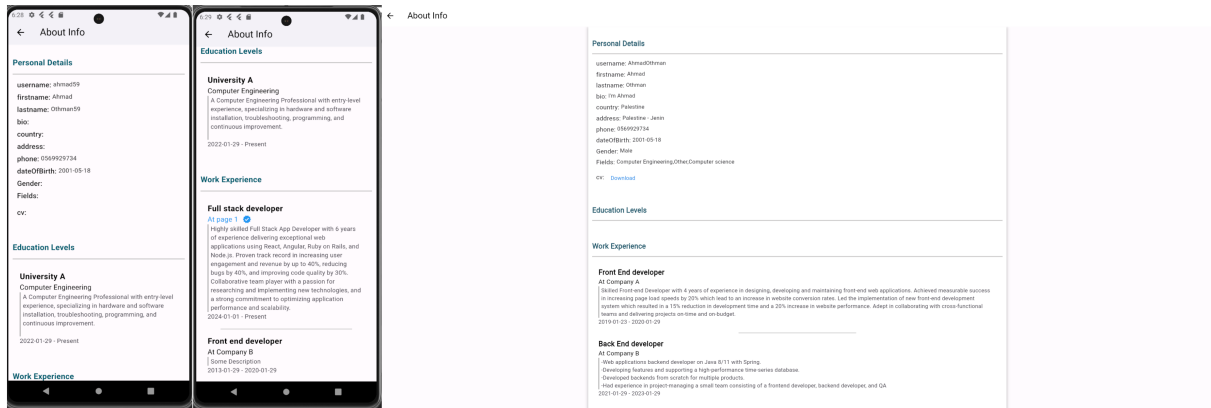


Figure 5.39: User Profile See About Info

On this page, the user can display all his details that he adds through the settings interface, which the rest of the users on the site can see. If user has uploaded a CV, he can download it from this page, and if the user has added the page's ID in addition to the specialty on a page where he is an employee a verification mark will appear next to the name of the page he works on. This feature contributes to improving the users' experience by verifying the validity of the information that the user provides.

## Add new Post

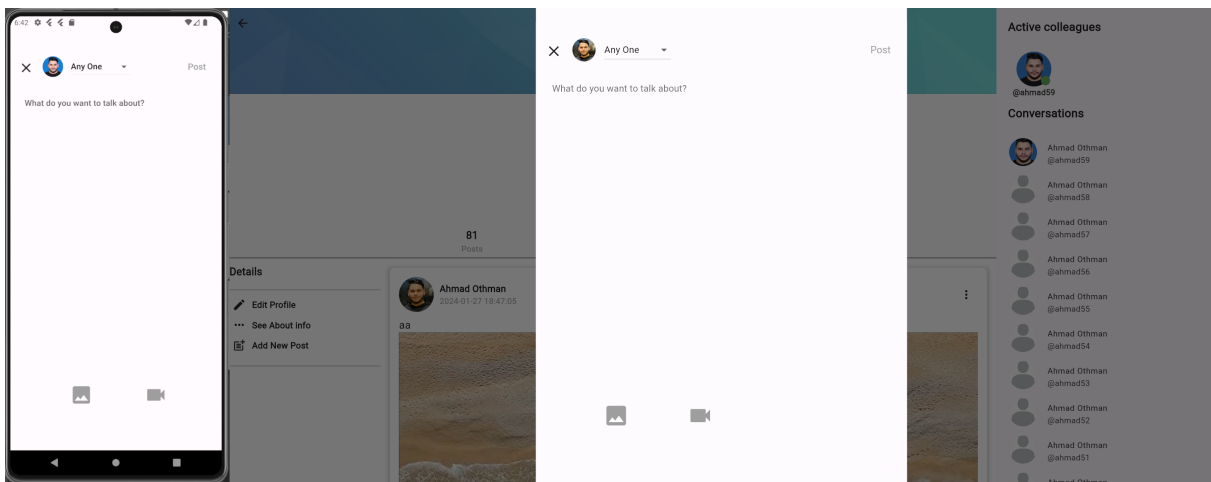


Figure 5.40: Add New Post

In this page, user can add new post, adding new post can be accessed through Add new post button on profile page or on the home screen, user can set the post privacy to Any One or Connections only, then user can add text/multiple images/multiple videos (one of them at least, or all of them at the same time), the post date/time recorded by the server.

### 5.5.9 Posts

The publishing posts feature provides the ability to publishing opinions, achievements, ideas, etc. to other users on the site. This feature is available for both the user and page, where it is possible to control a large number of functions that will be explained in this section.

#### Post in all possible body designs

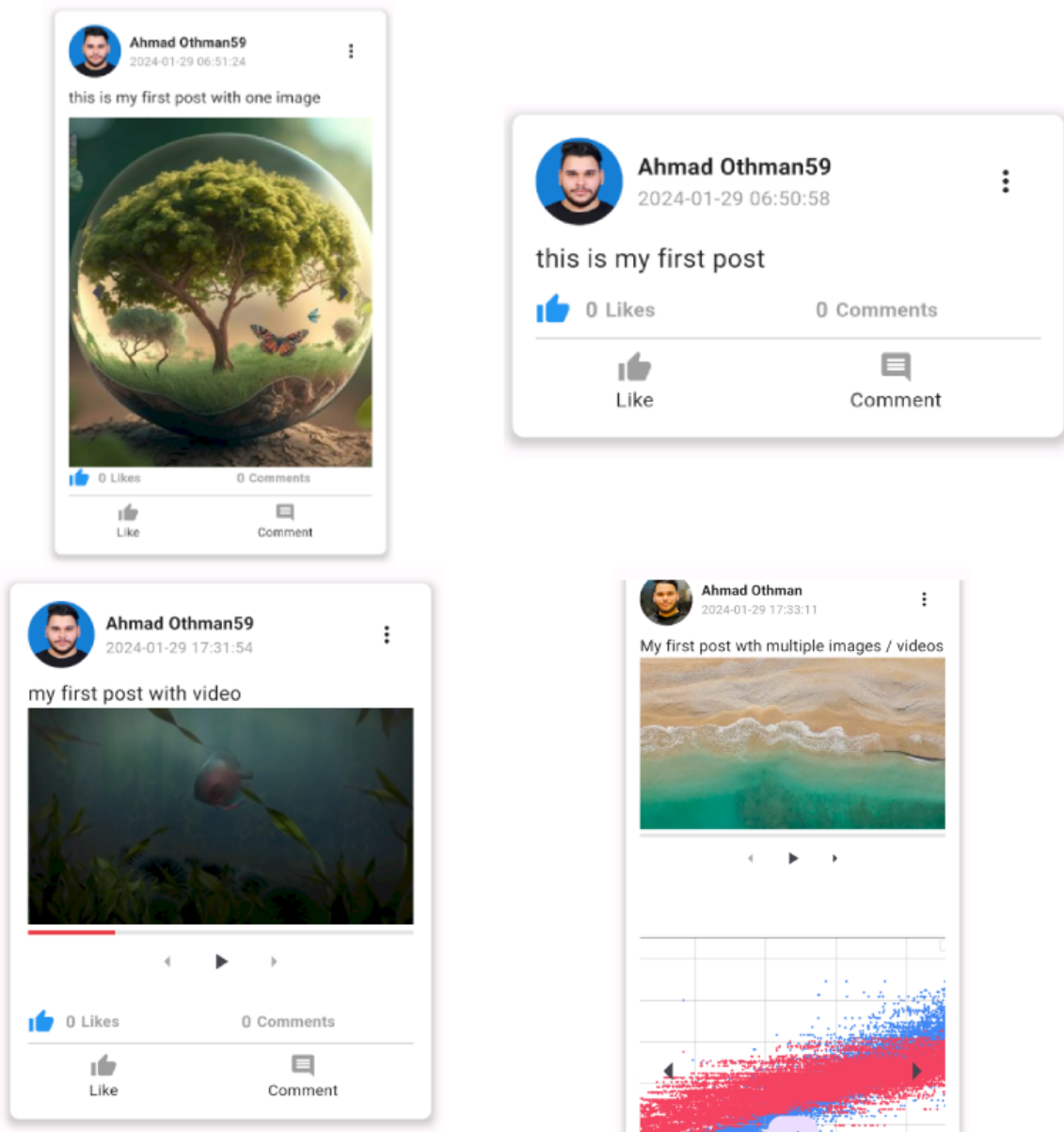


Figure 5.41: Post in all possible body designs

As we mentioned above, user can publish a post with text only, or a video only/a number of videos, or an image only/a number of images, or two of the options together, or all of them.

When there are a number of videos/images, user can move between them using these buttons:

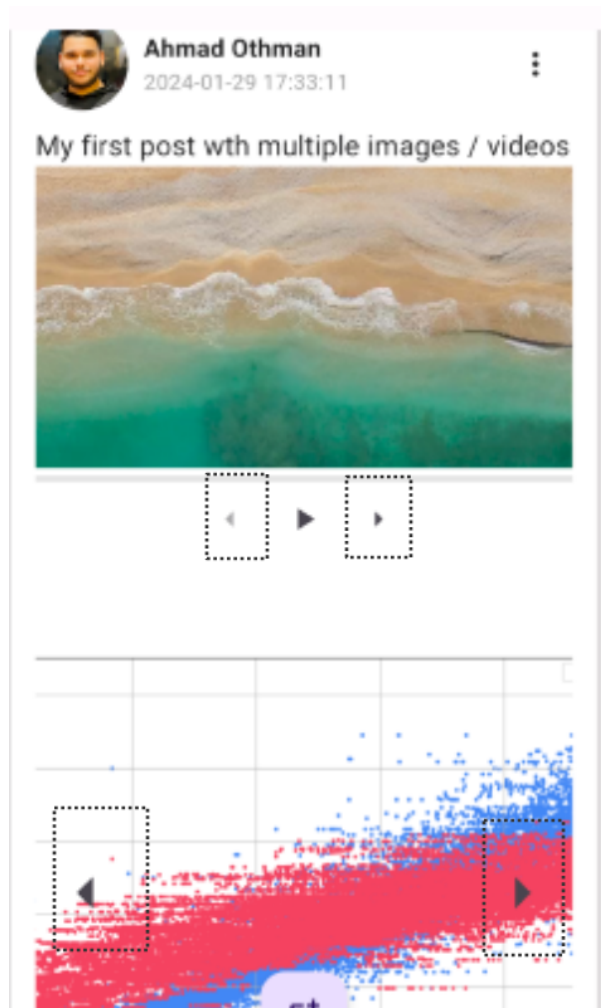


Figure 5.42: Post multiple images and videos

When the user/page is a post creator, he can edit the post/delete the post/see the edit history through the three dots in the upper right corner. When the user sees posts belonging to another creator, he can only see the edit history or Report the post.

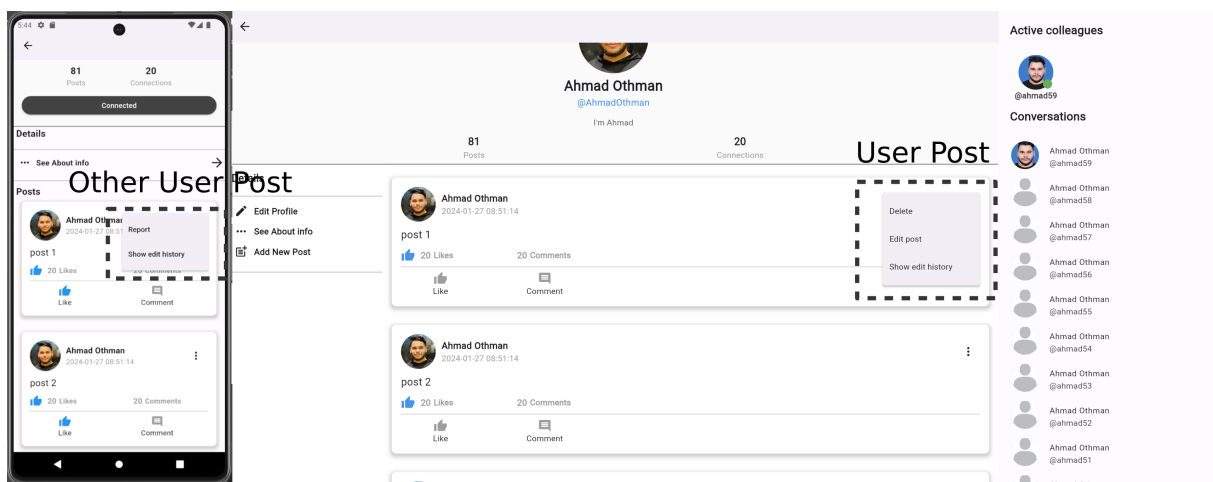


Figure 5.43: Post Options

## Post Options

- Delete Post: The post will be permanently deleted from the database.
- Edit post: user can edit the text of his post/ the privacy (Any One/Connections), page can edit the text only, the post date will be changed to the edit date, and the old date will be stored in the edit history page.

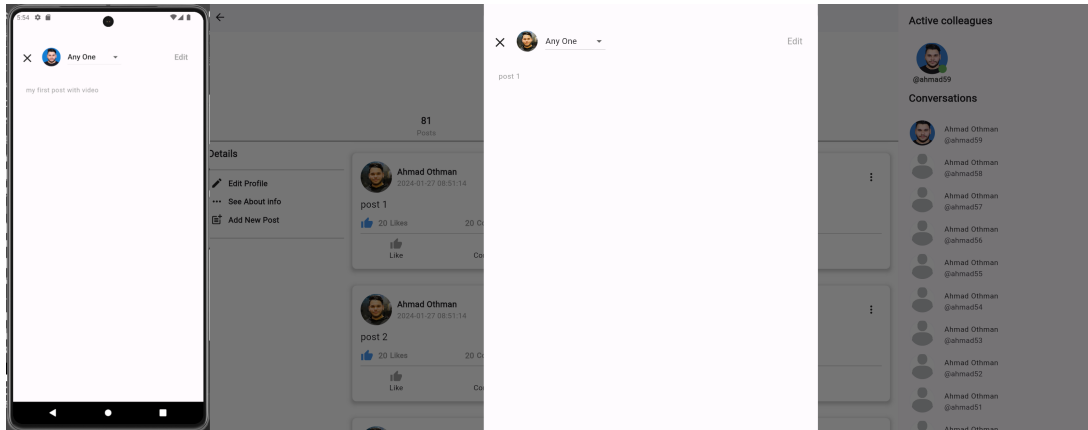


Figure 5.44: Edit Post

- Edit History: user can see his/other users/pages post's edit history.

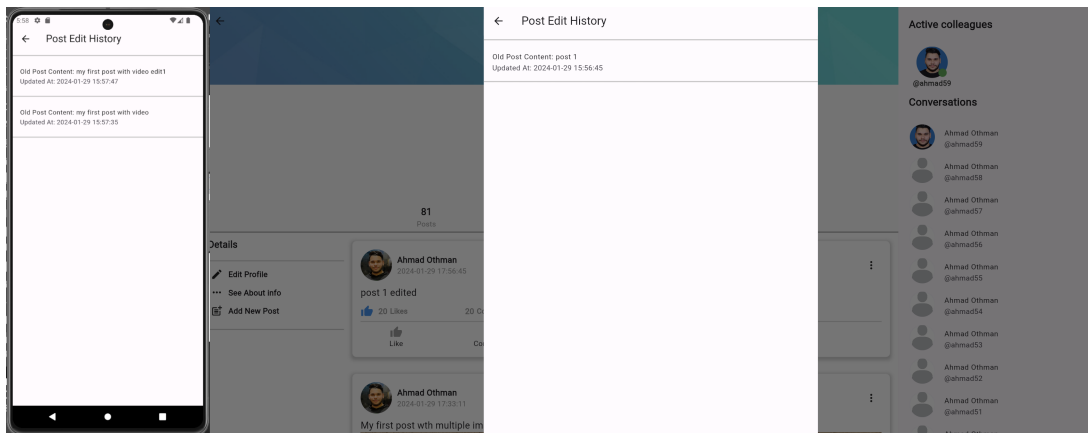


Figure 5.45: Post Edit History

- Report post: user can report other user's/page's posts, where the system admin can see reported posts from the admin panel and then take the necessary measures such as deleting the post/deleting the user.

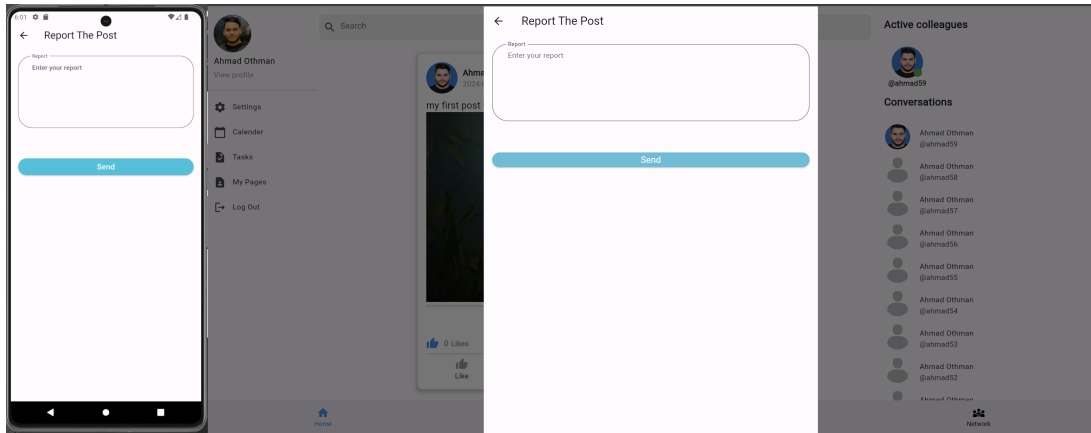


Figure 5.46: Report Post

## Post comments

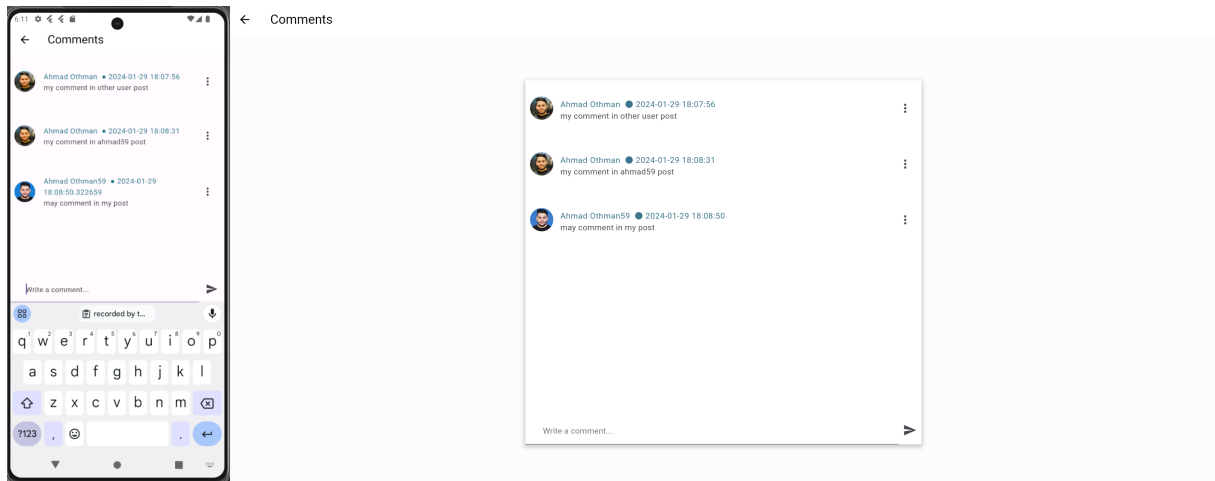


Figure 5.47: Post comments

User can see post's comments, and add a comment to his post/other users posts (taking into account post privacy)/page's posts, when user clicks on profile photo of other user/-page, he will be routed to the user/-page profile.

when user publish new comment on other user's post, a notification will be sent to other user (will be explained in details in Notification section).

## Comment options

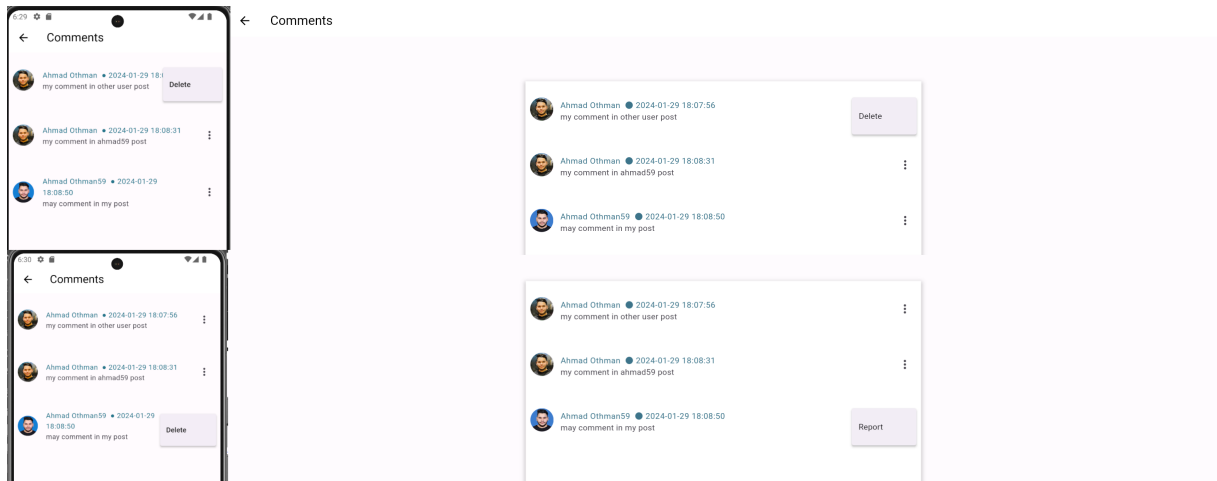


Figure 5.48: Comment Options

User/page can delete any comment in his post or his comment in others posts, if the user/page is not the comment creator or the post creator, he can report the comment only.

report option will rout the user to the report comment page where the system admin can see reported Comments via admin panel and then take the necessary measures such as deleting the comment/deleting the user.

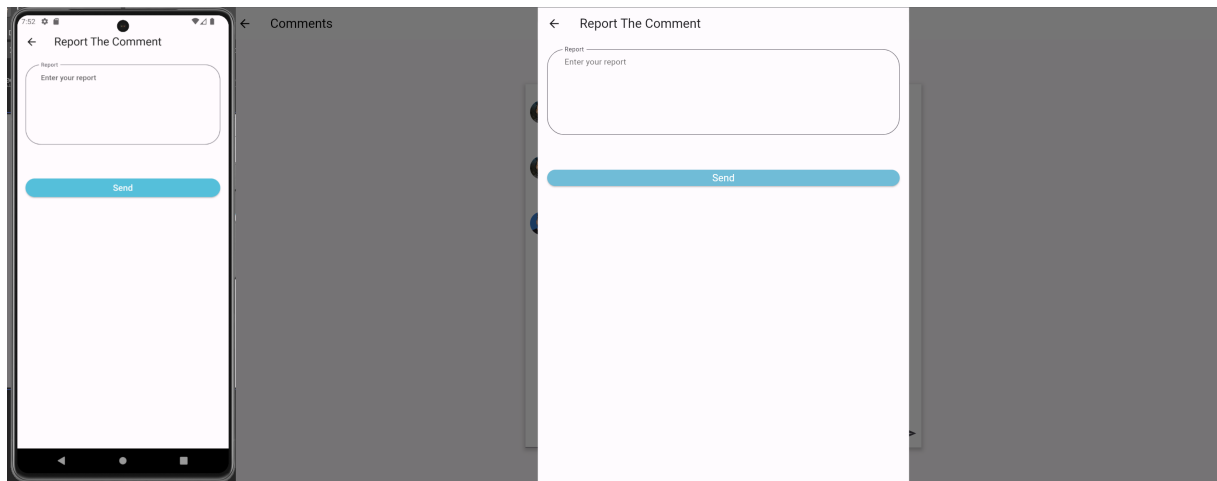


Figure 5.49: Comment Report

## Post likes

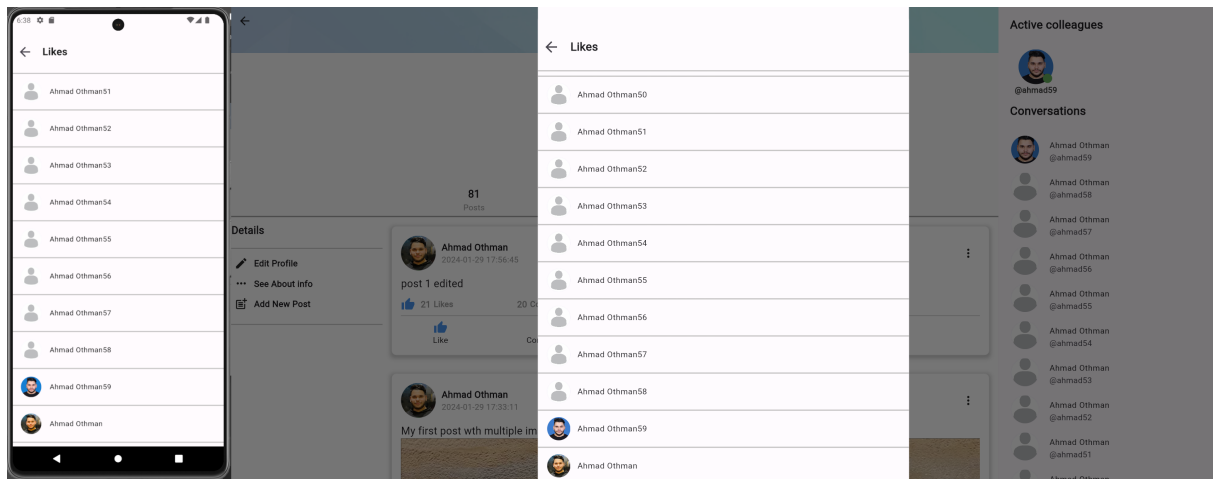


Figure 5.50: Post Likes

User can view/add/remove like on his post/other users posts (taking into account post privacy)/page's posts, when user clicks on profile photo of other user/page, he will be routed to the user/profile page.

When user add new like on other user's post, a notification will be sent to other user (will be explained in details in Notification section).

### 5.5.10 Other users profiles

Other users' pages can be accessed through a large number of routes, the most important of which are: search, the user's photo in the post, comments, likes, job applications.

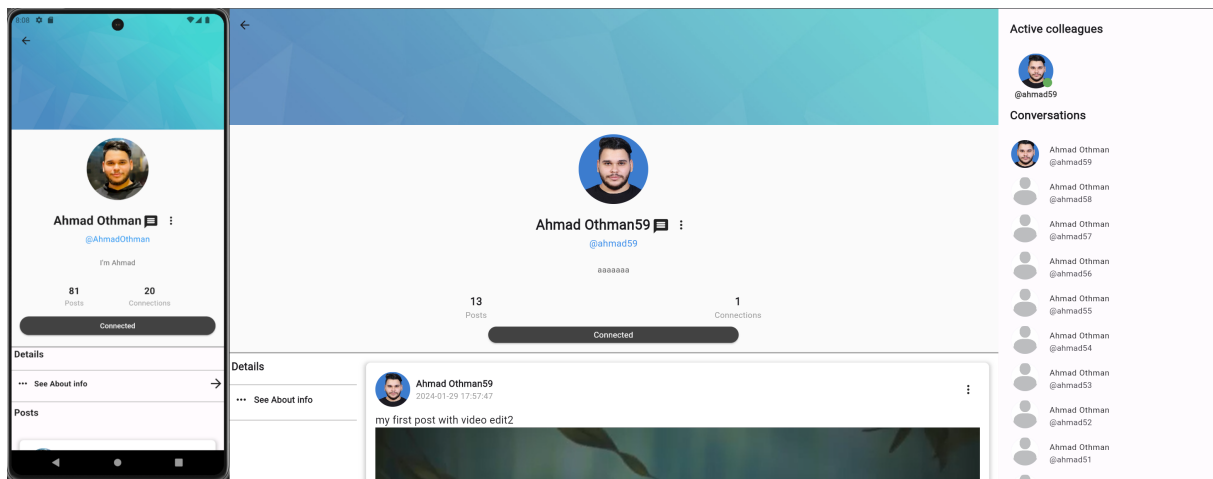


Figure 5.51: Other Users Profile

From here user can see the other user's cover image, photo, name, username, bio, number of posts / connections, user posts (10 every time user scroll to the bottom of the page), see about info.



If the user is in other user's connections, he can see all of his posts, if not, he can see only the "Any One" posts.

### Types of connections

- Connect: user can send a connection request to other user, a notification will be sent to other user that he has a new connection request (will be explained in details in Notification section).

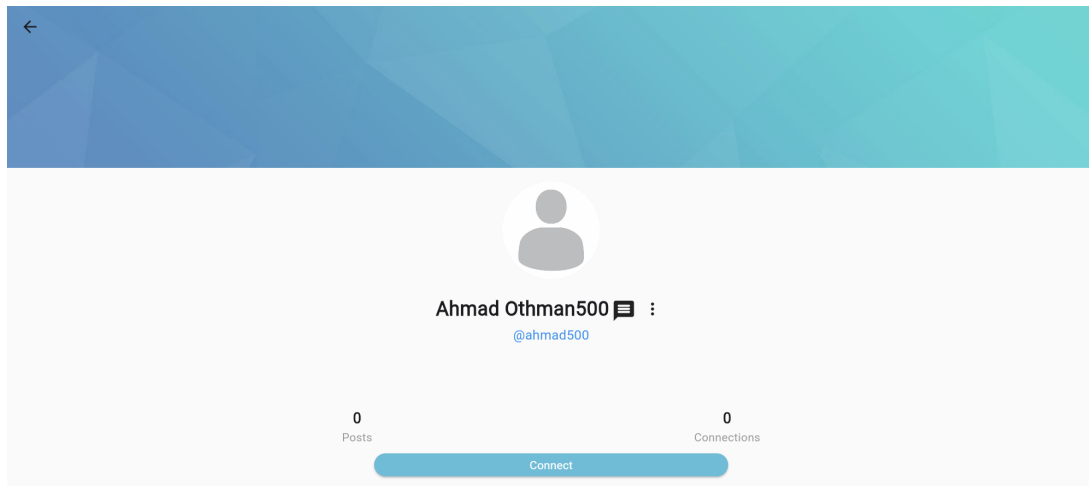


Figure 5.52: connect

- Accept / Delete connection request: user can accept or delete a connection request from other users, if user accepted a connection request, a notification will be sent to the other user (will be explained in details in Notification section).

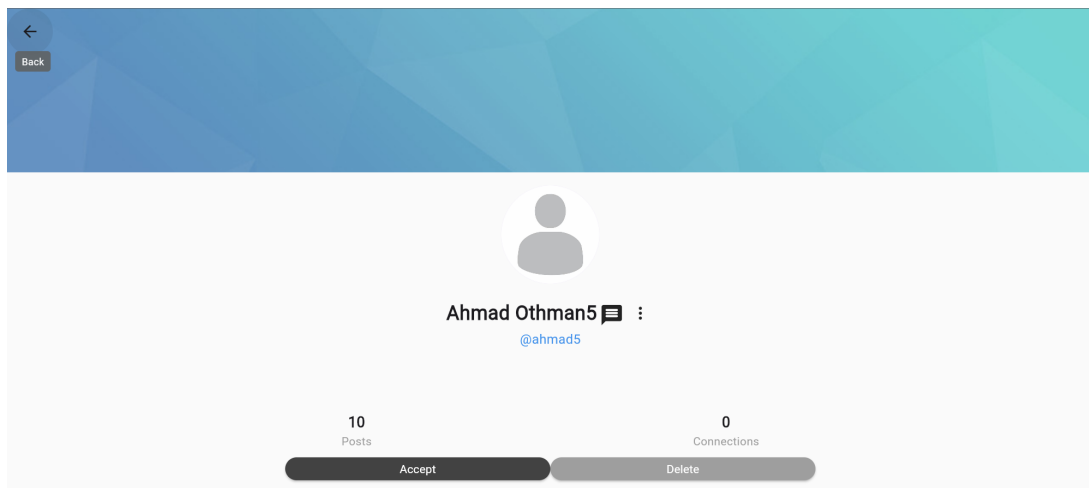


Figure 5.53: Accept / Delete connection request

- Requested: when user send a connection request to other user, the connect button will be converted to Requested, user can press on the Requested button to remove the connection request, and the connection request notification will be deleted for the other user.

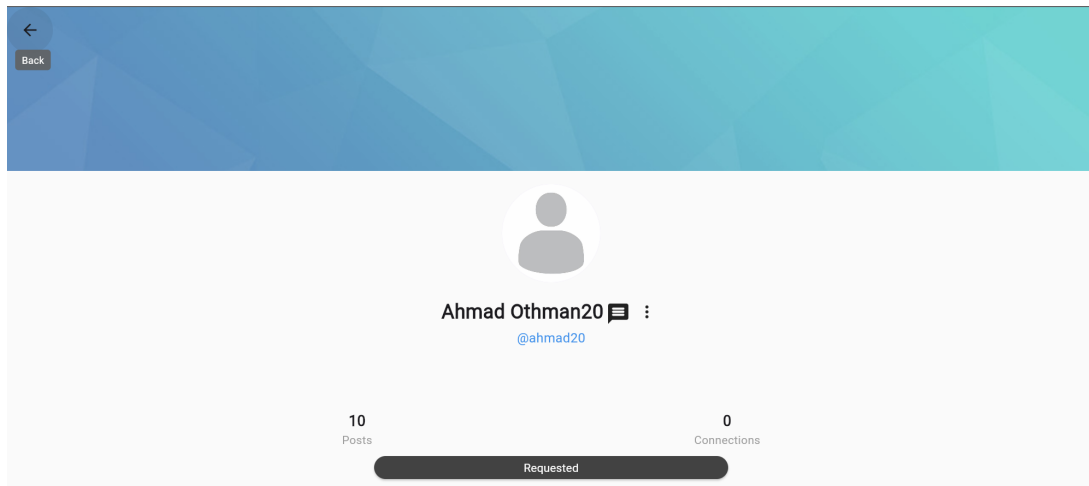


Figure 5.54: Requested

- **Connected:** when user accept the connection request, both users now are on each other connections, also user can press on Connected button, and a pop up dialog will appear to confirm removing the user from connection, if user approves then both users are not connected now and user need to send a new connection request again if he want to reconnect.

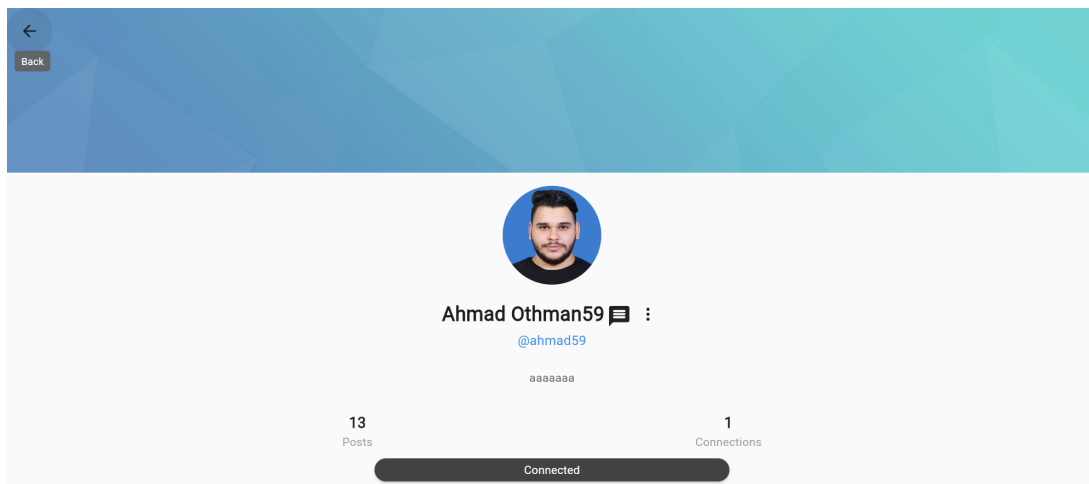


Figure 5.55: Connected

## Report user

user can report other users via the three dots next to the user name, report option will rout the user to the report user page where the system admin can see reported users via admin panel and then take the necessary measures such as deleting the user.

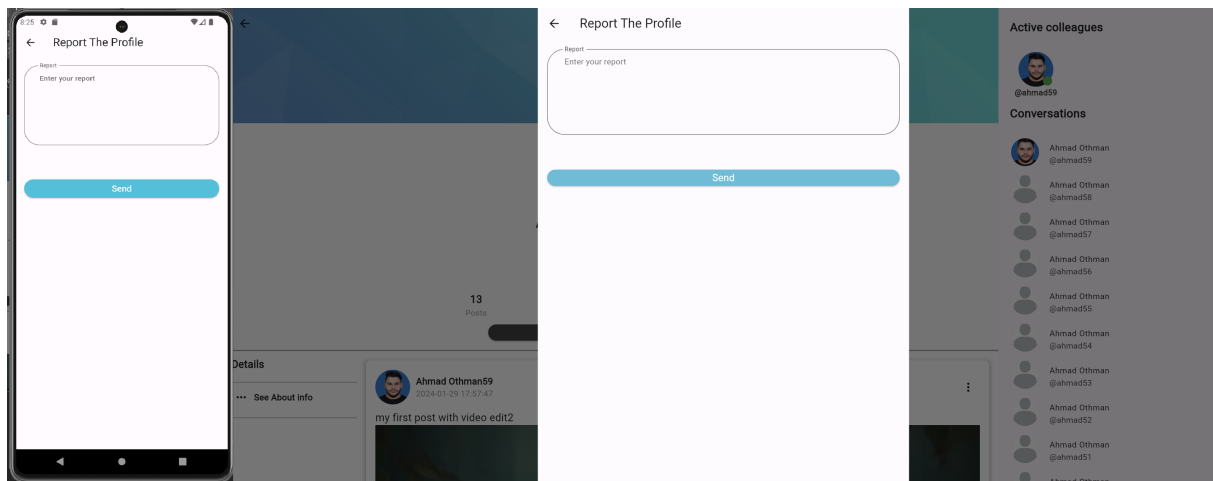


Figure 5.56: Report User

## See about info

user can see the full information of other users as its appear from user profile

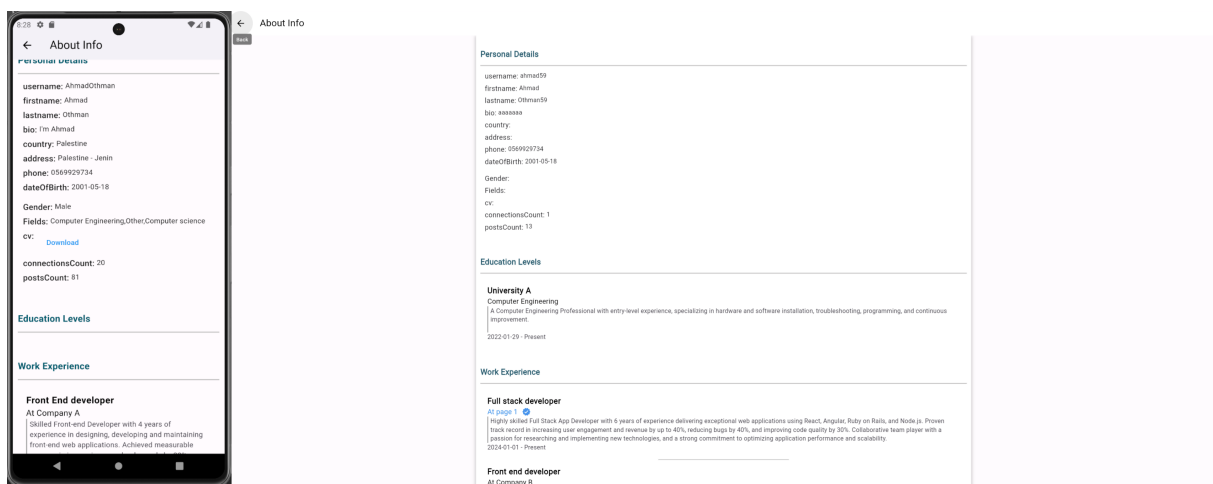


Figure 5.57: Other users see about info

### 5.5.11 Chat

User can access chat with other users/Pages via the message button next to user/page name, or if user have previous messages with other user/page, he can access the chat through the chat list in the right side of the screen.

In the chat screen user can see his active connections and his previous messages,

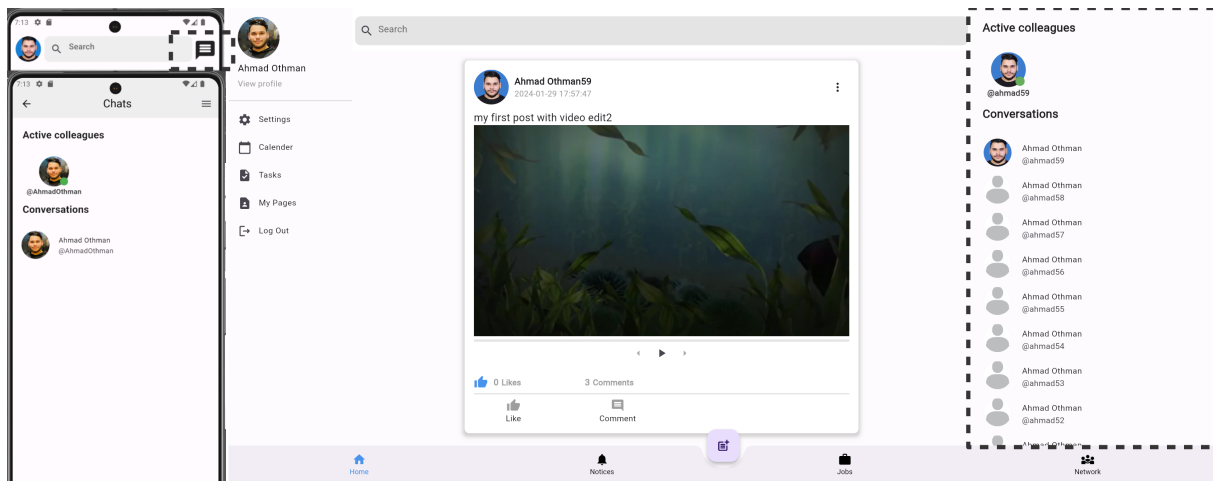


Figure 5.58: User Chats

When user press on a chat, he will be routed to messages page and a socket will be opened with the server, If the user exits this screen the socket will be closed.



Figure 5.59: Users Chat messages

## User messages

User can send text/image/video on of them or all at the same time, the message will sent via sockets on Server B, if the two users are in the chat page (so both sockets are opened), the message will appear directly for both, if the other user is not in the chat page, a notification will be sent to other user (will be explained in details in Notification section).

## User Audio call

User can call other users via audio only when pressing on the audio call in the top right corner, when pressing on the button, SDP offer will be sent to other user via sockets, if the other user in the same chat page, a calling request will appear directly, if not, a notification will be sent to the other user that routes him to the calling screen ( for both cases, a ringing tone will be played for the duration of the call request "25 second").

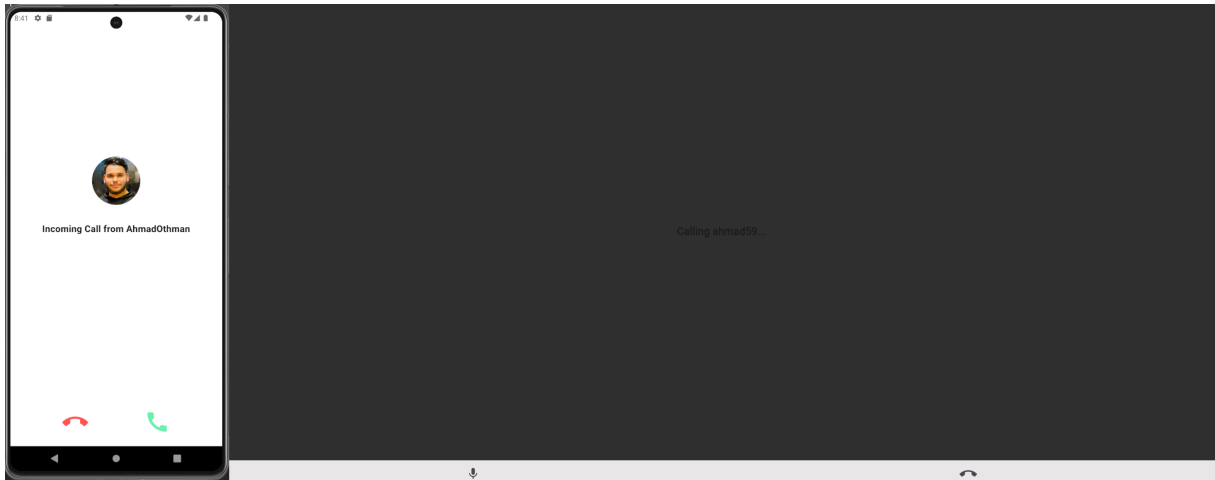


Figure 5.60: Users Audio Call Request

if the other user didn't respond for the call, the call for both users will be declined after 25 and the ringing tone will stop.

if the other user declined the call, the call will be end for both users before the 25 seconds timer ends.

if the other user accept the call, SDP Answer will be sent to other user via socket, and both users exchange the ICE candidates, then users can mute mic or leave the call from the bottom bar, if one of the user leaved the call, the call will be ended for both users.



Figure 5.61: Users Call Accepted

### User Video and Audio call

User can call other users via Video and Audio when pressing on the video call in the top right corner, when pressing on the button, SDP offer will be sent to other user via sockets, if the other user in the same chat page, a calling request will appear directly, if not, a notification will be sent to the other user that routs him to the calling screen ( for both cases, a ringing tone will be played for the duration of the call request "25 second").



Figure 5.62: Users Video Call Request

if the other user didn't respond for the call, the call for both users will be declined after 25 and the ringing tone will stop.

if the other user declined the call, the call will be end for both users before the 25 seconds timer ends.

if the other user accept the call, SDP Answer will be sent to other user via socket, and both users exchange the ICE candidates, then users can mute mic/ turn off camera/ switch camera (for mobile only)/ share screen / record call (from web only) / leave the call from the bottom bar, if one of the user leaved the call, the call will be ended for both users.



Figure 5.63: Users Video Call Accepted

### Share screen

One user or both users can share screen at the same time.

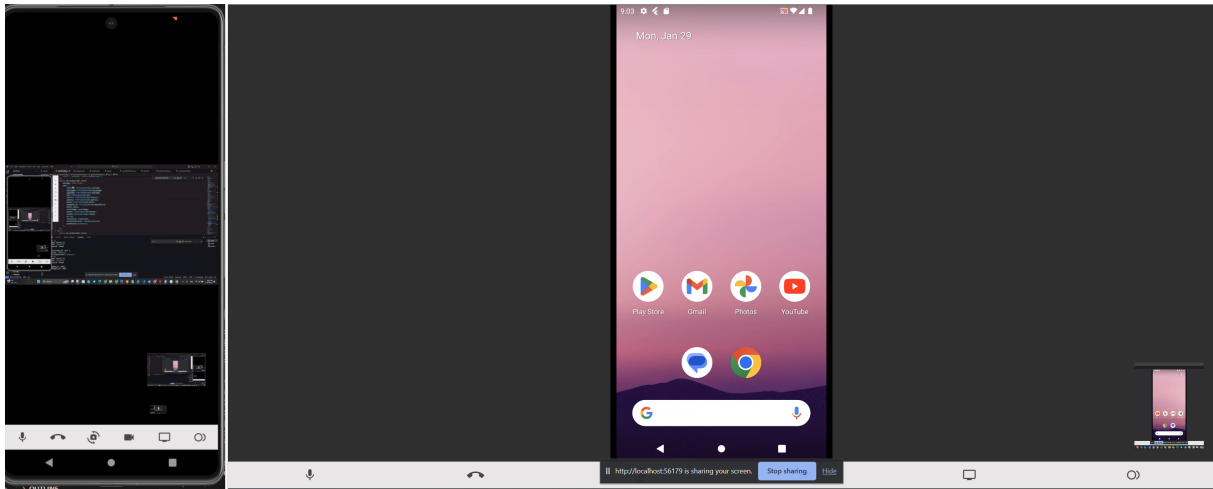


Figure 5.64: Video Call Share Screen

## Calling Process

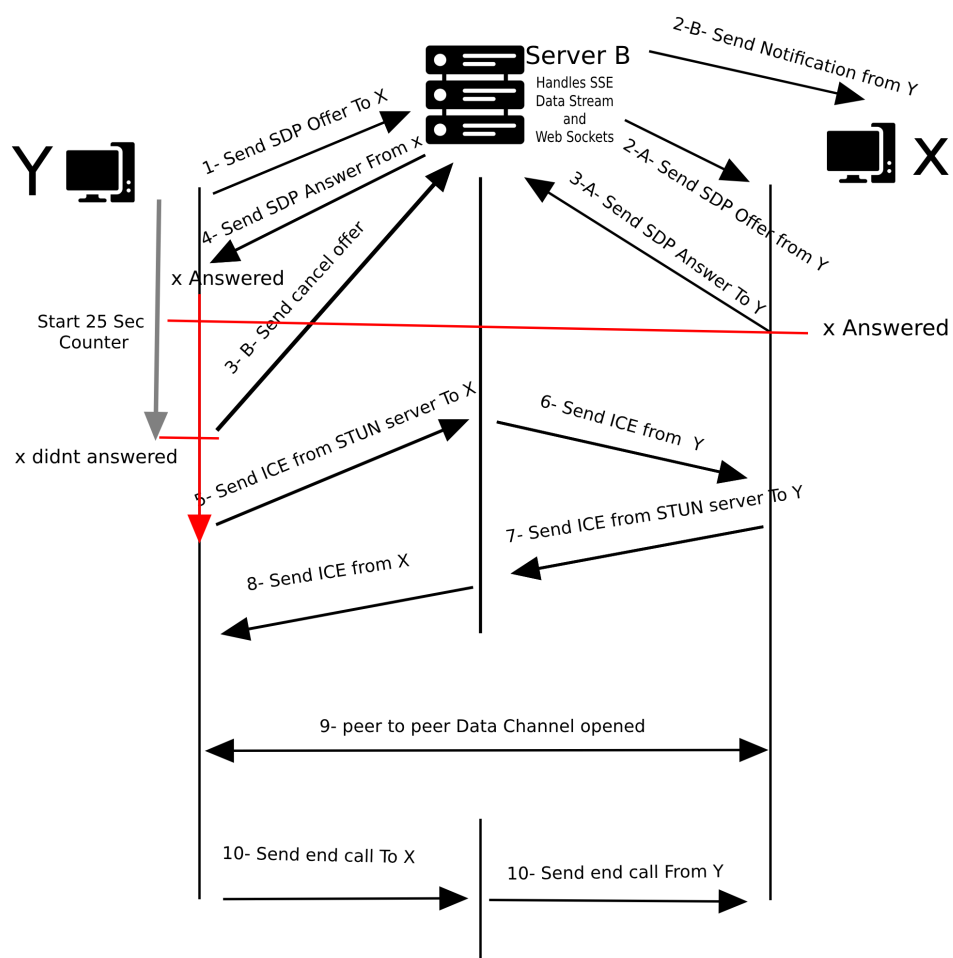


Figure 5.65: Call Process

### 5.5.12 User pages

Our application allows users to create pages representing the organizations/companies they manage. User pages can be accessed by clicking on the My Pages button on the right side of the screen.



Figure 5.66: Other Users Profile

### Create new page

User can create a new page from the button on top right corner

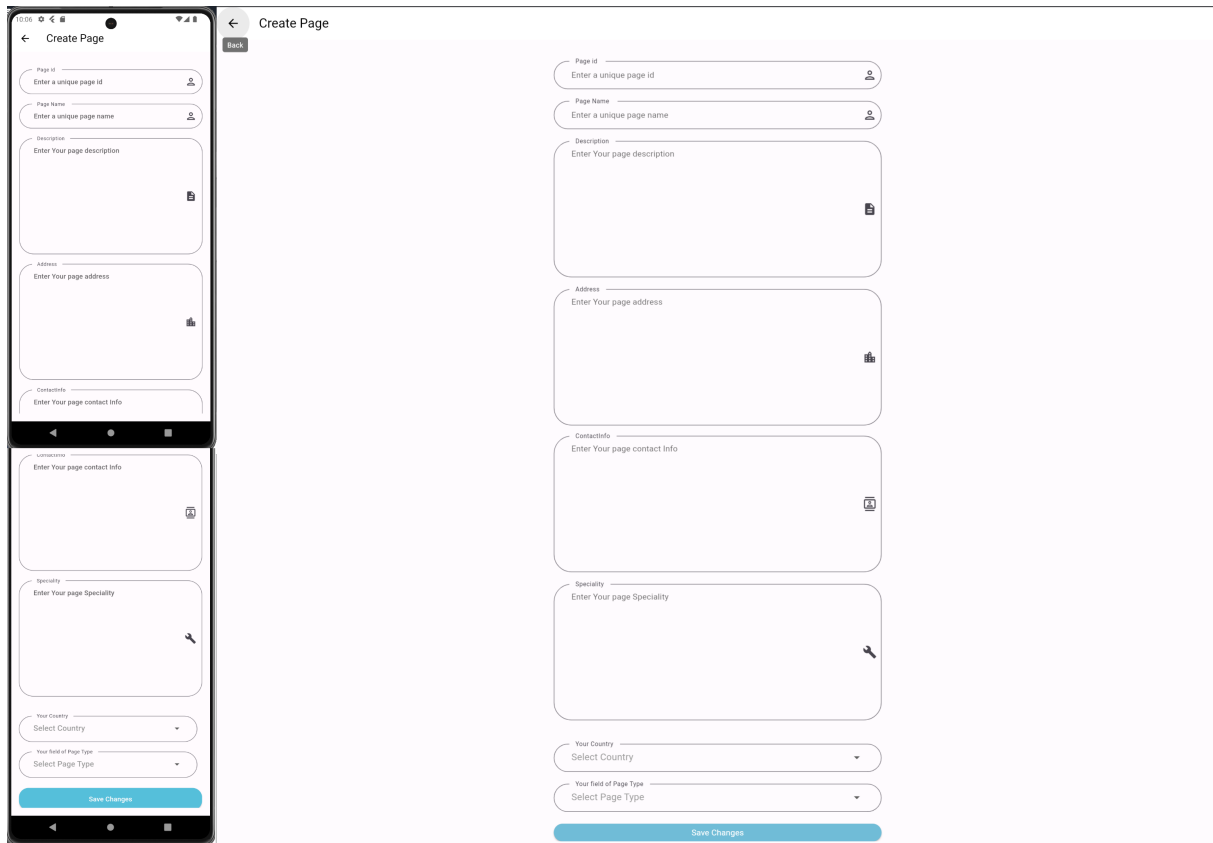


Figure 5.67: create new page



user can enter the main page information, and later can add more information from the page settings screen.

## Page types

There are two types of pages that user can create:

- **Public page:** The public page usually represents an organization/company, where any user can access the public pages through search. The public page has a large number of features, the most important of which are: adding/deleting/editing admins, adding/deleting/editing employees, creating groups, displaying/posting jobs, viewing the applications for jobs with the ability to communicate with the applicant and see his profile and create a meeting for the interview, add/delete public events (any user can view them), view chats, publish posts, and users can follow them.

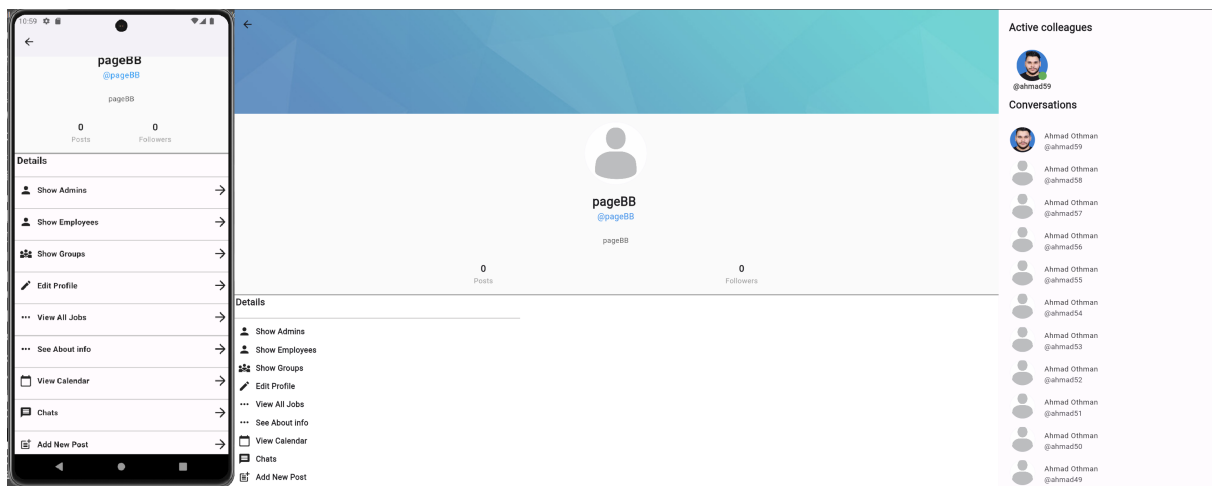


Figure 5.68: Public Page

- **Private page:** Our application provides a way to create a private page that contains a smaller number of features, in order to allow to a group of users who have a special project that they want to organize and they want to take advantage of the features that we provide for the pages to create a private page that is not searchable, and cannot be followed by other users, cannot create Jobs, cannot receive or send messages.

Examples of uses of this feature: graduation projects, university course projects, private projects, etc. Of course, the user can create a page of the type he wants and for any purpose he wants

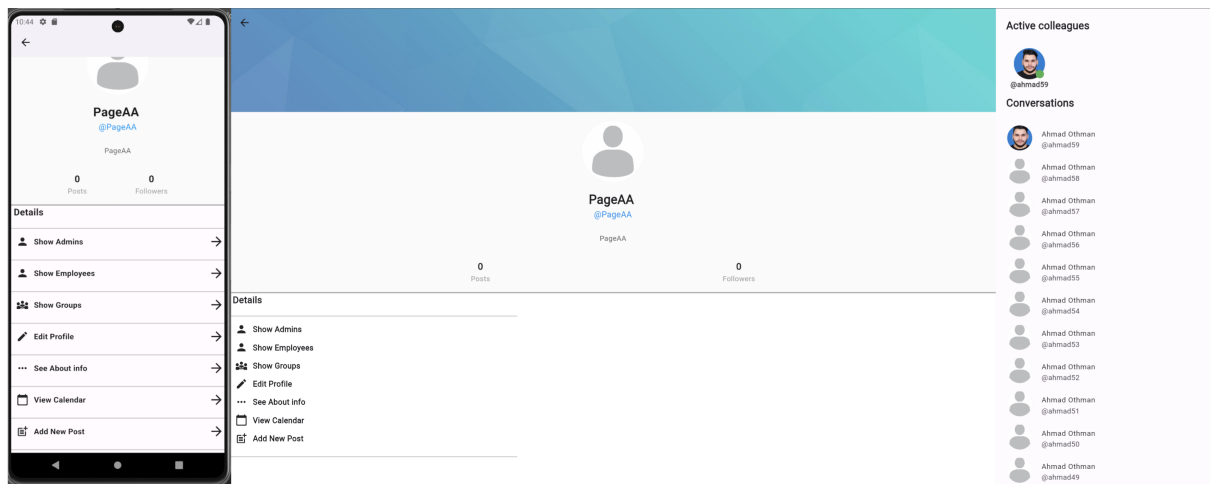


Figure 5.69: Private Page

## Page Admins

Admin can see page admins from the list in the page profile:

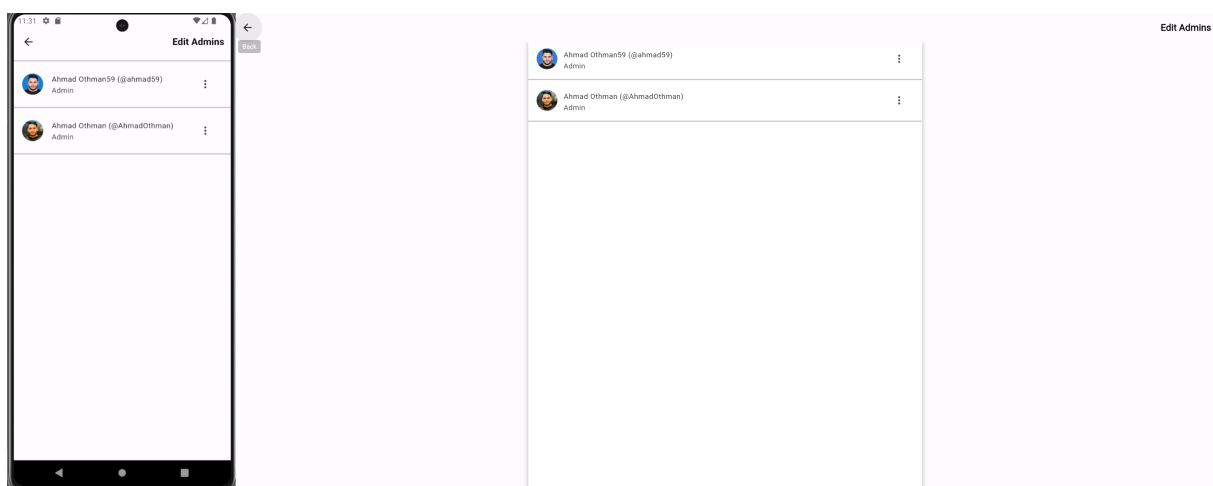


Figure 5.70: Page Admins

Admin can add a new admin/edit existing admin type from the top right corner:

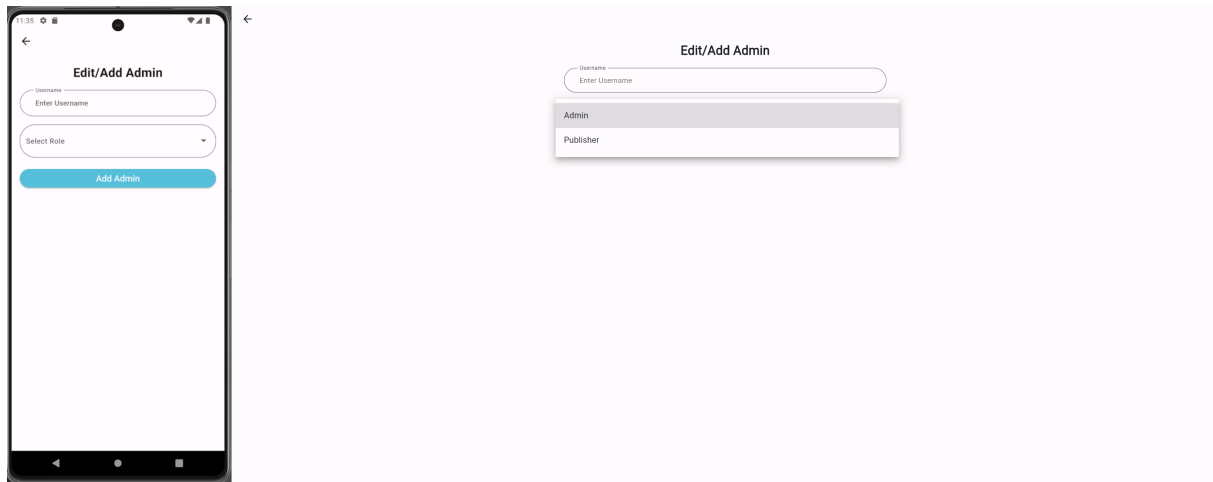


Figure 5.71: Add New Admin

There are two types of admins:

- Admin: can reach and control any information in the page like the page creator.

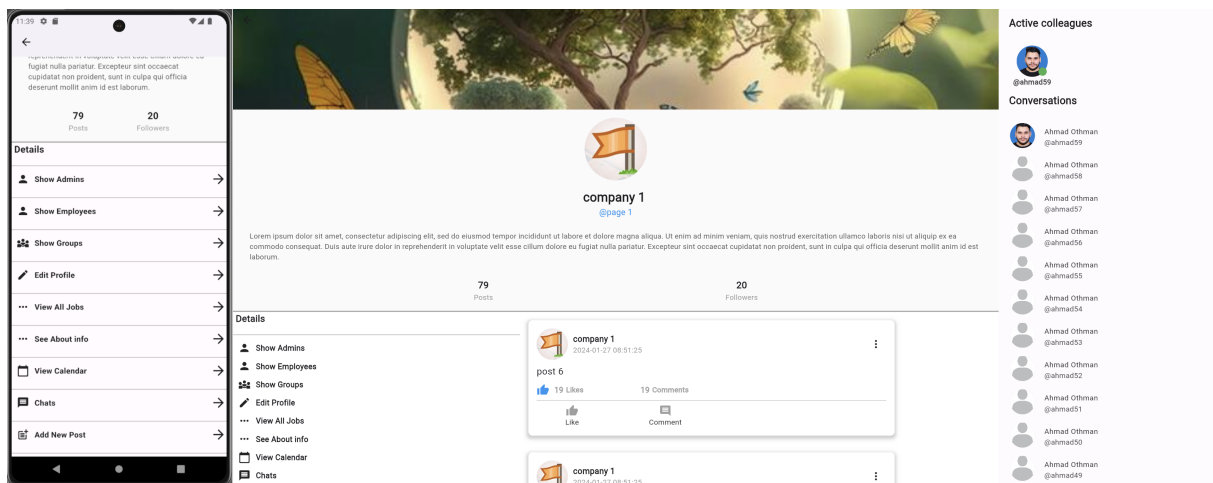


Figure 5.72: Admin View

- Publisher: The publisher can only access these features: view page information, view/add/delete events on the calendar, reply to messages, publish posts/comment/like with the name of the page.

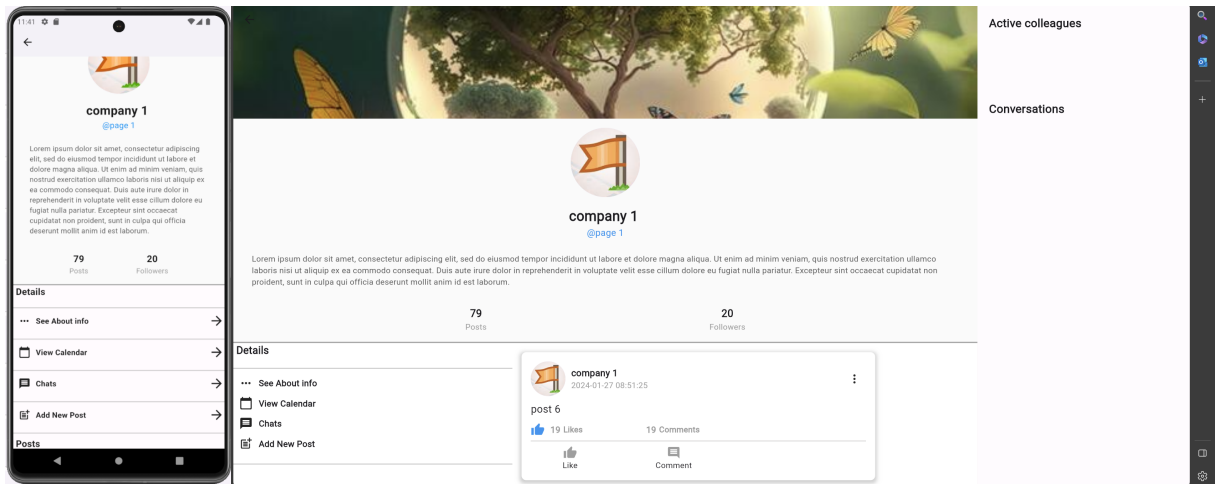


Figure 5.73: Publisher View

## Page Employees

Admin can see page Employees from the list in the page profile:

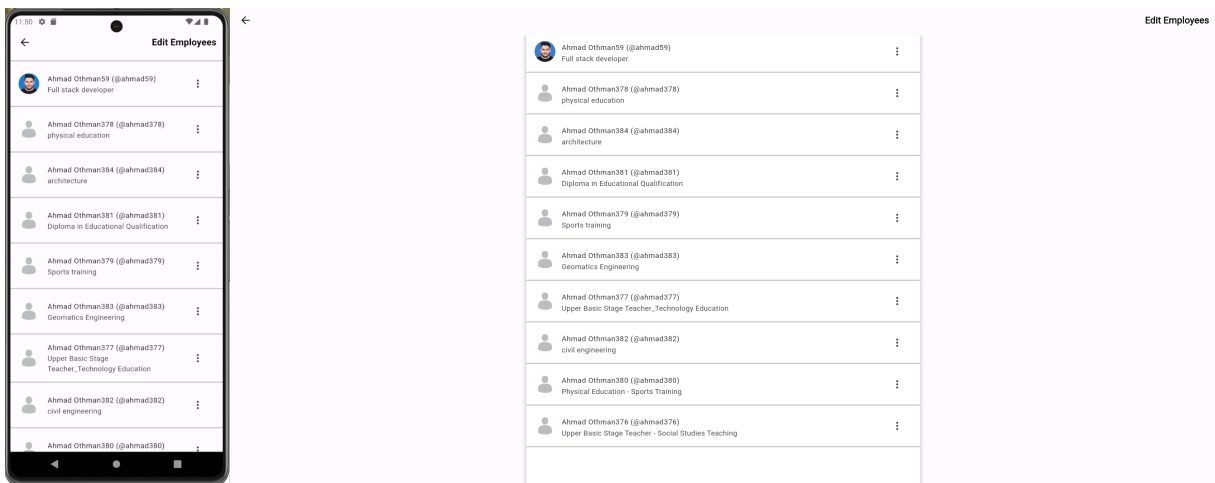


Figure 5.74: Page Employees

Admin can add a new Employee/edit existing Employee field from the top right corner:

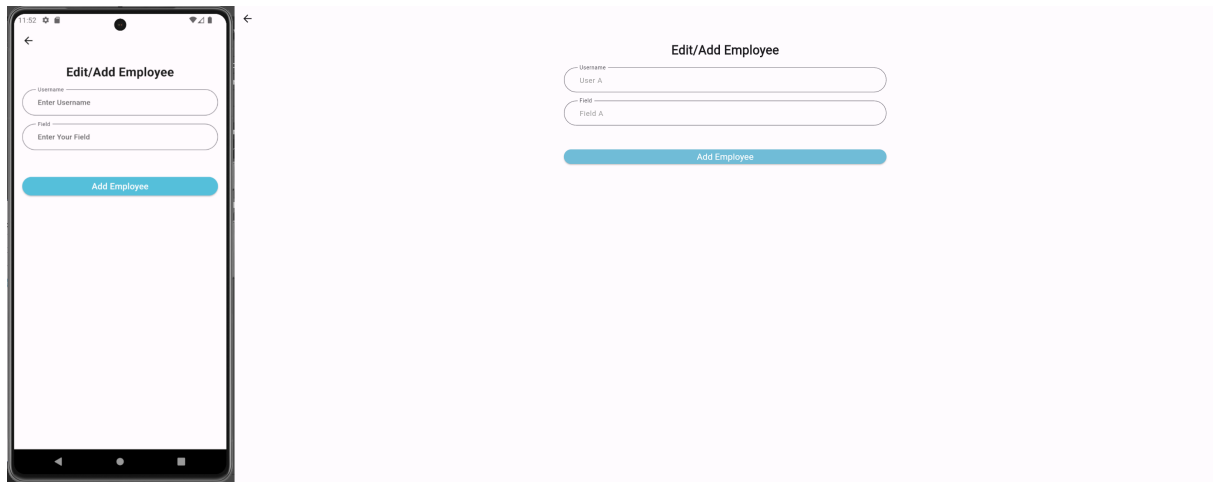


Figure 5.75: Add New Employee

When adding a new employee, the employee can add the Page ID and field that was specified for him to his Work Experience, and a verification mark will appear for all users viewing his profile next to the page, as mentioned previously.

## Edit Page Profile

Admin can see/edit page info including cover image and profile image from this interface:

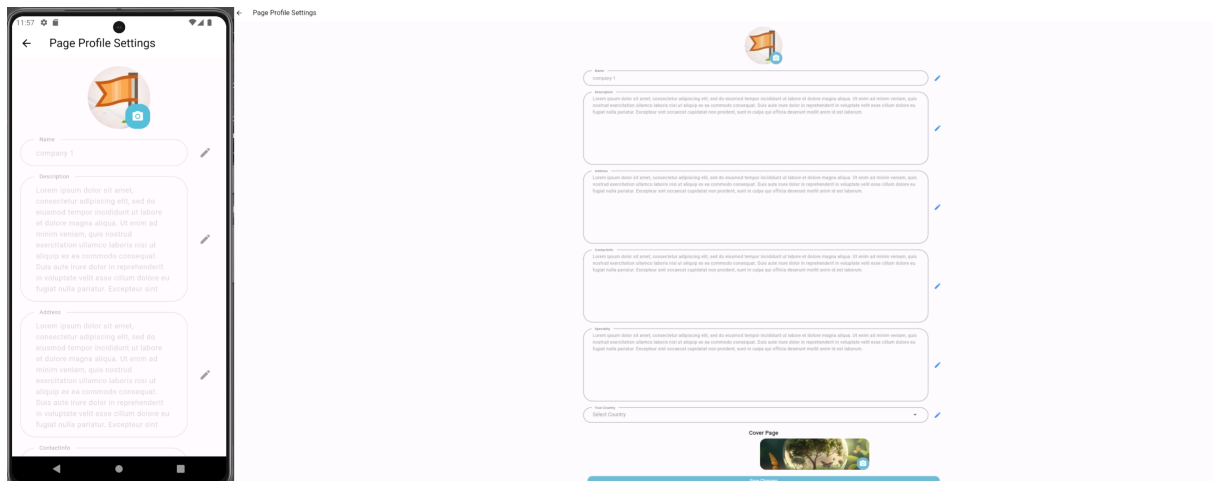


Figure 5.76: Edit Page Profile

## Page jobs

Admin can see page jobs from the list in the page profile, and can delete a job from the three dots in the upper right corner:

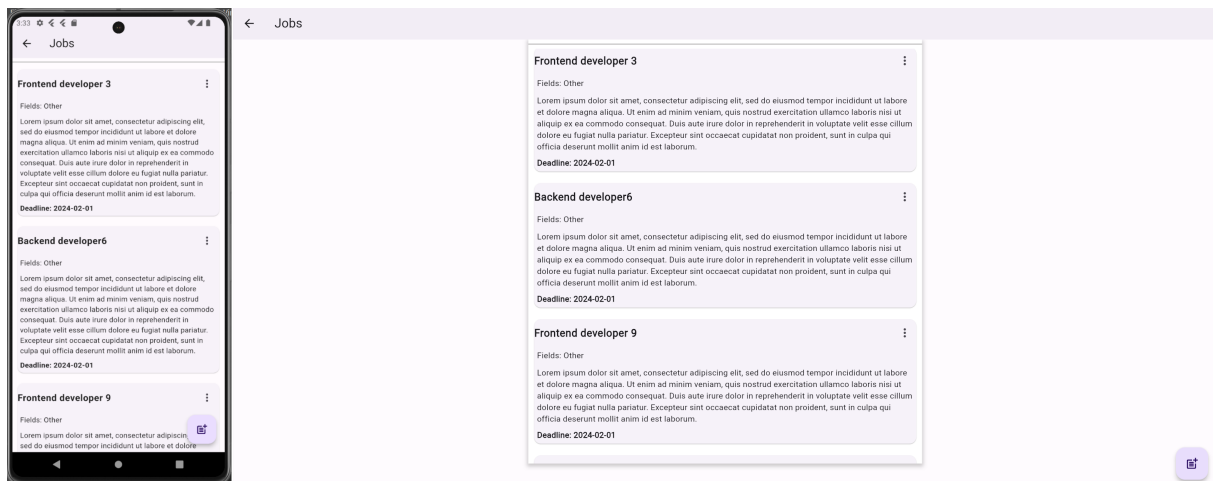


Figure 5.77: Page Jobs

Admin can add a new job from the bottom right corner:

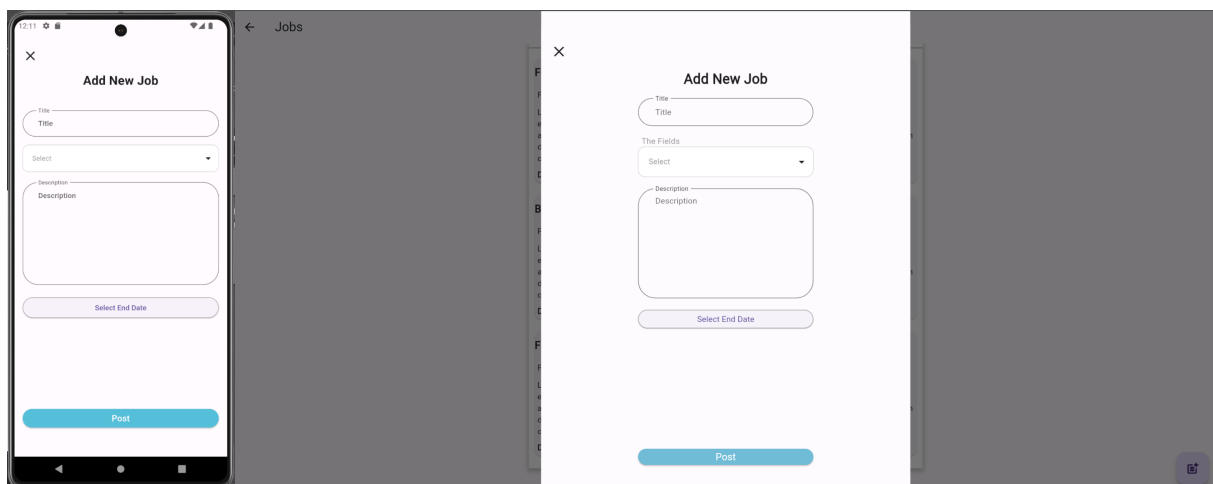


Figure 5.78: Add New Job

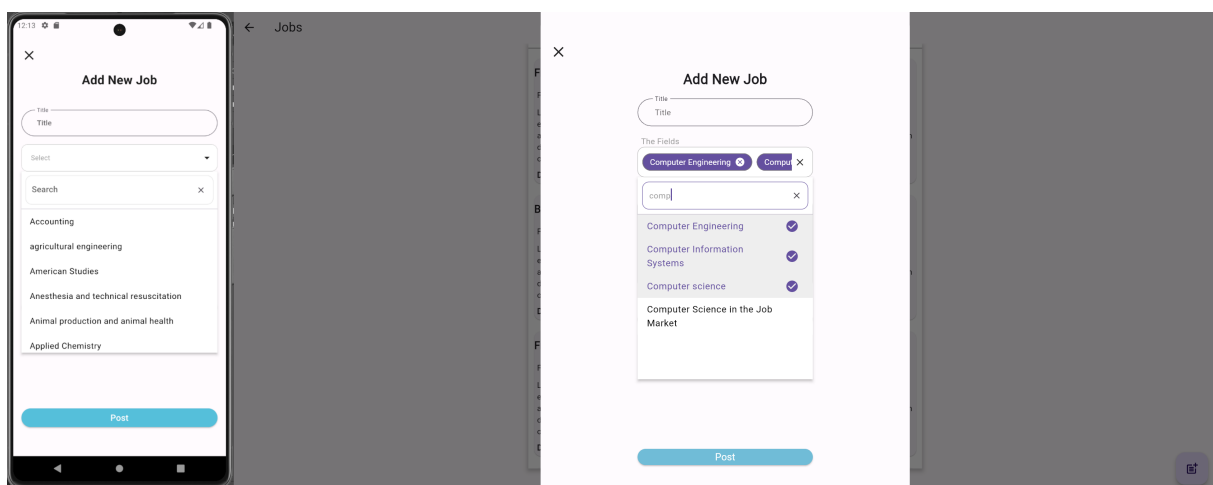


Figure 5.79: Job Fields

When the admin adds a new job, a notification is sent to all users who follow the page and at least one of their fields is similar to at least one of the fields for the new job (will be explained in details in Notification section).

## Job applications

When the admin press on a job in Page jobs, he can view the details of the job, and all job applications:

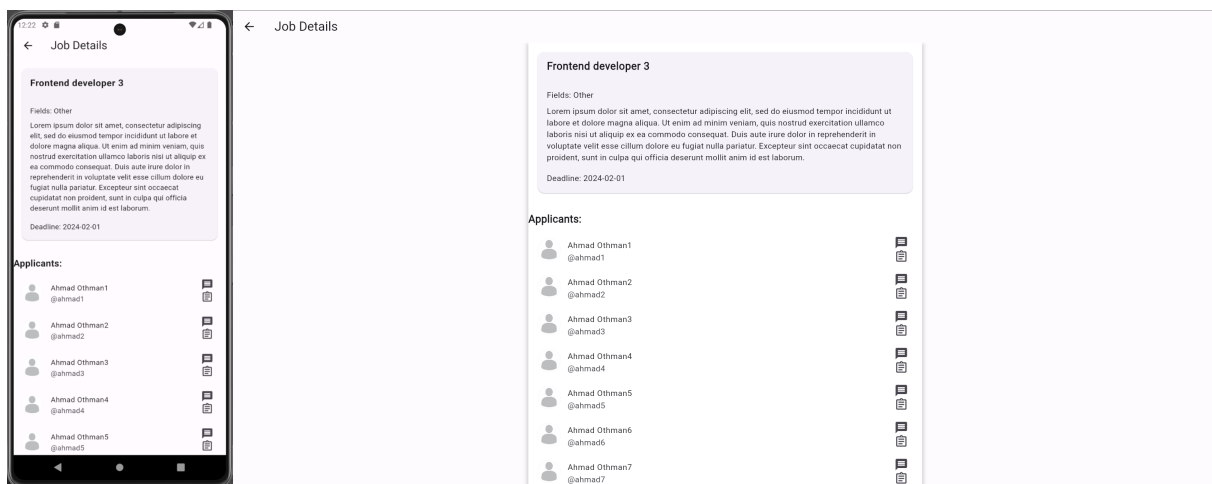


Figure 5.80: Job Applications

Admin can view job application details for a user and he can see the CV (if user Added a CV with the application) and a note (if user Added a note with the application), and also can view user profile with all of user information.

Admin can also starts a conversation with applicant and make a video/audio call for any reason, such as conducting an interview as the video call in our application provides the Share Screen feature. When the page starts conversation with a user, a notification will be sent to the user so he knows that there is a page want to contact him.

When the user opens the chat, now the user and the page have their sockets opened and can communicate in real-time.

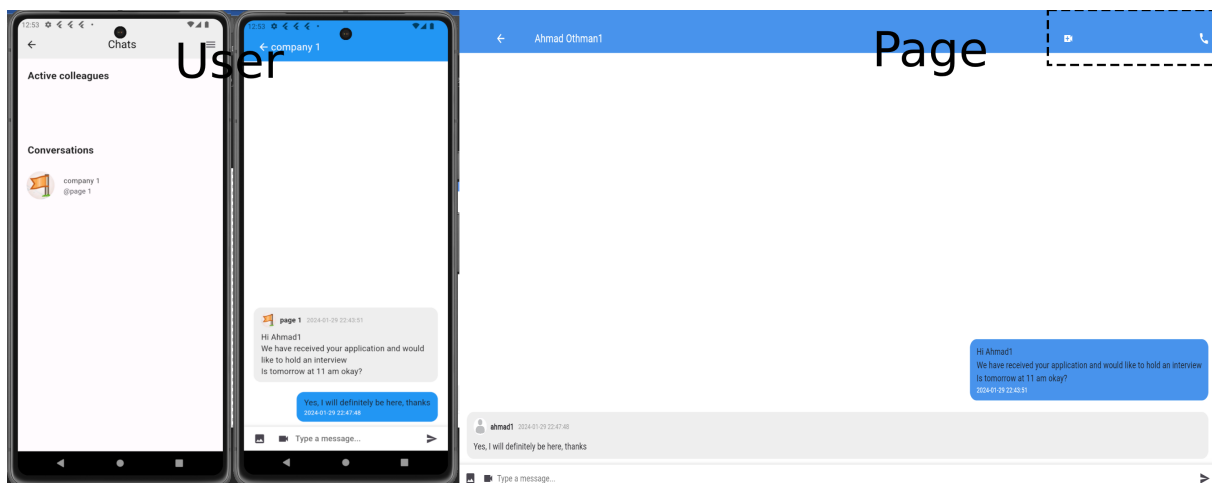


Figure 5.81: Page - User Chat

The company can start a call with the user, but the user cannot start a call with the company (to prevent spamming). Also, the user does not receive a notification when he is contacted by a page (to prevent spamming). The user can put a reminder in his calendar so that he is present in the conversation at the time that they agree on.



Figure 5.82: Page - User Call

As we mentioned previously, the page can communicate with users in only two cases: starting a conversation by communicating with job applicants, and responding to incoming messages. Thus, we can see the new conversation of the page when we click on the chat button in the side menu of the page.



Figure 5.83: Page Chats

## Page calendar

The admin can access the public page calendar through the list in the page profile, then he can see/add/delete events that all users on the application can see.



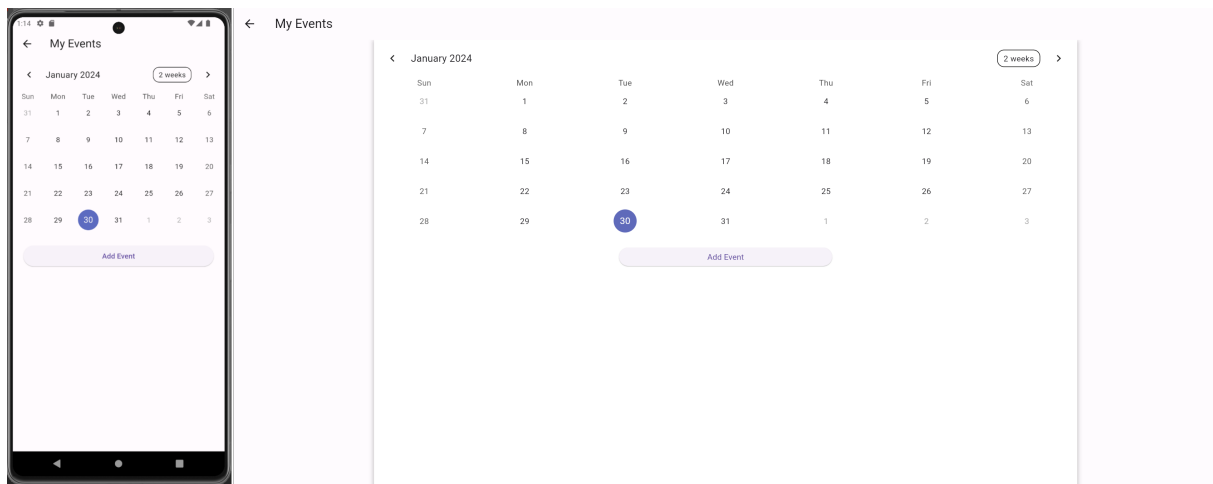


Figure 5.84: Page Calendar

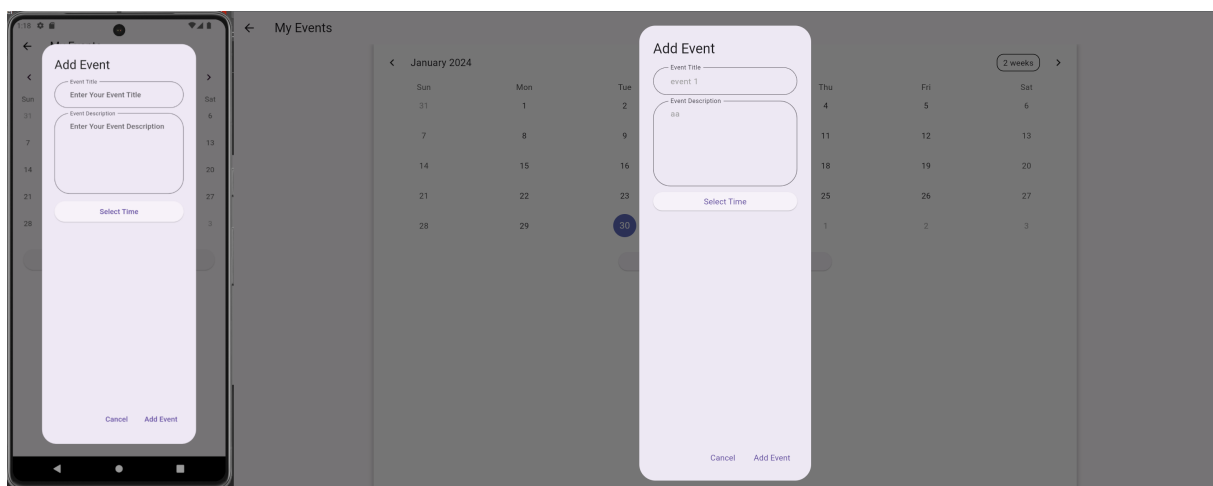


Figure 5.85: Add Event To Page Calendar

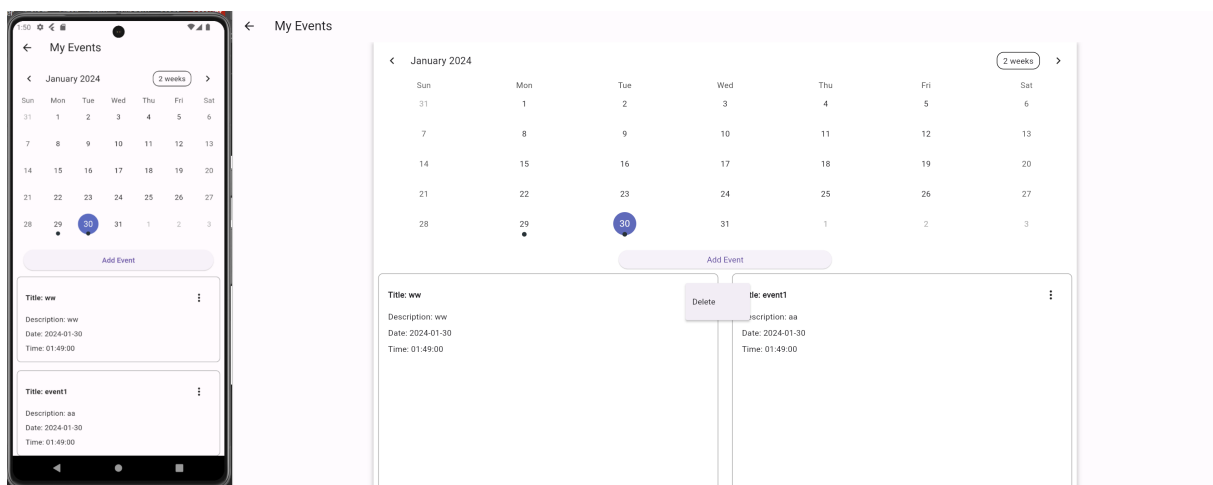


Figure 5.86: Page Event Added

## Page Posts

The admin can publish posts and add a comments/likes in the name of the page like a regular user, in addition to editing/deleting/seeing the edit history of the posts and also deleting comments on the page's posts.

### 5.5.13 Page groups

Our application allows the creation of groups with a tree structure, as groups provide many features, the most important of which are:

- Add the company's employees to the groups and divide them in a way that suits the company. It is also possible to add members from outside the company's employees who can represent: trainees, customers, etc. This feature can be used in any way that suits the company.
- It is possible to create more than one parent group (that does not have a parent group) (more than one tree), and each parent/subgroup can contain an infinite number of child groups.
- A number of admins can be assigned to each group so that they can: add tasks for members, add private events to the group, add admins/members to the group, modify group settings, start a meeting, create child groups and distribute members to them.

The admin on the page can see/control all the groups that the page owns, and the admin in the group can see/control all the groups that fall under his own group (child groups).

- Real-time conversation between group members

When the admin enters the groups page from group button in page profile list he can see all of main groups (groups doesn't have parent group)

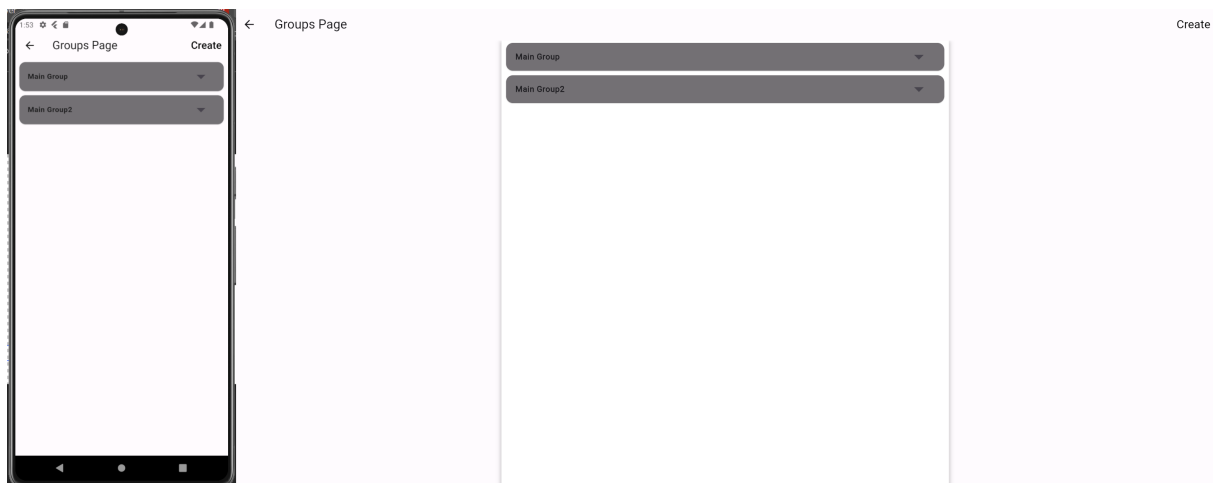


Figure 5.87: Main Page Groups

When the user press on the arrow next to the group name, he can view the child groups of this group, and if there is a child groups for the child group he can do the same thing again, etc.

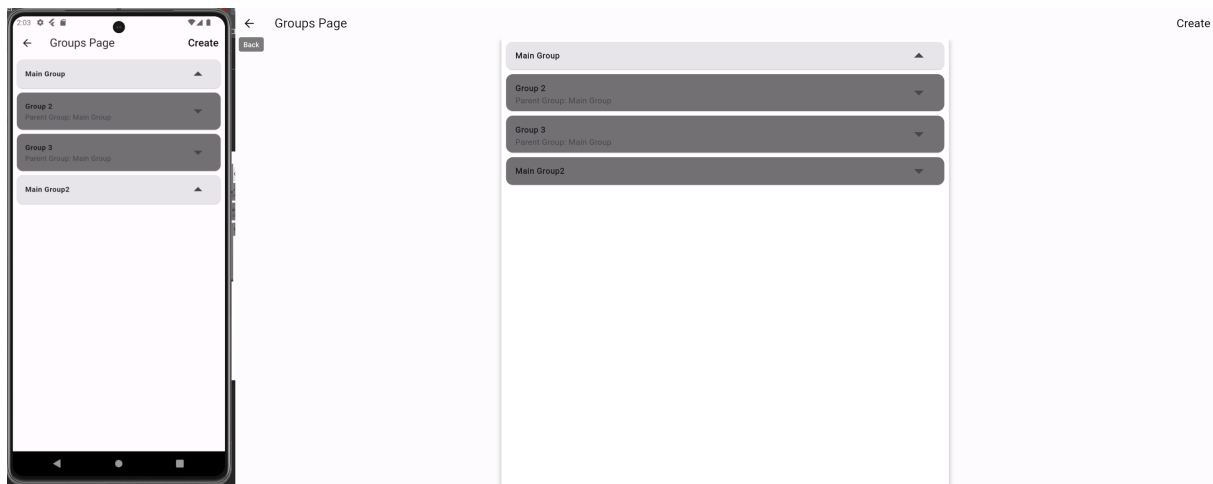


Figure 5.88: Show Child Groups

Admin can create group from the button on the upper right corner, and then he can enter the group name, description and the parent group or leave it null if he wants to create main group.

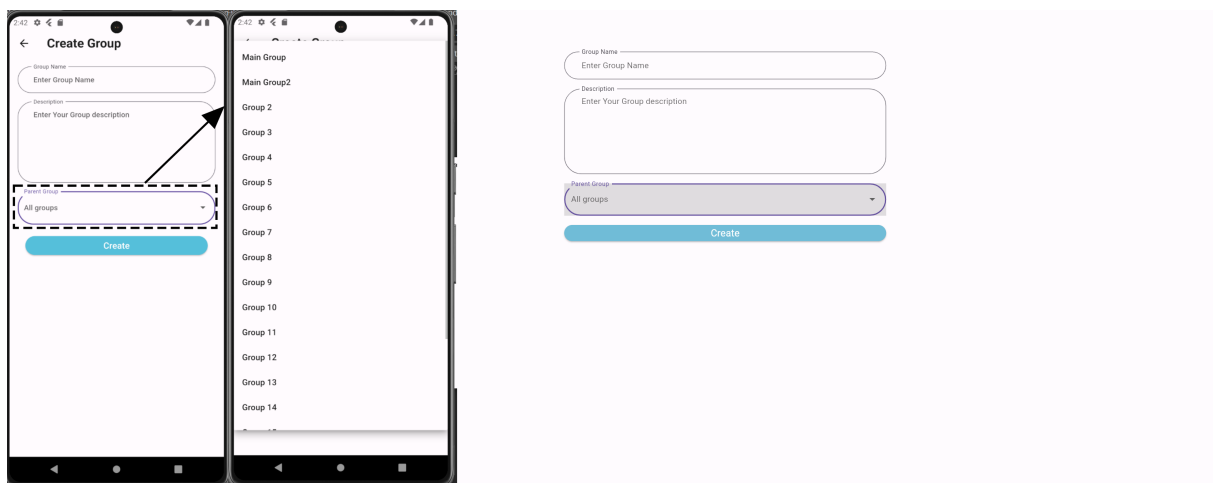


Figure 5.89: Show Child Groups

When user press on the group name he can enter the group chat:

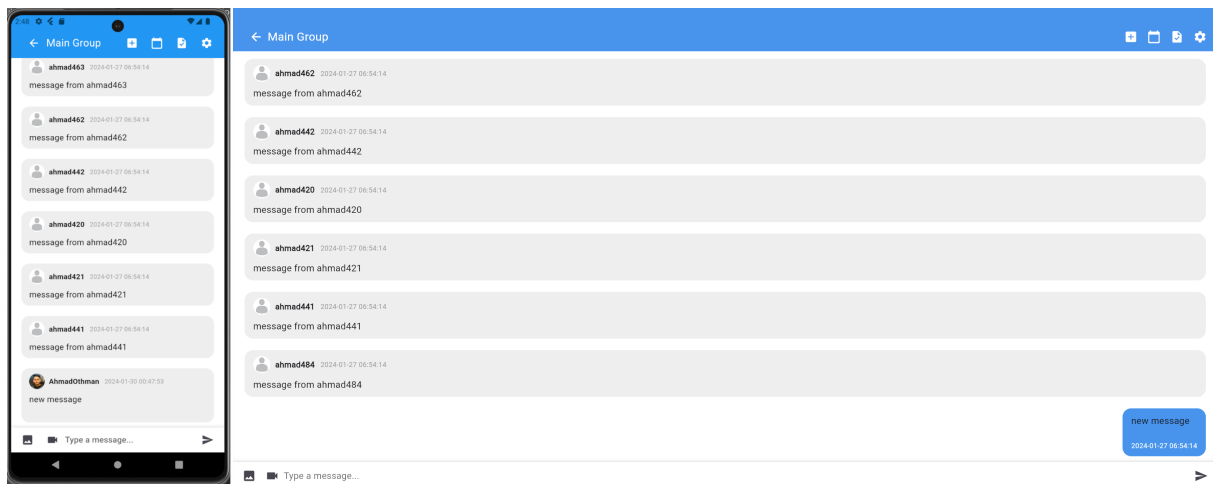


Figure 5.90: Admin Group Chat

## Group settings

Admin can reach group settings from the upper bar, where he can edit the main settings of the group.

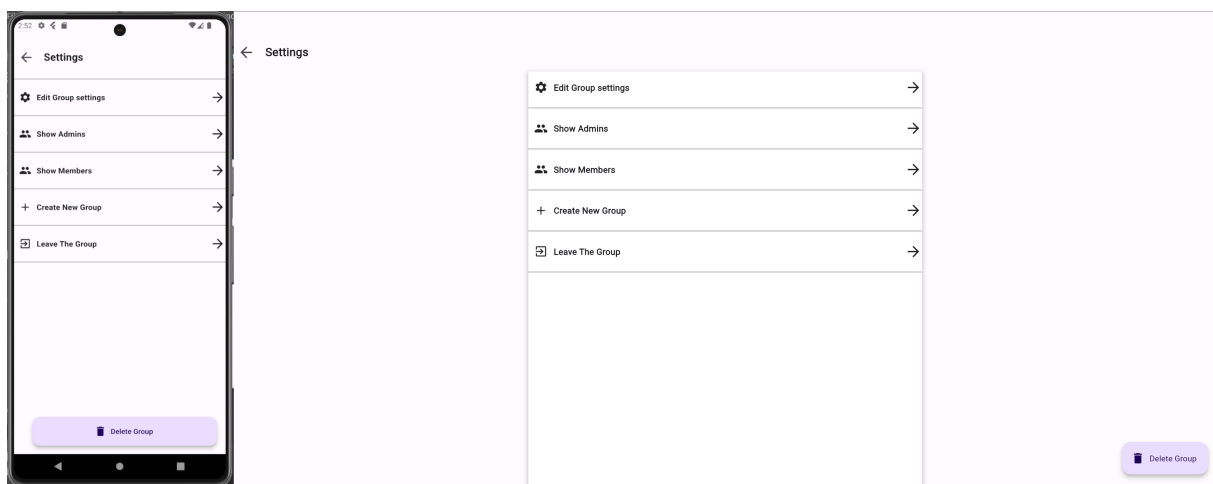


Figure 5.91: Group Settings

Group's Admins can:

- Delete group and all of its child groups:

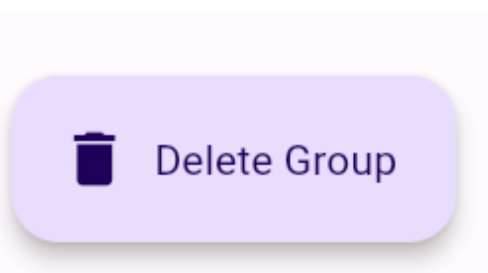


Figure 5.92: Delete Group

- Edit group info: group name, description, if members can send messages or not.

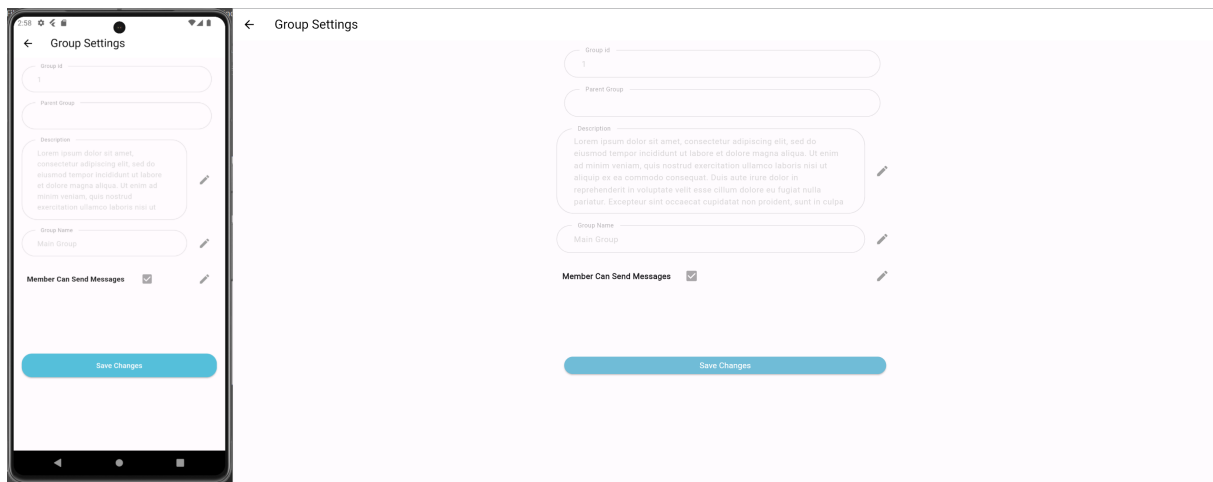


Figure 5.93: Edit Group Info

- Show admins, add new admin from the group members, delete (from the three dots next to admin name) group admins

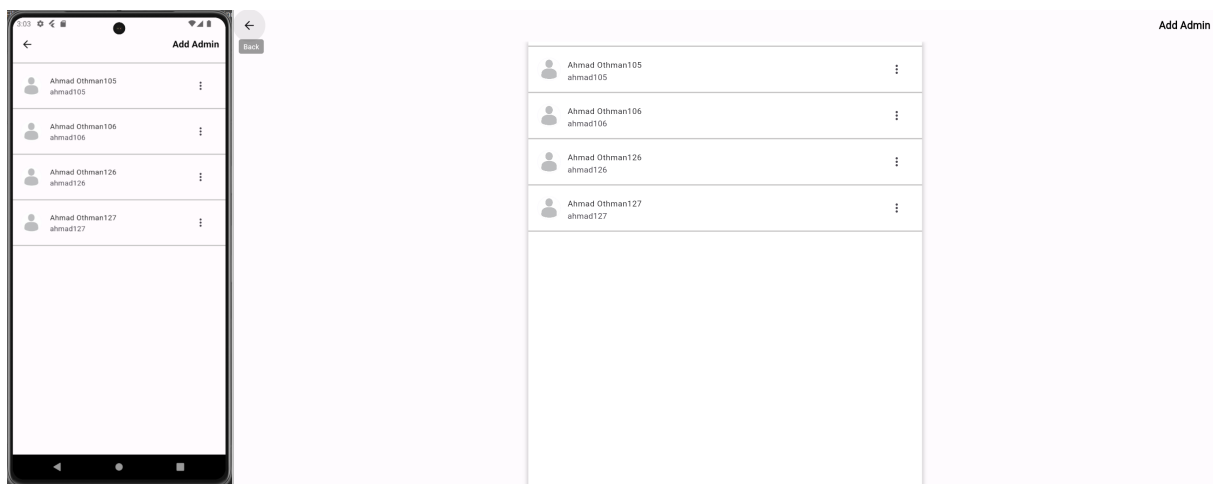


Figure 5.94: Show Group Admins

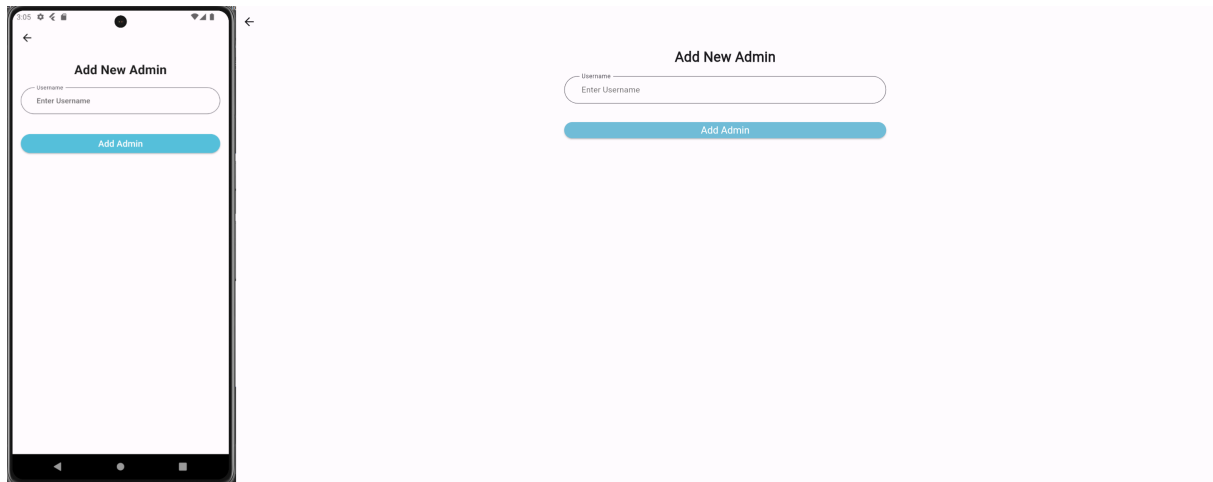


Figure 5.95: Add New Group Admin

- Show members, add new member from the page employees, Add an external member (not an employee of the page) , delete (from the three dots next to admin name) group members.

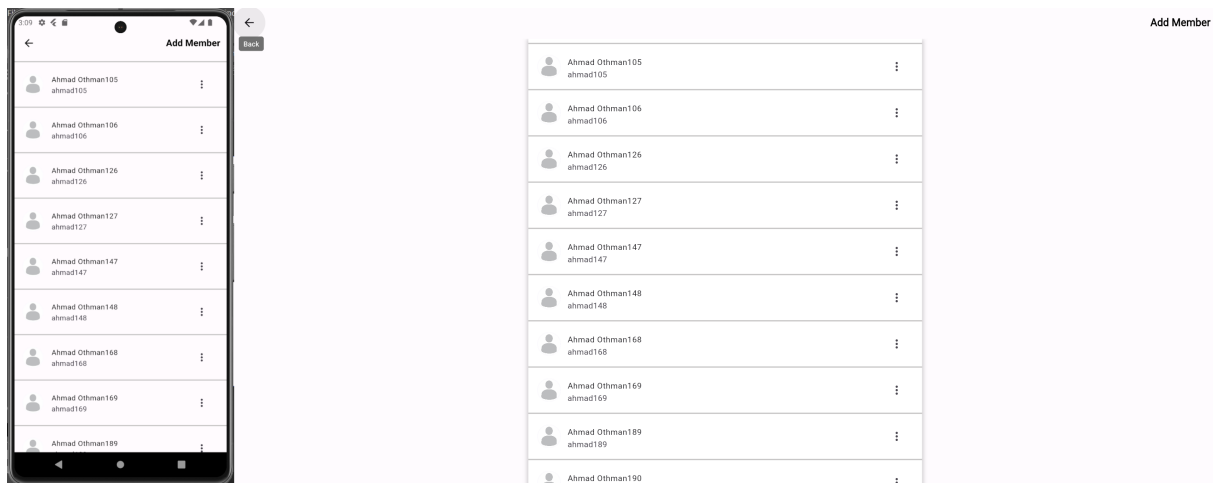


Figure 5.96: Show Group Members

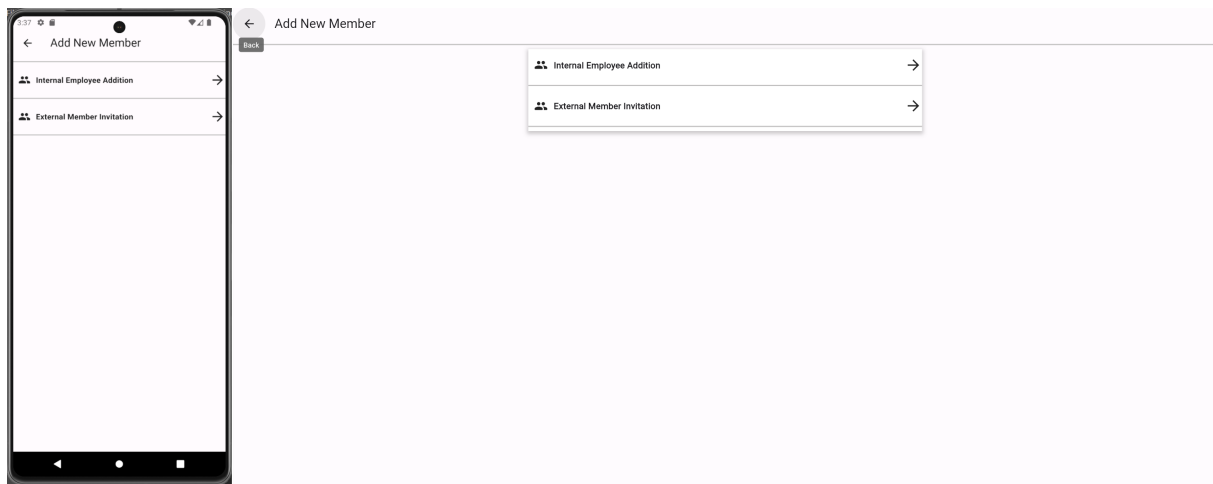


Figure 5.97: Add New Group member

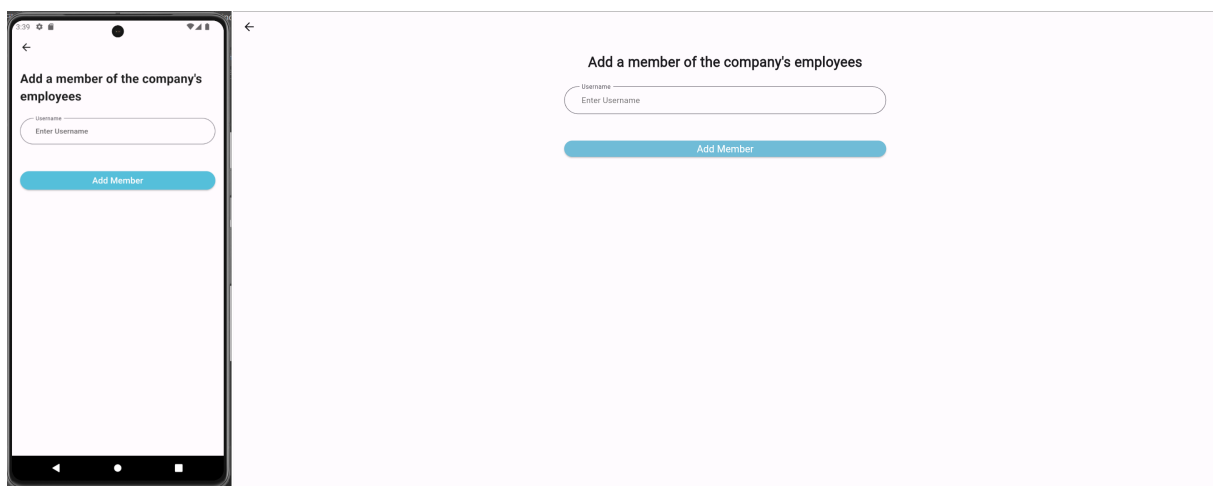


Figure 5.98: Add Member From Page Employees

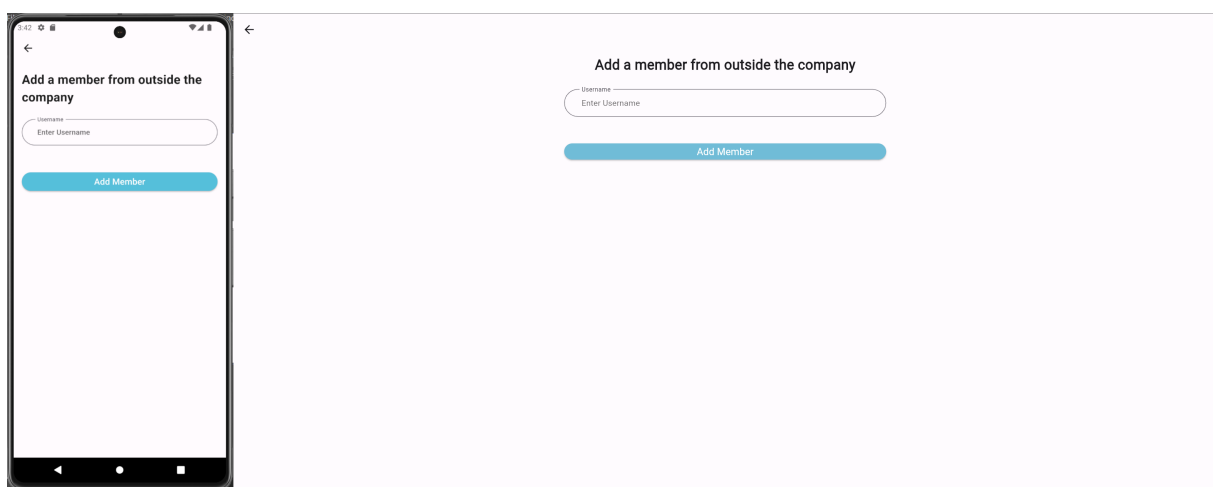


Figure 5.99: Add External Group Member

- Leave group (for direct group admin only).

**Why add a member to a group separated into two interfaces even though they offer the same functionality?** To add a layer of protection, the group admin can be aware that he is adding a member from outside the company's employees.

## Group tasks

The admin can view/add/delete tasks for members from this interface. This interface appears when admin click the second button from the upper right corner.

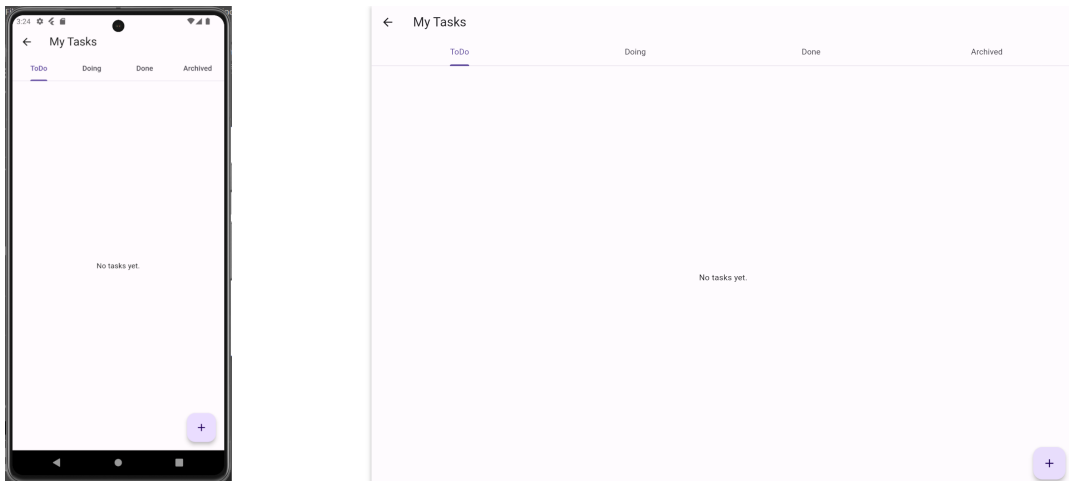


Figure 5.100: Group Tasks

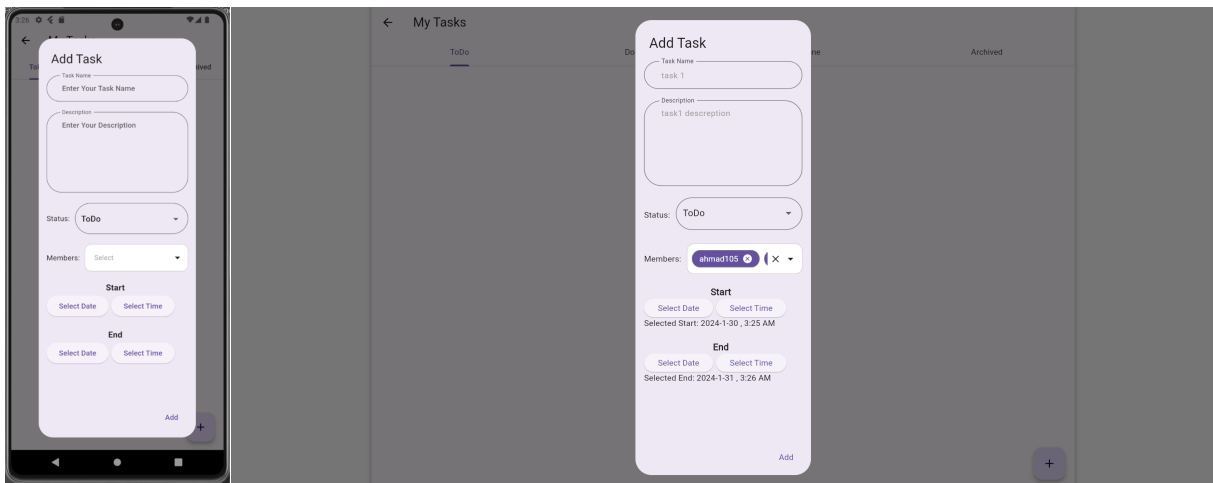


Figure 5.101: Add Group Task



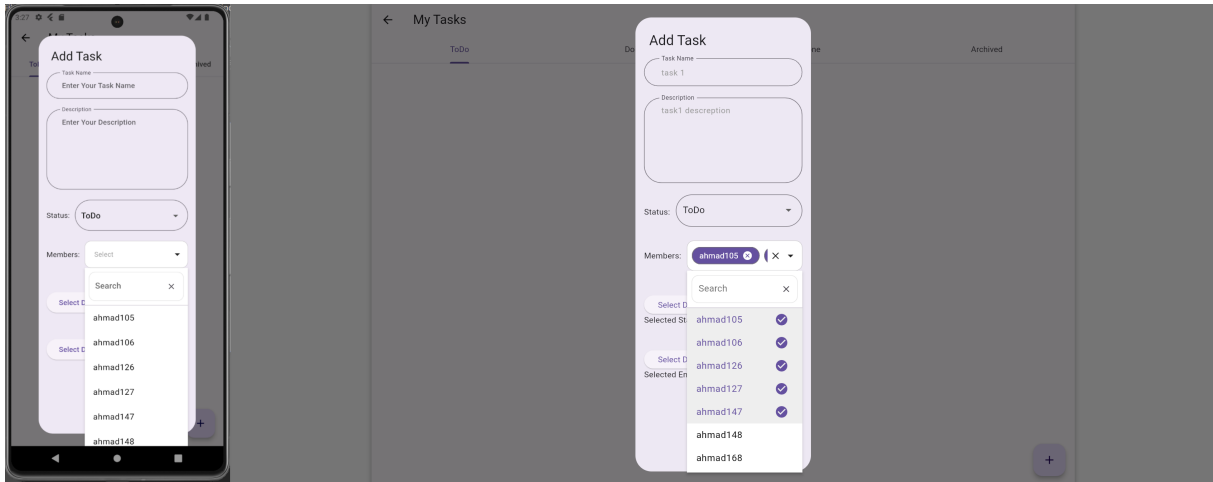


Figure 5.102: Select Task Members

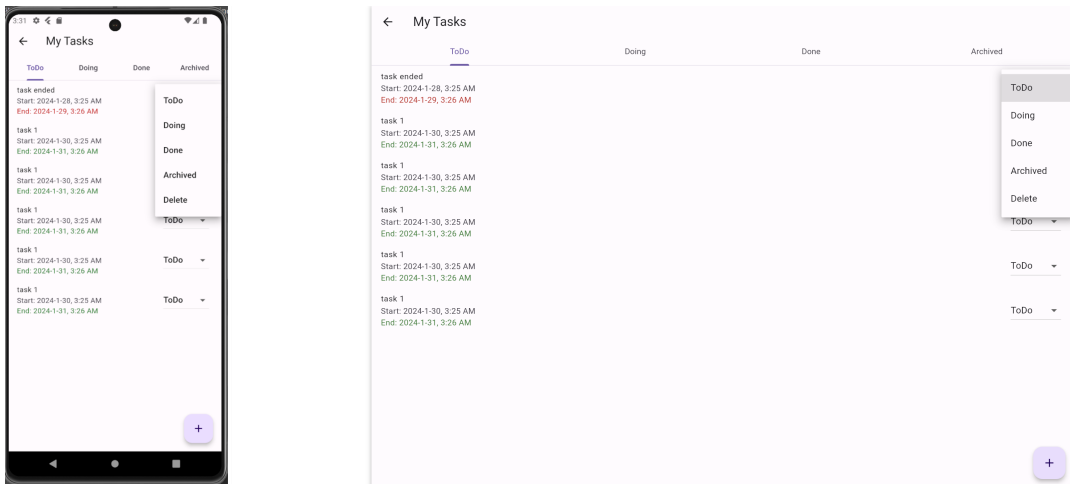


Figure 5.103: Change Group Task Status

## Group Calendar

The admin can view/add/delete events from this interface. This interface appears when admin click the third button from the upper right corner, members only can see the events.

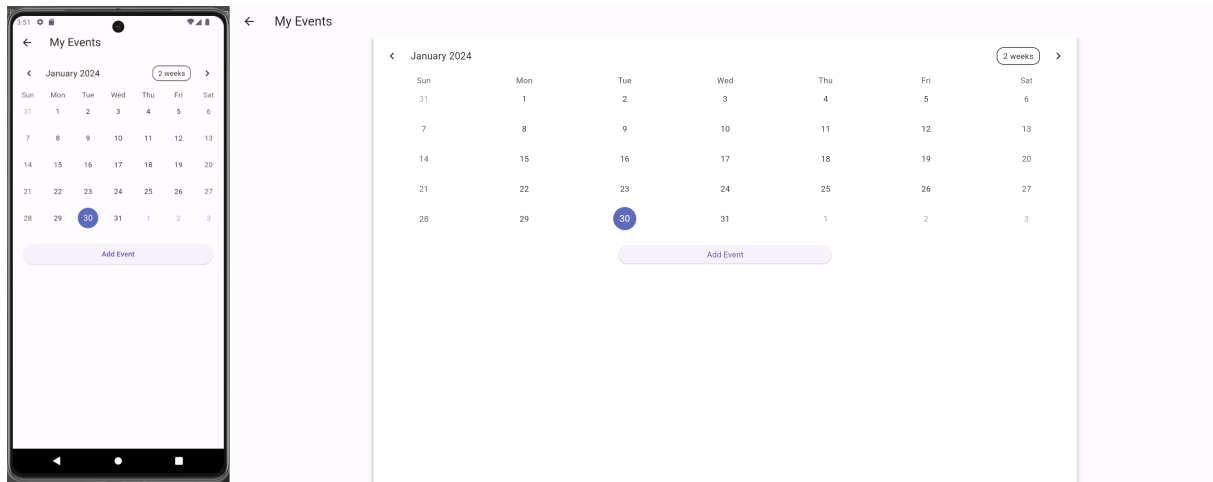


Figure 5.104: Group Calendar

## Group messages

The admin (and users if they are allowed) can send and receive text messages/images/videos through sockets. After sending the message, it will be broadcast to all group members.

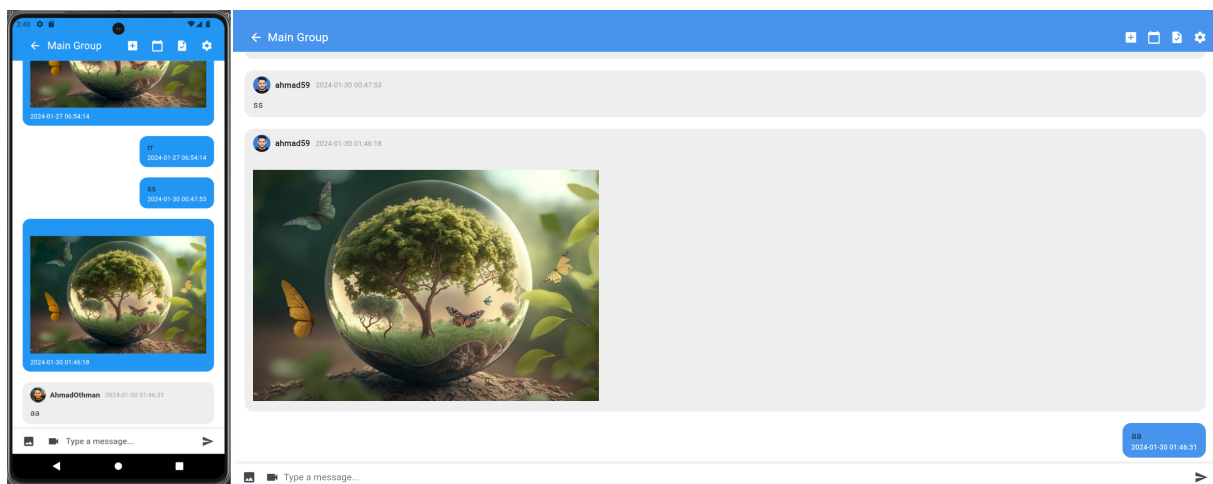


Figure 5.105: Group Messages

## Divide groups

Here are some suggestions for dividing groups for organizations/companies:

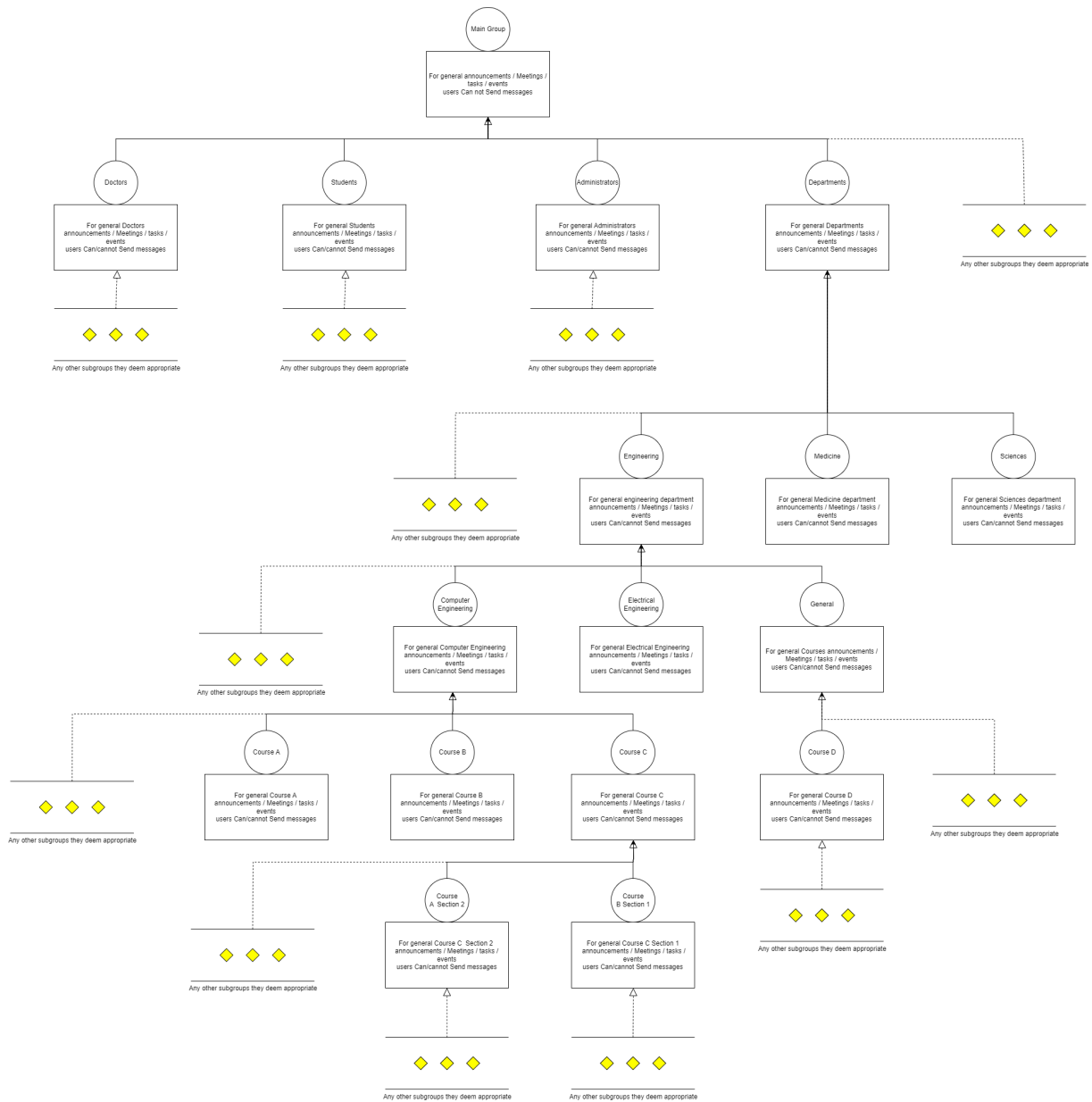


Figure 5.106: A proposal to divide groups for a university

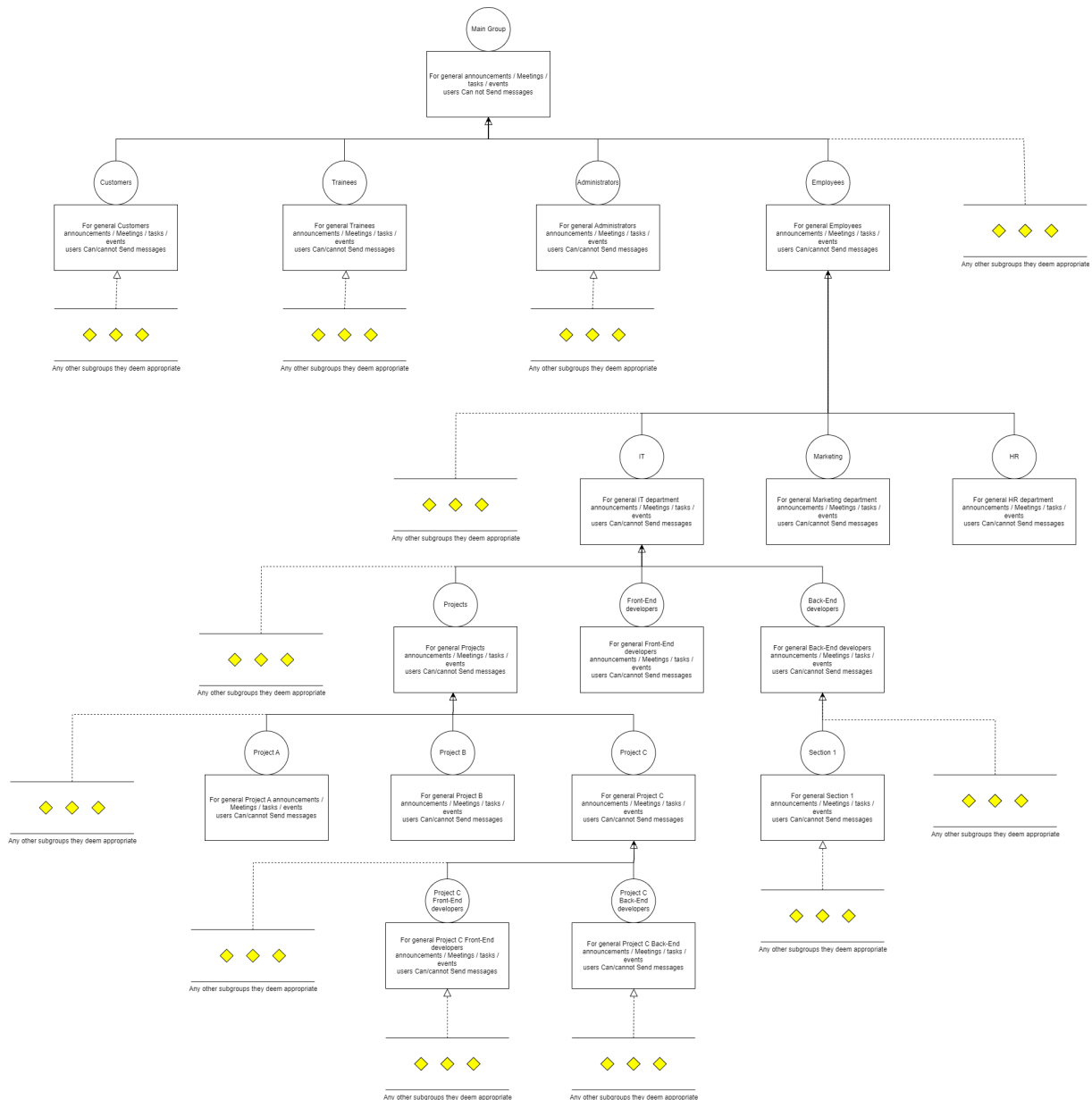


Figure 5.107: A proposal to divide groups for a Company

### 5.5.14 Group meeting

Group admin can create one meeting at a time for all group members. One group can contain a number of meetings at the same time, provided that each admin creates one meeting at a time. A meeting ID is created consisting of the username of the admin and the group ID. then it will be broadcast via sockets to all group members in a message. The admin can create a meeting and the user can join the meeting through the fourth button in the upper right corner.

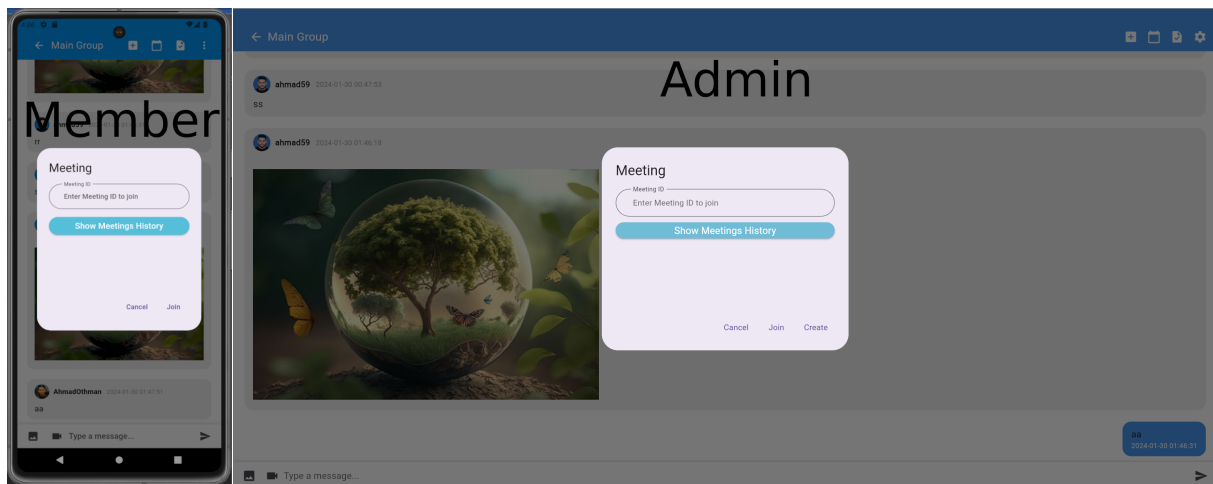


Figure 5.108: Group Meeting

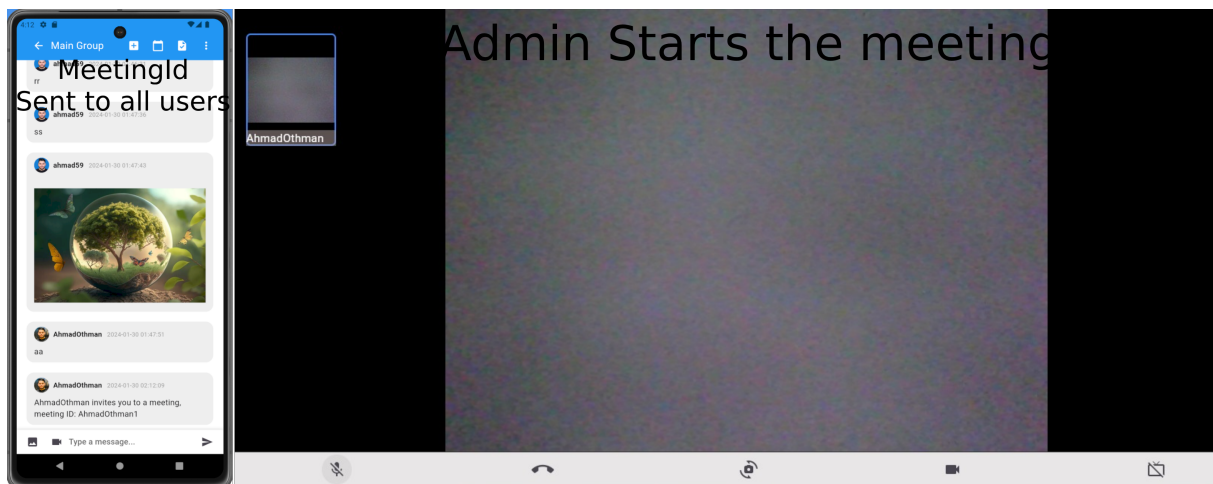


Figure 5.109: Admin Creates Meeting

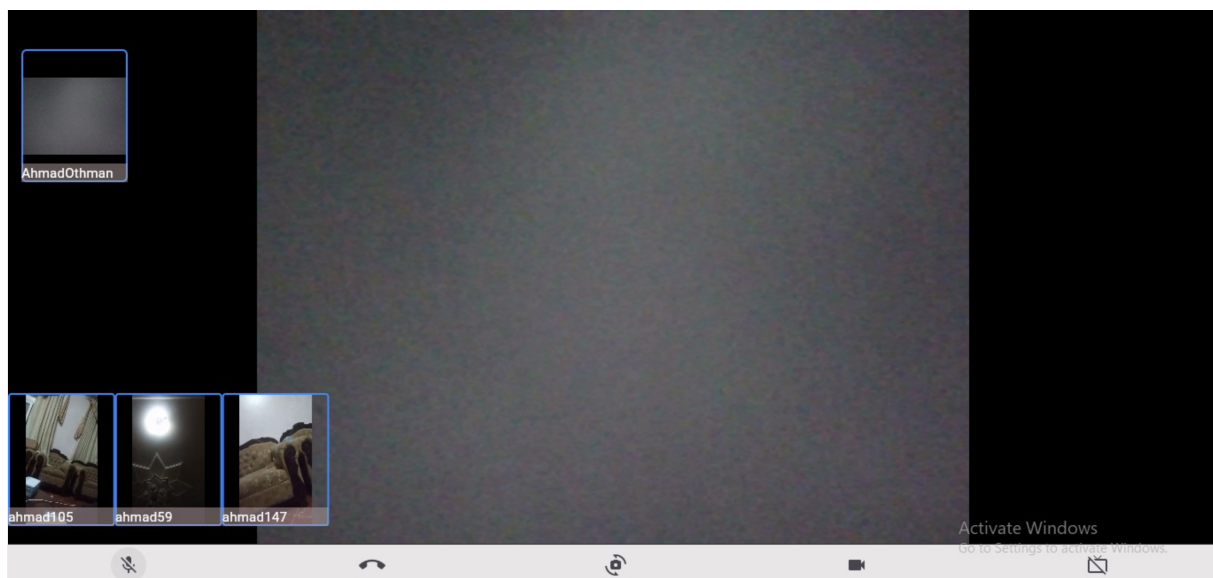


Figure 5.110: Members Join Meeting

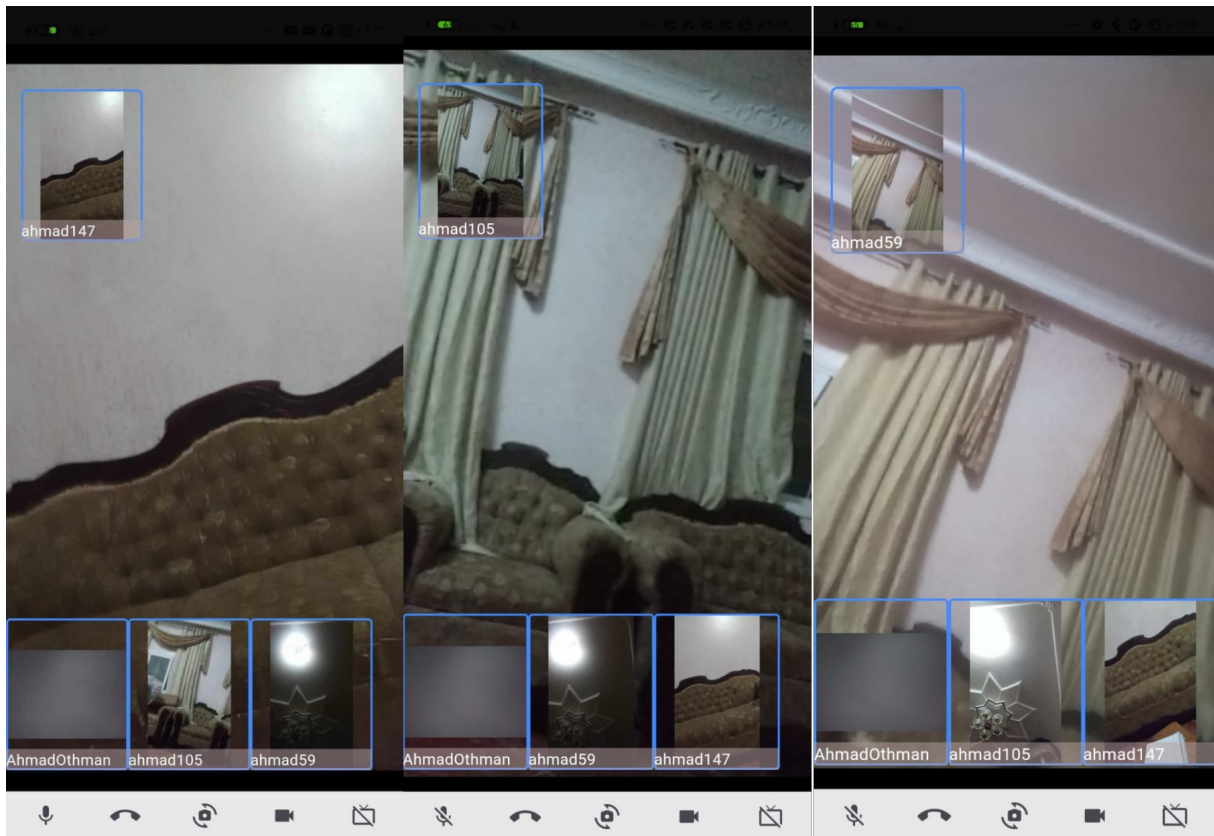


Figure 5.111: Meeting Members Seeing Each Other

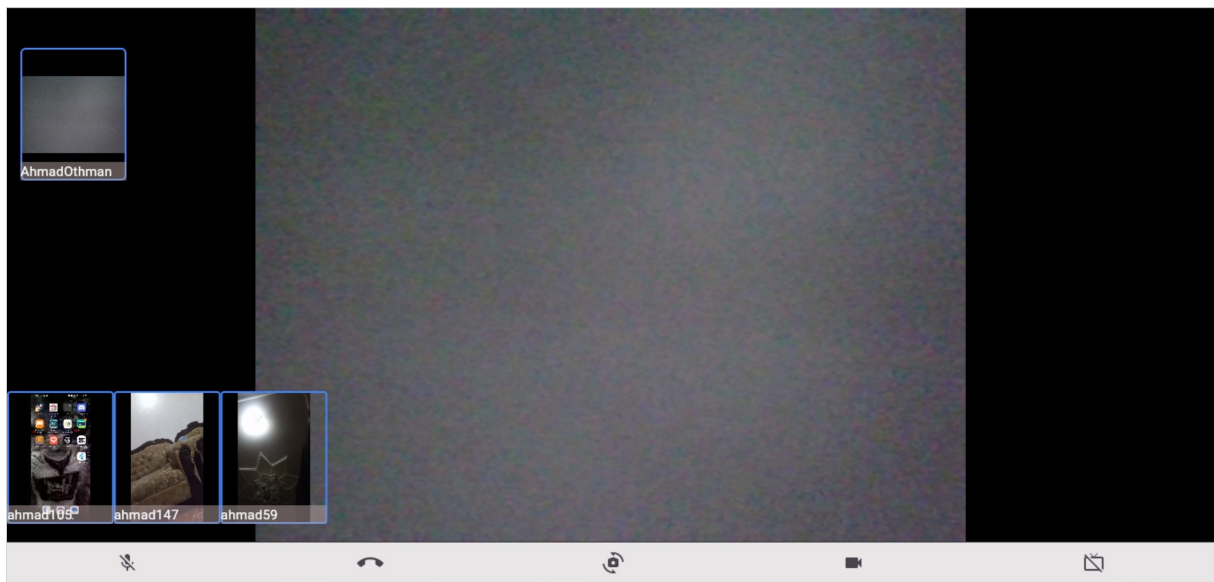


Figure 5.112: One Member Share Screen From Mobile

When user press on the edge of other user screen, he can make it as the main screen in the middle.

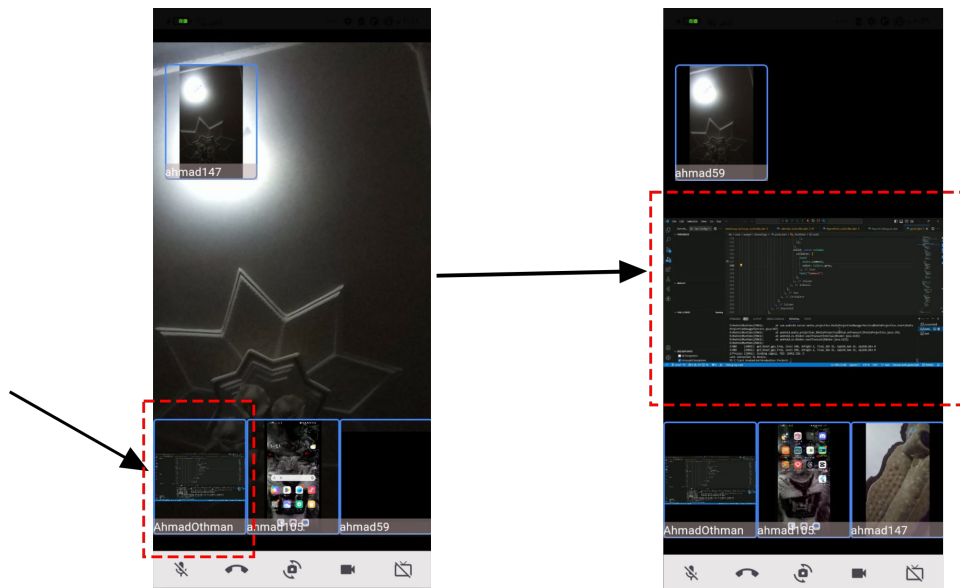


Figure 5.113: Two Members Share Screen From Mobile And Web

Meetings History	Meetings History
A meeting was held with id: AhmadOthman1 At: 2024-01-30 02:11:08 Period: 21m 57s	A meeting was held with id: AhmadOthman1 At: 2024-01-30 02:12:09 Period: 3s
A meeting was held with id: AhmadOthman1 At: 2024-01-30 02:11:19 Period: 21m 46s	A meeting was held with id: AhmadOthman1 At: 2024-01-30 02:11:08 Period: 1m 1s
A meeting was held with id: AhmadOthman1 At: 2024-01-30 02:12:01 Period: 21m 4s	A meeting was held with id: AhmadOthman1 At: 2024-01-30 02:11:19 Period: 50s
A meeting was held with id: AhmadOthman1 At: 2024-01-30 02:12:09 Period: 20m 56s	A meeting was held with id: AhmadOthman1 At: 2024-01-30 02:12:01 Period: 8s
A meeting was held with id: AhmadOthman1 At: 2024-01-30 02:33:05 Period: Meeting is taking place now	

Figure 5.114: Meeting History

## Meeting process

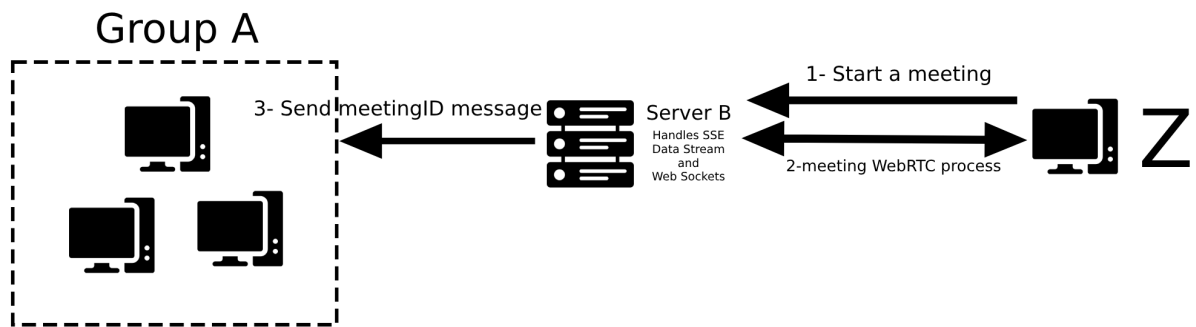


Figure 5.115: Meeting Process - Admin Starts A Meeting

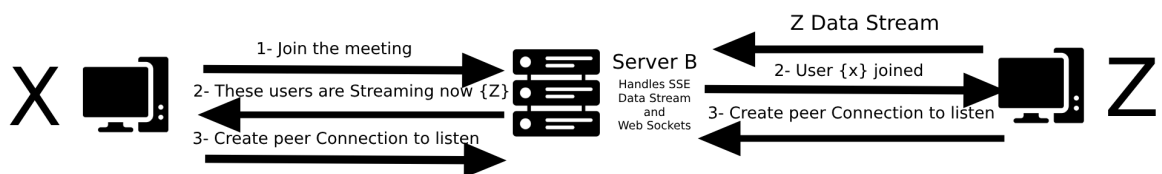


Figure 5.116: Meeting Process - User X Joined



Figure 5.117: Meeting Process - Two User Listening To Each Other



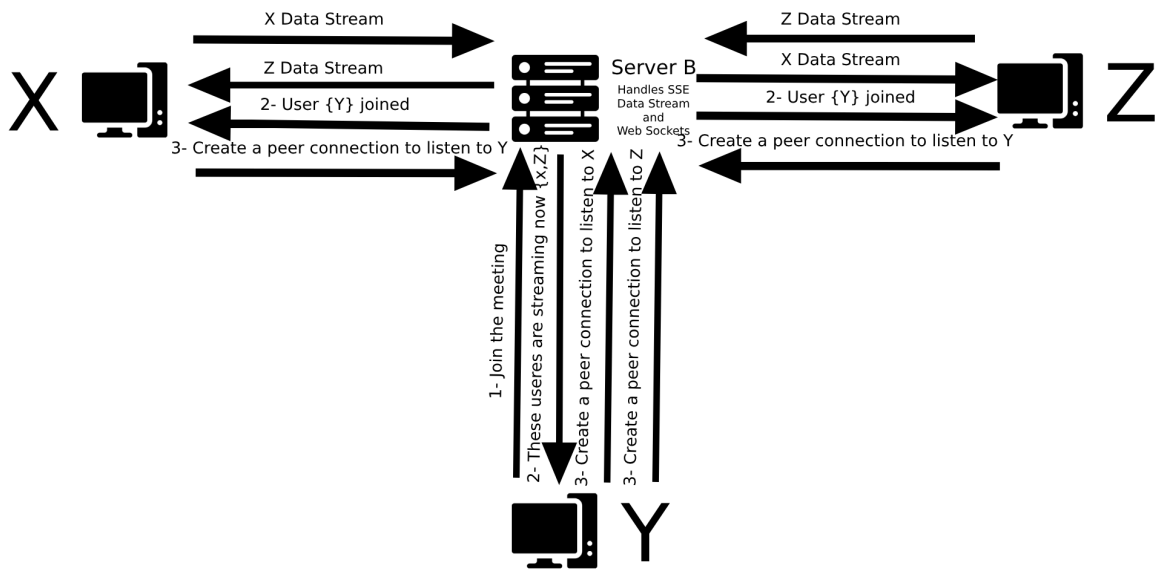


Figure 5.118: Meeting Process - User Y Joined

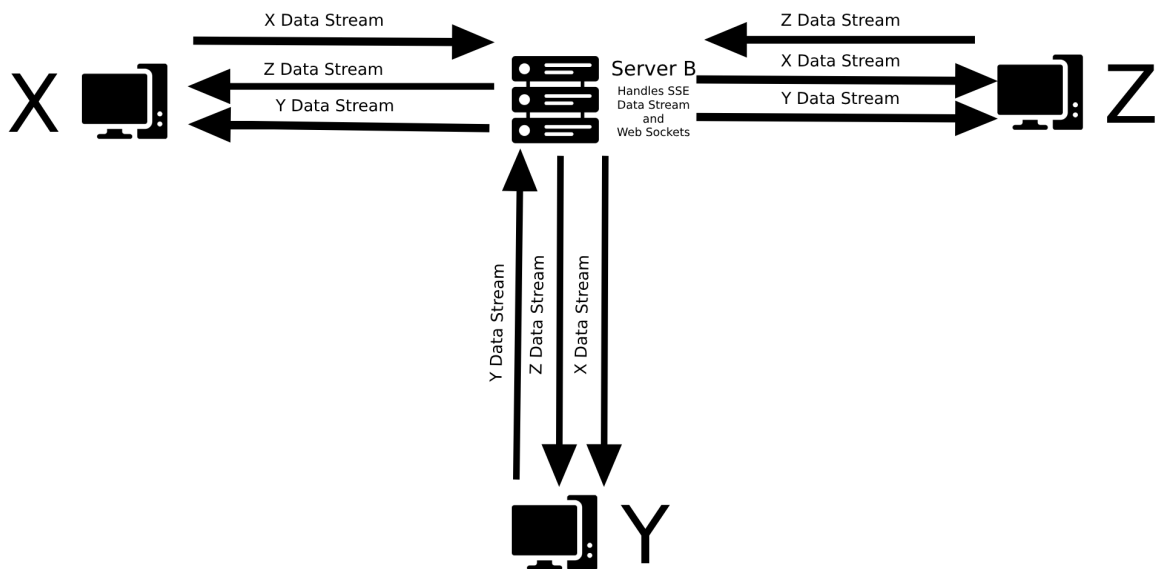


Figure 5.119: Meeting Process - All User Listening To Each Other

### 5.5.15 Networks

User can reach his networks from the bottom bar of the home screen (10 items will be loaded on each page).

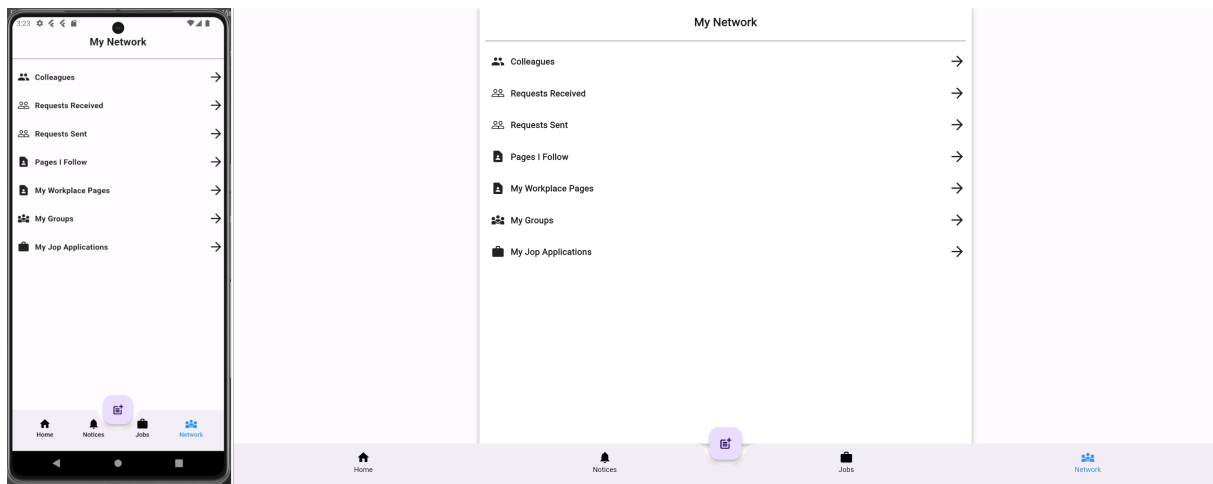


Figure 5.120: User Networks

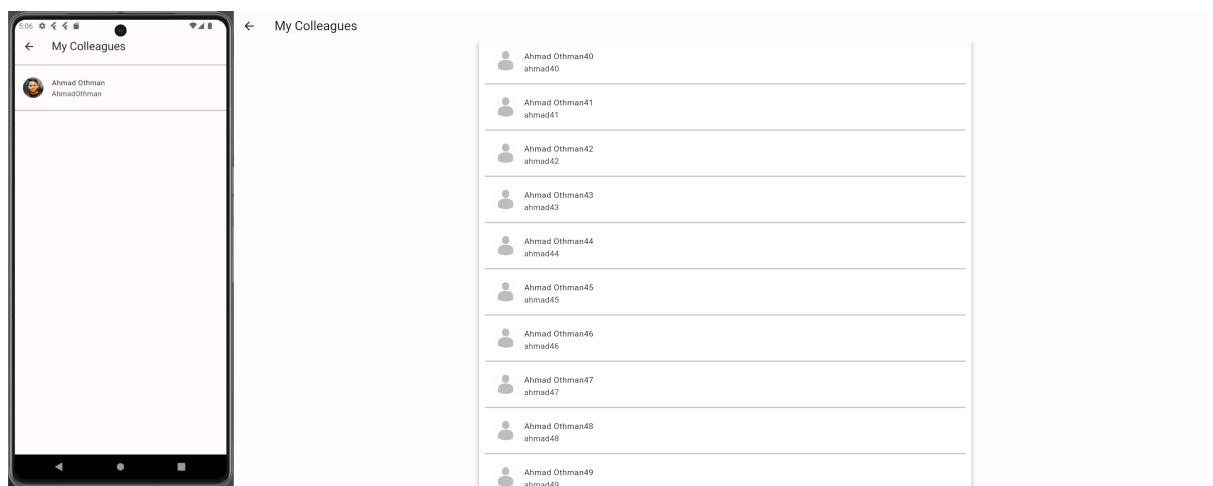


Figure 5.121: User Connections

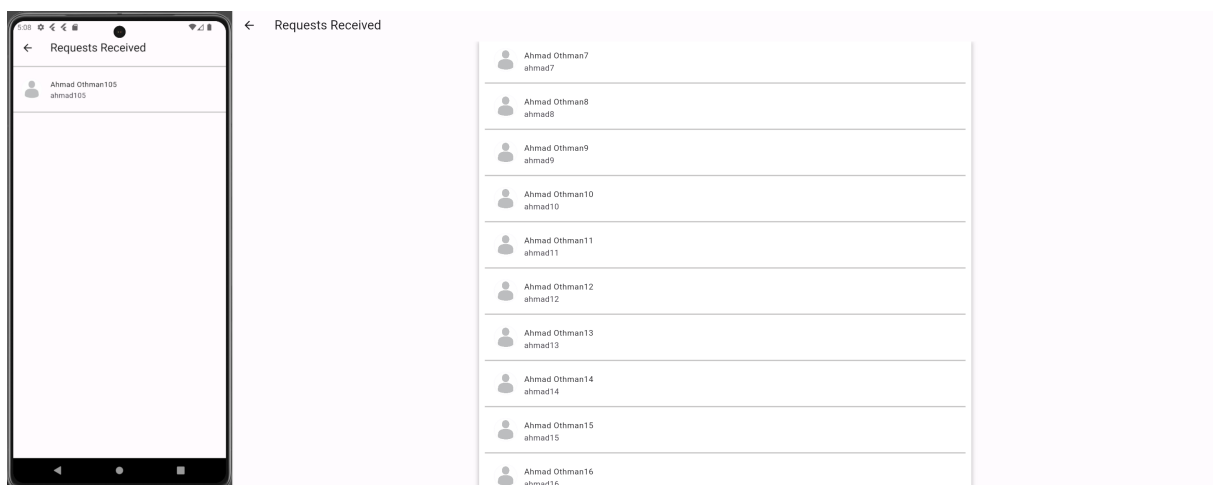


Figure 5.122: Requests Received

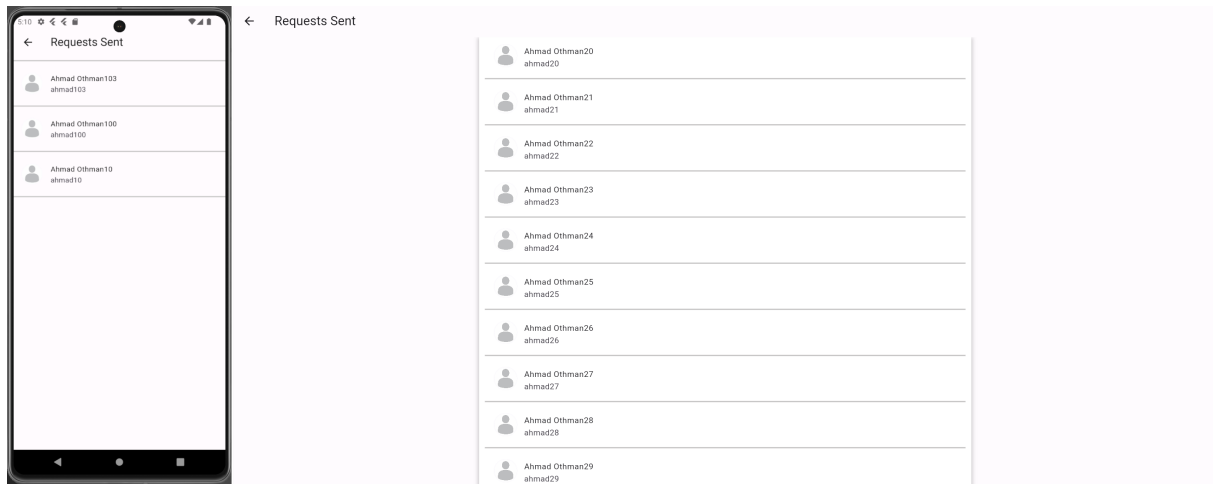


Figure 5.123: Requests Sent

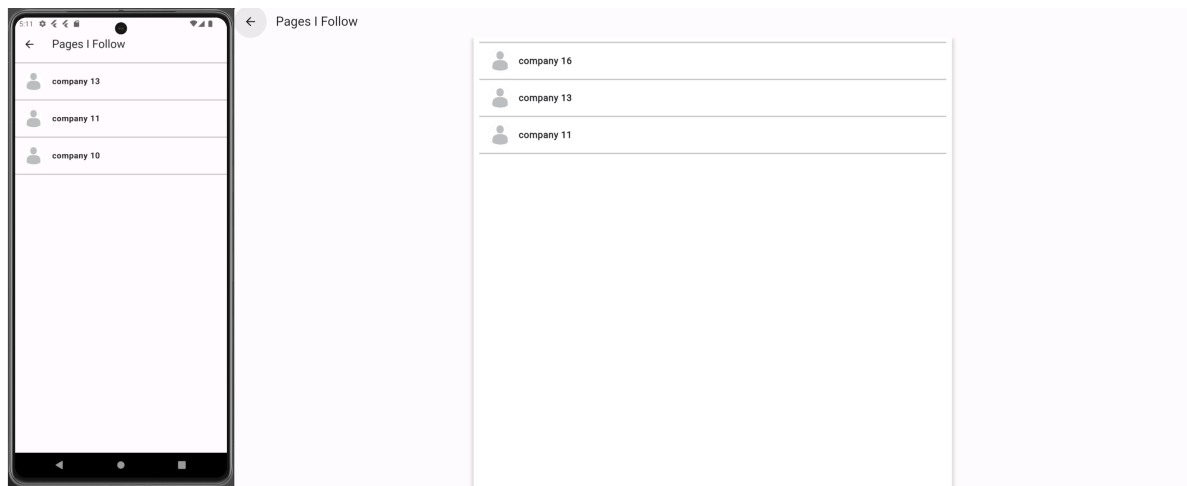


Figure 5.124: Pages User Follows

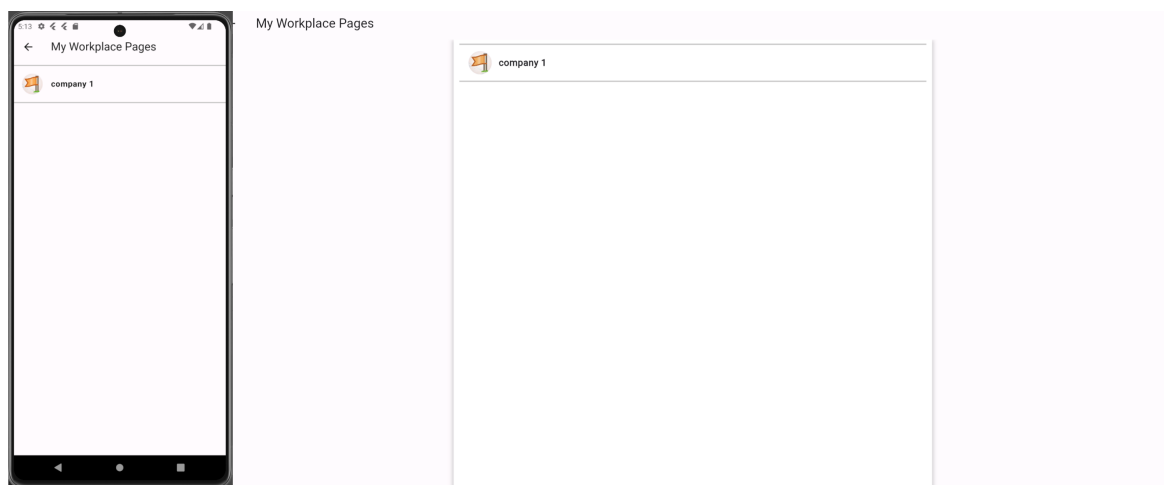


Figure 5.125: Pages User Employee In

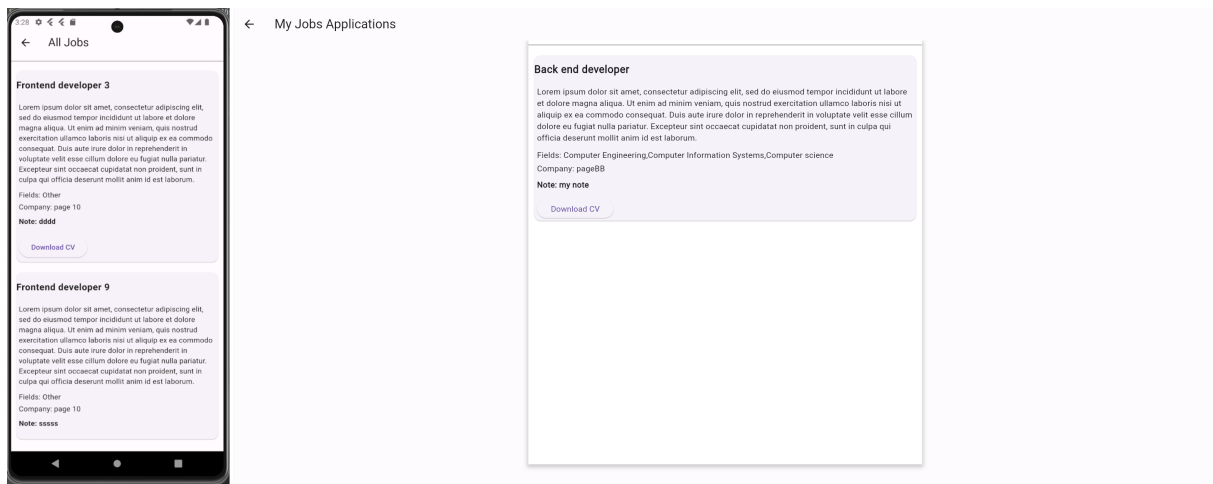


Figure 5.126: User Job Applications

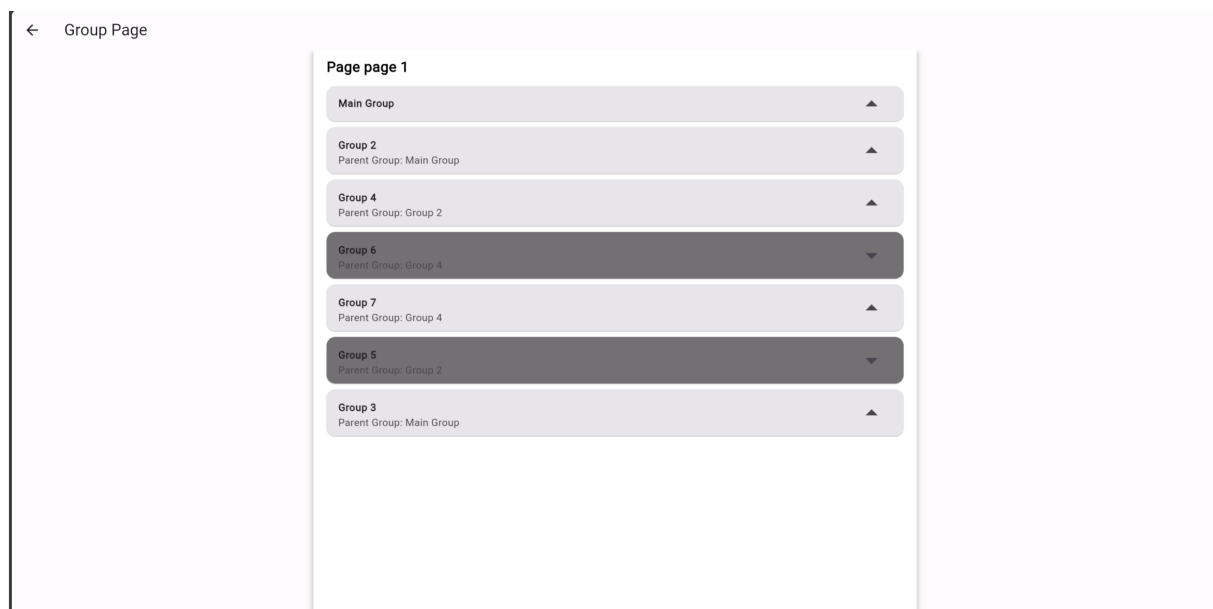


Figure 5.127: Groups User Is Admin Or Member In

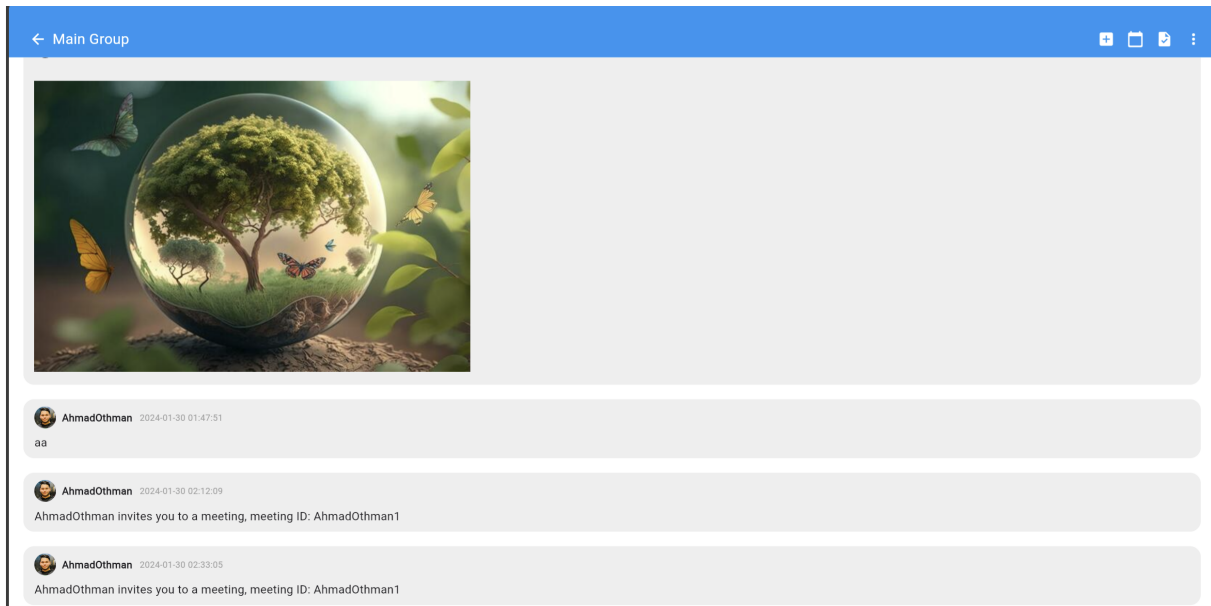


Figure 5.128: If Member Cannot send messages From Group Settings

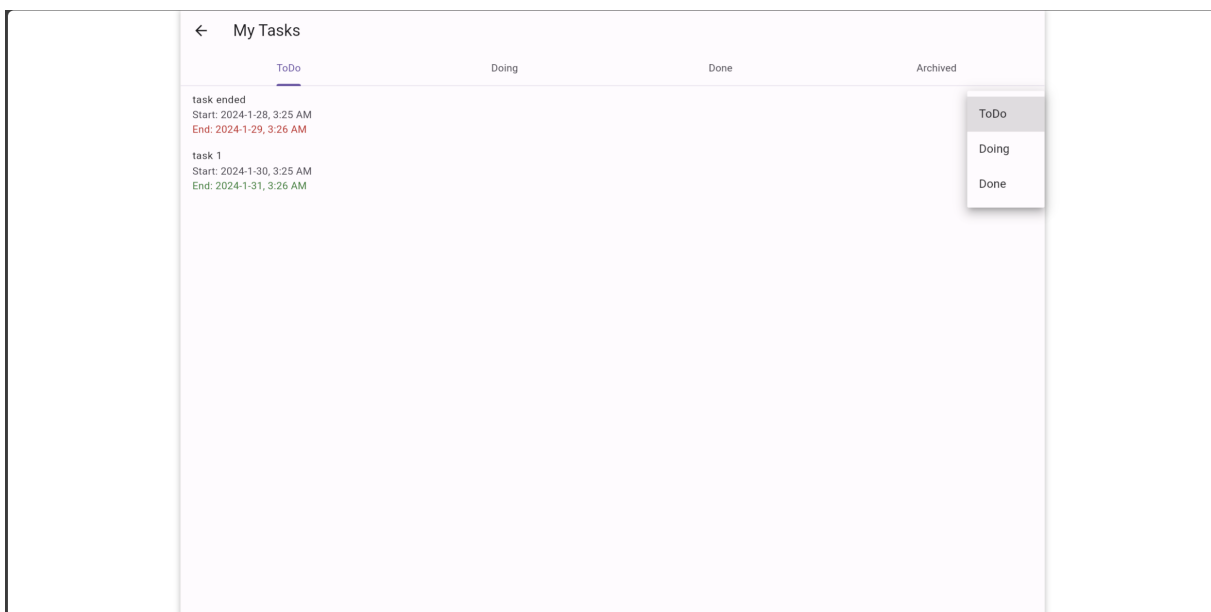


Figure 5.129: Group Member Tasks And Change Status

### 5.5.16 Jobs

In this page, user can see all the jobs for the pages he follow (10 jobs will be loaded every time user scroll to the bottom of the screen).

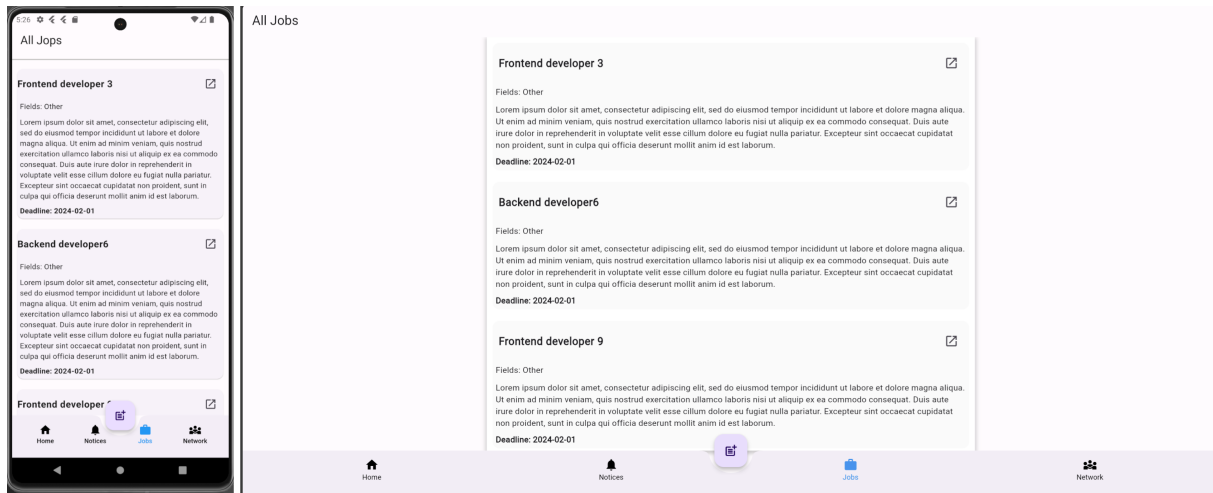


Figure 5.130: User Jobs

If the application for the job has not ended, and the user is not an admin or employee on the page or has previously applied for the same job, the user can attach a note or CV to his application to apply for the job.

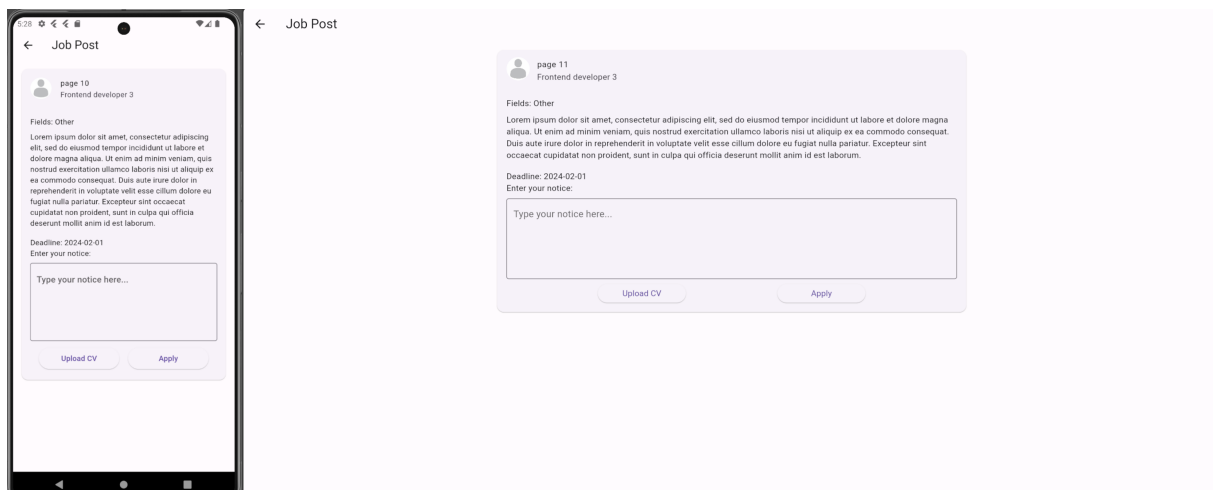


Figure 5.131: See Job Details And Apply



Figure 5.132: The time to apply for job has ended

### 5.5.17 Search

In this page, user can Search for other users, pages, jobs, and can choose one or more filtration option at the same time.

user can search for a username by writing "@" before the name he want to search for, and the same for specific page id.



Figure 5.133: Search Filtration

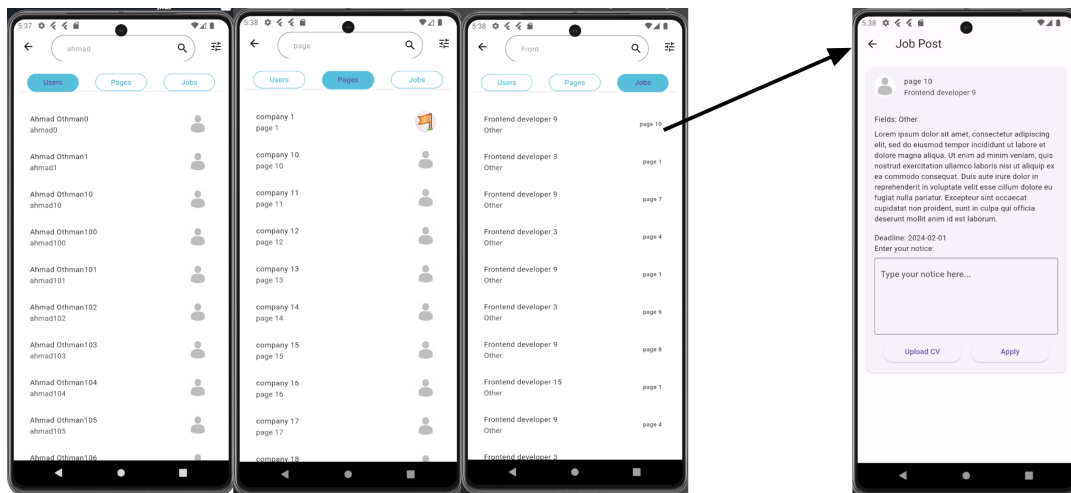


Figure 5.134: Search Results

### 5.5.18 Other Pages

When user press on page he is not admin in, he can follow the page, see posts, calendar, jobs, start a conversation, report the page, add like/comment, report post, report comment, etc. Most of these functions have been explained previously.

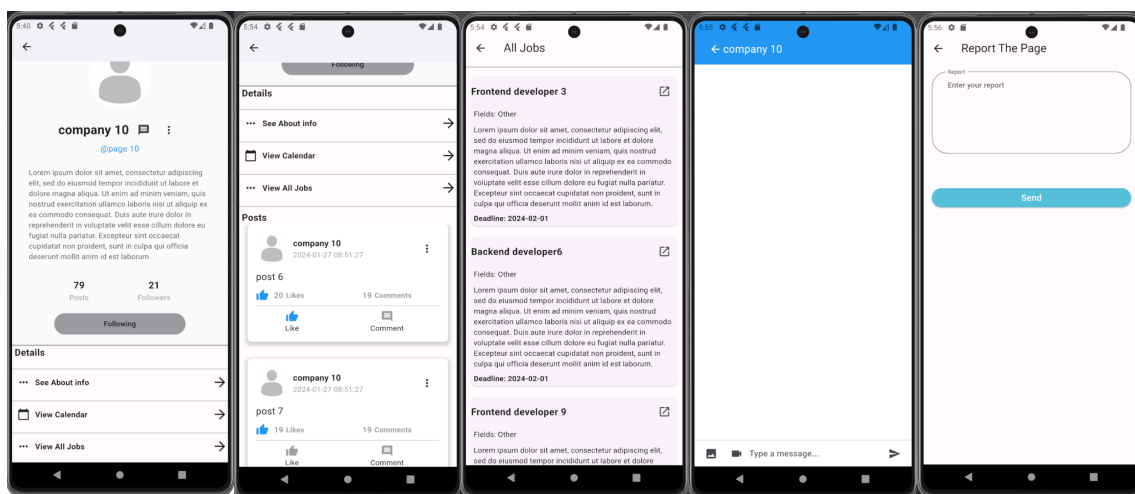


Figure 5.135: Others Page

### 5.5.19 Notifications

The user can receive notifications when he opens the application directly and automatically, as Function subscribes to the Events server, so the server can send notifications when they are available. The user receives notifications when:

- Gets a new like/comment on his post.
- Receives a connection request.
- A user approves his connection request.
- A user/page sends him a message.



- A user is calling him (with a ringing sound playing for 25 seconds).
- A page he follows publishes a job such that at least one of the job's fields matches one of the user's fields.

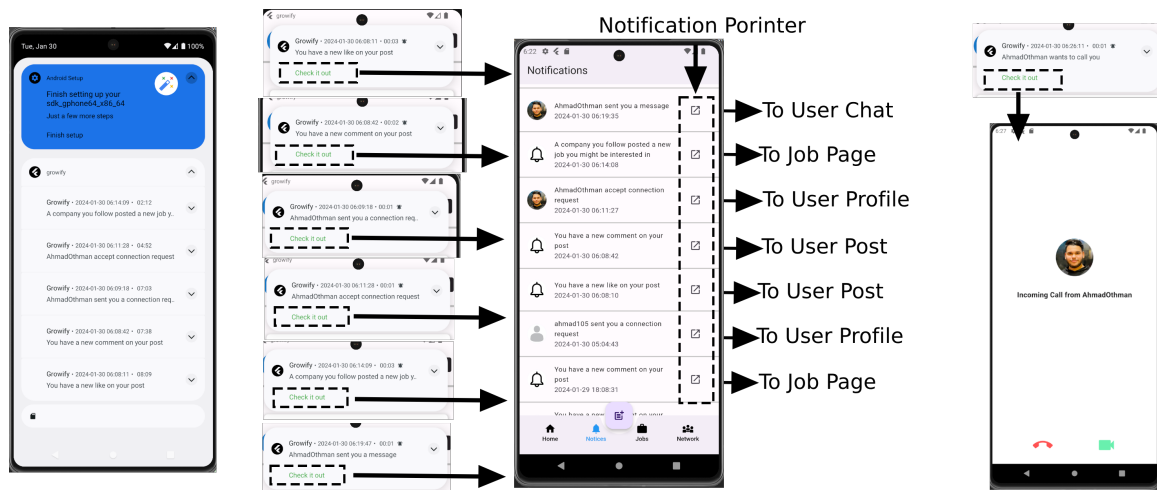


Figure 5.136: User Notification

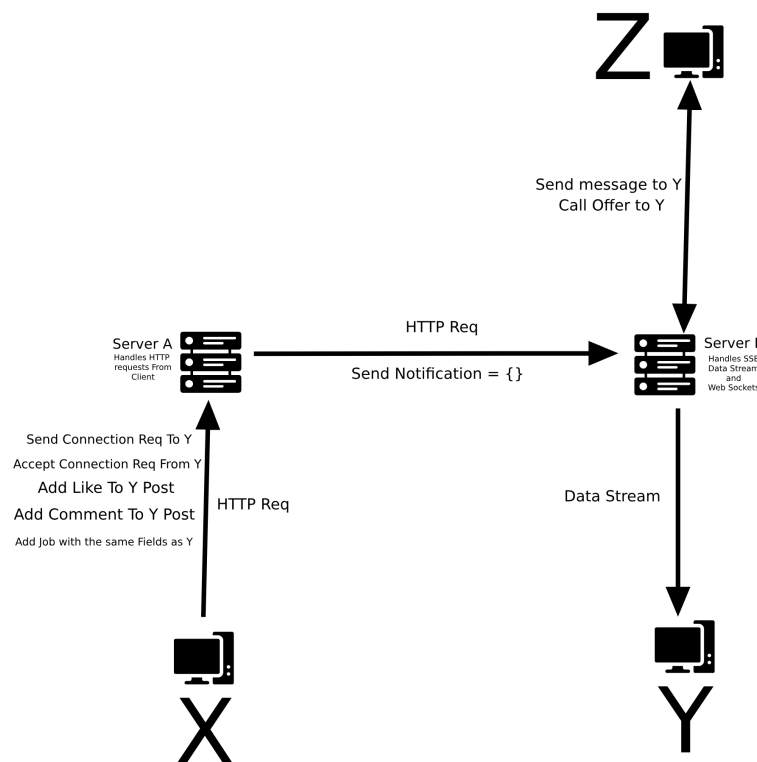


Figure 5.137: Notification process

## 5.5.20 Admin panel

The user can log in to the admin panel if his type in the user table is admin:

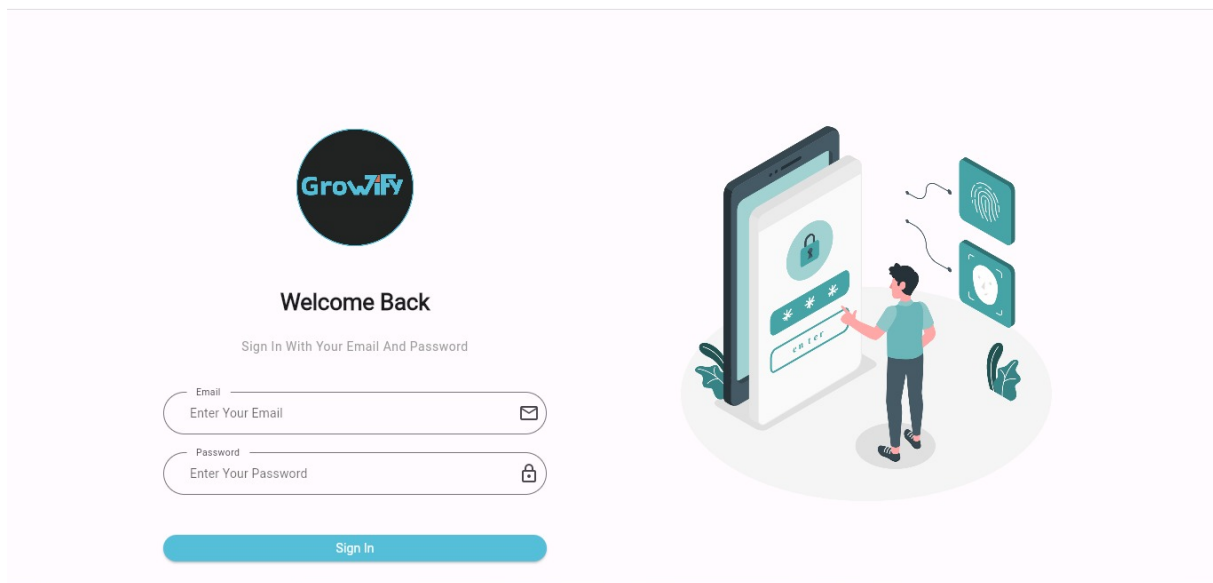


Figure 5.138: Admin Login Page

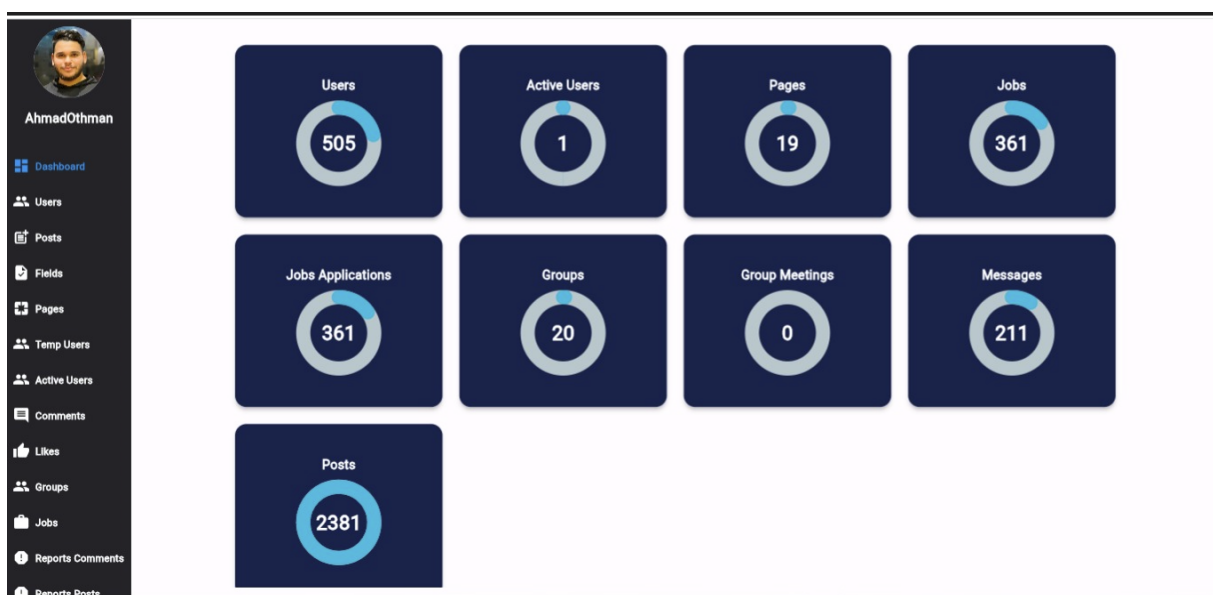


Figure 5.139: Dashboard table

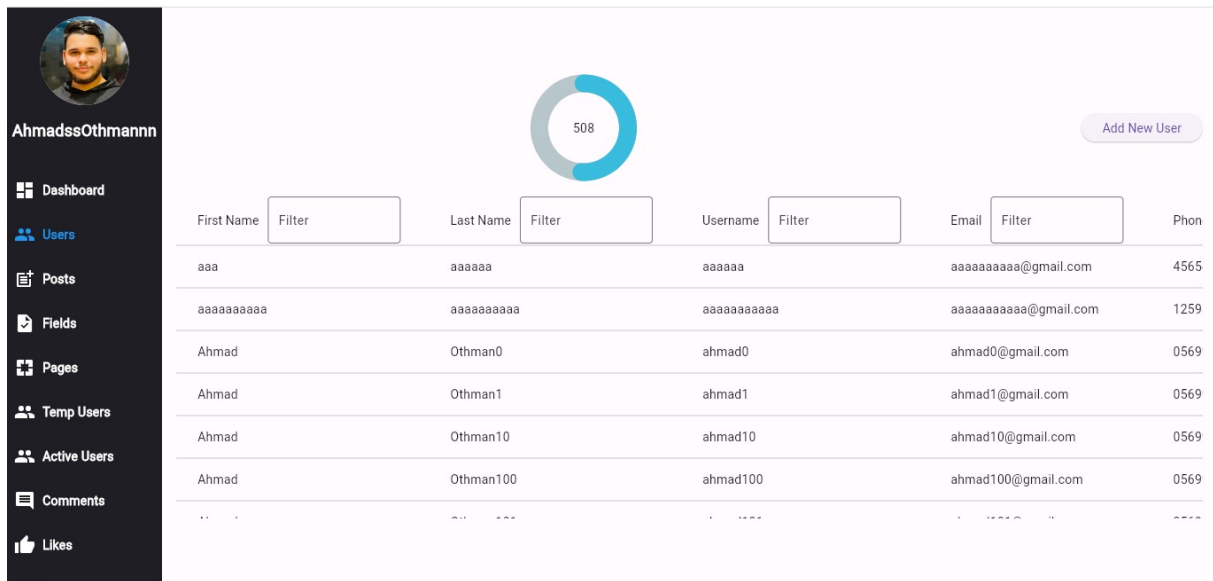


Figure 5.140: Users table

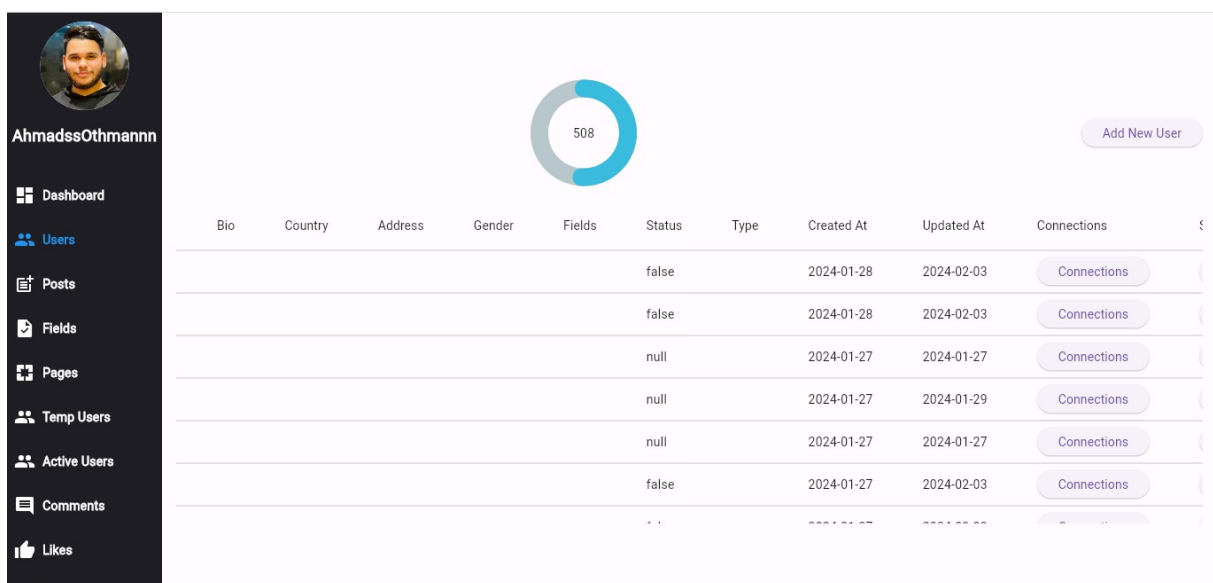


Figure 5.141: Users table

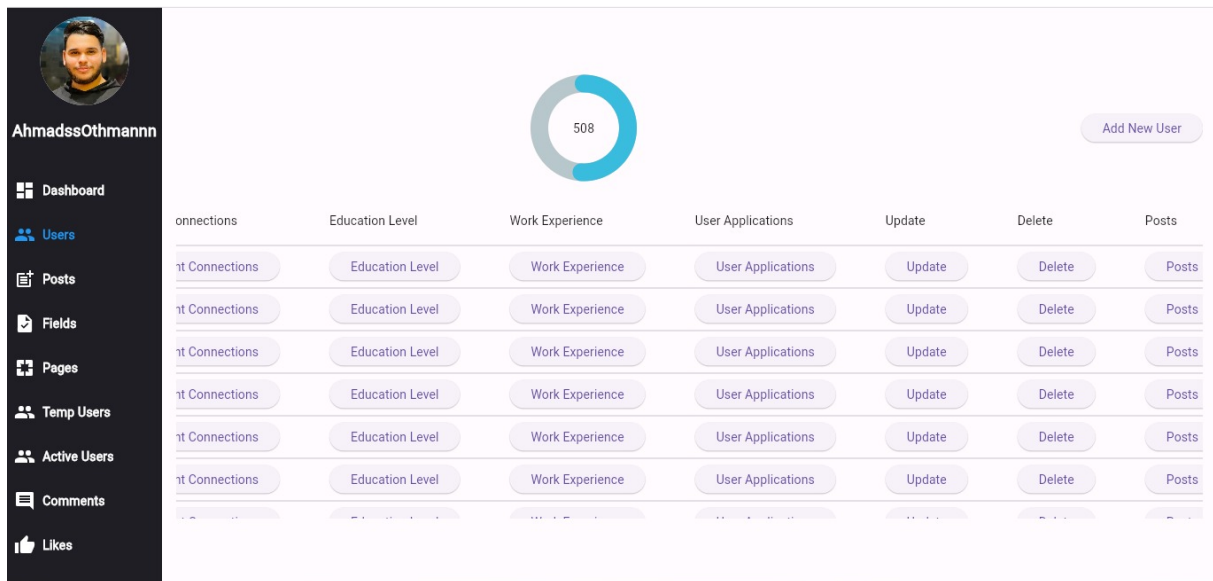


Figure 5.142: Users table

The 'Add New User' form includes the following fields:

- First name: Enter Your first name
- Lastname: Enter Your last name
- Username: Enter Your Username
- Email: Enter Your Email
- Phone: Enter Your Phone
- Password: Enter Your Password
- DateOfBirth: (with a calendar icon)

To the right of the form is an illustration of a woman holding a smartphone with a notification bell icon, surrounded by icons for a gift, a heart, and a megaphone.

Figure 5.143: Add New User

←

Profile Settings

Firstname

Ahmadss

Lastname

Othmann

Address

Palestine - Jenin

Your Fields

Computer Engineering

Other

Compu

Your Country

Palestine

Your Gender

Male

DateOfBirth

2001-05-18

Phone

0569929734

Figure 5.144: Update User

←

User Connections

21

receiverUsername	Filter	senderUsername	Filter	date	Filter	created At	Filter
ahmad40		AhmadOthman		2024-01-27		2024-01-27	
ahmad41		AhmadOthman		2024-01-27		2024-01-27	
ahmad42		AhmadOthman		2024-01-27		2024-01-27	
ahmad43		AhmadOthman		2024-01-27		2024-01-27	
ahmad44		AhmadOthman		2024-01-27		2024-01-27	
ahmad45		AhmadOthman		2024-01-27		2024-01-27	
ahmad46		AhmadOthman		2024-01-27		2024-01-27	
ahmad47		AhmadOthman		2024-01-27		2024-01-27	
ahmad48		AhmadOthman		2024-01-27		2024-01-27	
ahmad49		AhmadOthman		2024-01-27		2024-01-27	

Figure 5.145: User Connection

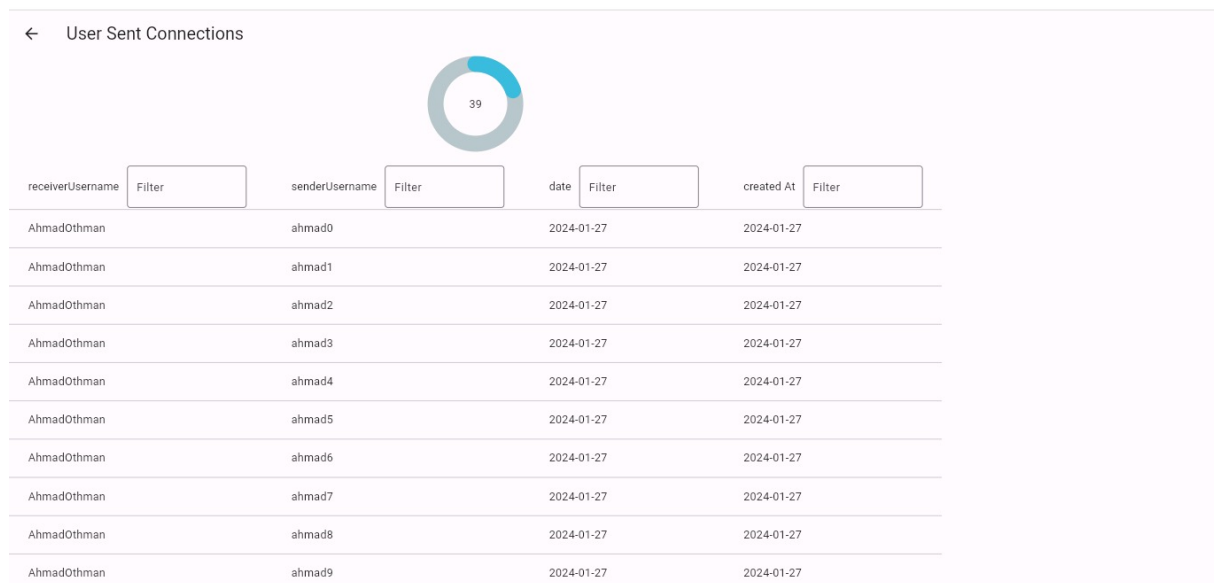


Figure 5.146: Sent Connection table

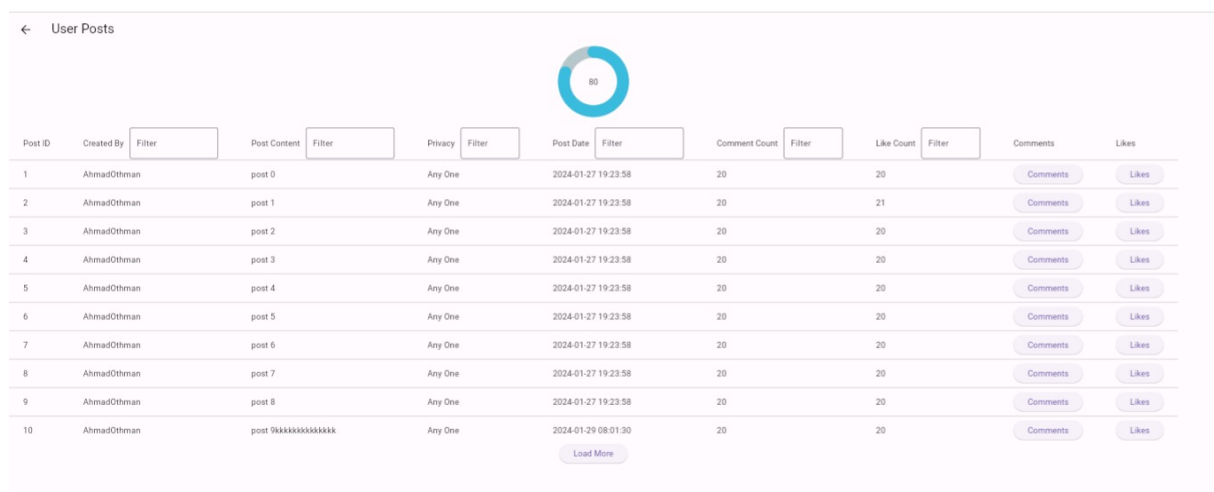
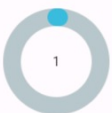


Figure 5.147: Table to display User Posts

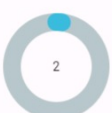
← Educational Levels



ID	Username	specialty	School	description	startDate
1	AhmadOthman	programmer	araba school	2024-01-29	love math

Figure 5.148: educational level table

← Work Exp



ID	Username	specialty	Company	description	startDate
1	AhmadOthman	Front	Asal	2024-01-29	in our company
2	AhmadOthman	back	fothill	2024-01-29	ok my company

Figure 5.149: work experience table

← User Applications

19

idFilter

pageJobIdFilter

usernameFilter

ahmad1	1	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut er
ahmad1	2	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut er
ahmad1	3	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut er
ahmad1	4	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut er
ahmad1	5	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut er
ahmad1	6	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut er
ahmad1	7	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut er
ahmad1	8	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut er

Figure 5.150: User Applications table

← Post Comments

20

IDFilter

Post IDFilter

ContentFilter

DateFilter

1	1	ahmad40	2024-01-29
80	1	ahmad41	2024-01-29
159	1	ahmad42	2024-01-29
238	1	ahmad43	2024-01-29
317	1	ahmad44	2024-01-29
396	1	ahmad45	2024-01-29
475	1	ahmad46	2024-01-29
554	1	ahmad47	2024-01-29

Figure 5.151: Post Comments for each post



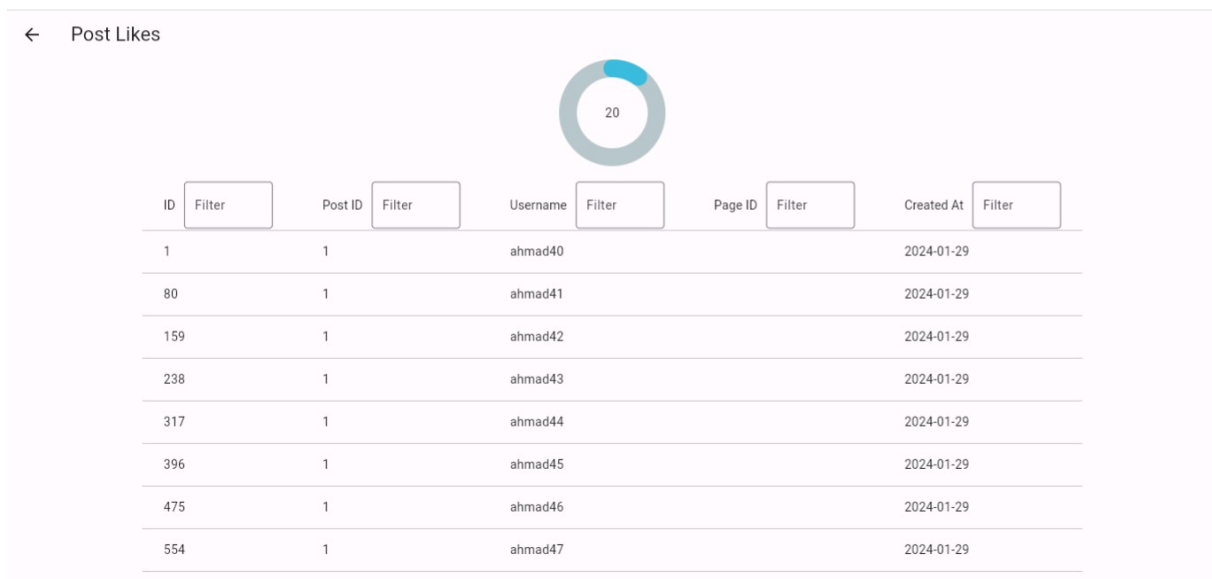


Figure 5.152: Likes for each User Posts

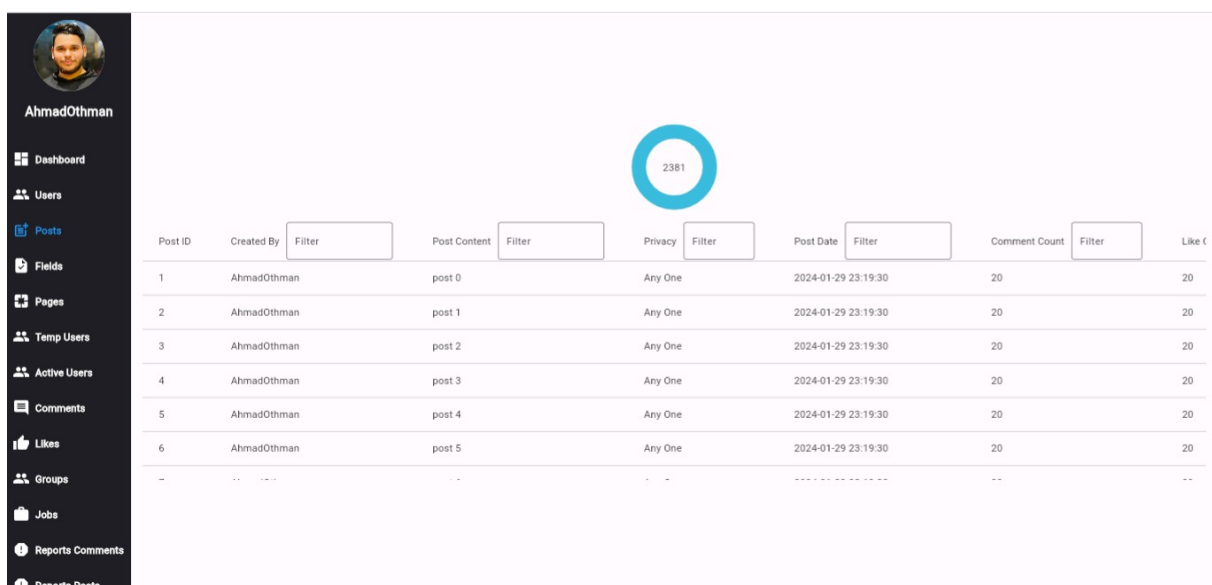


Figure 5.153: Get All Posts

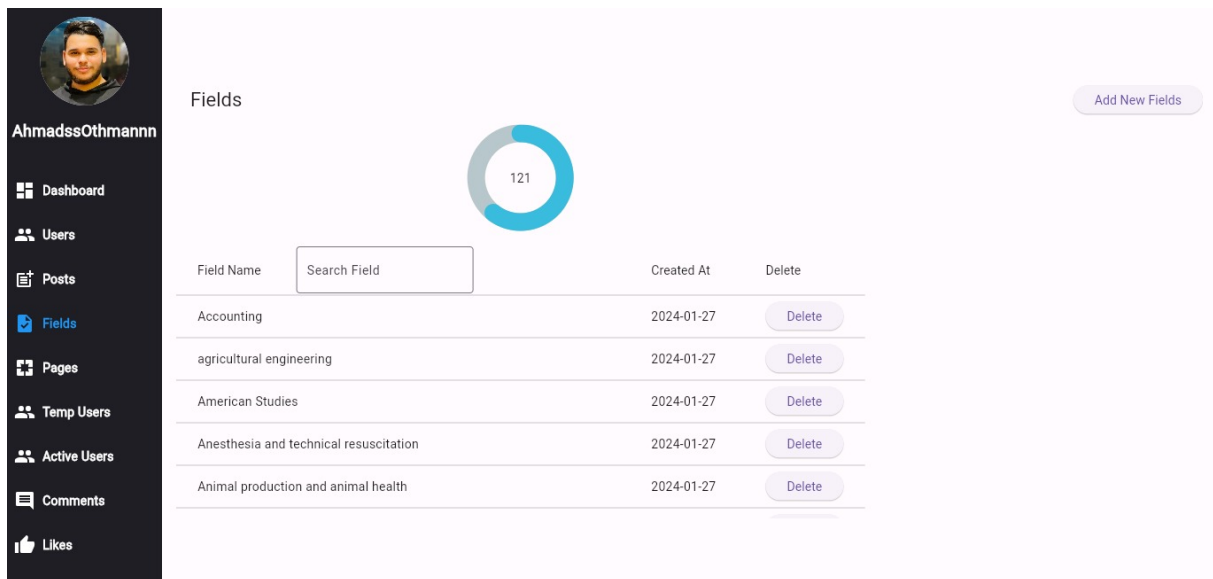


Figure 5.154: Show All Fields

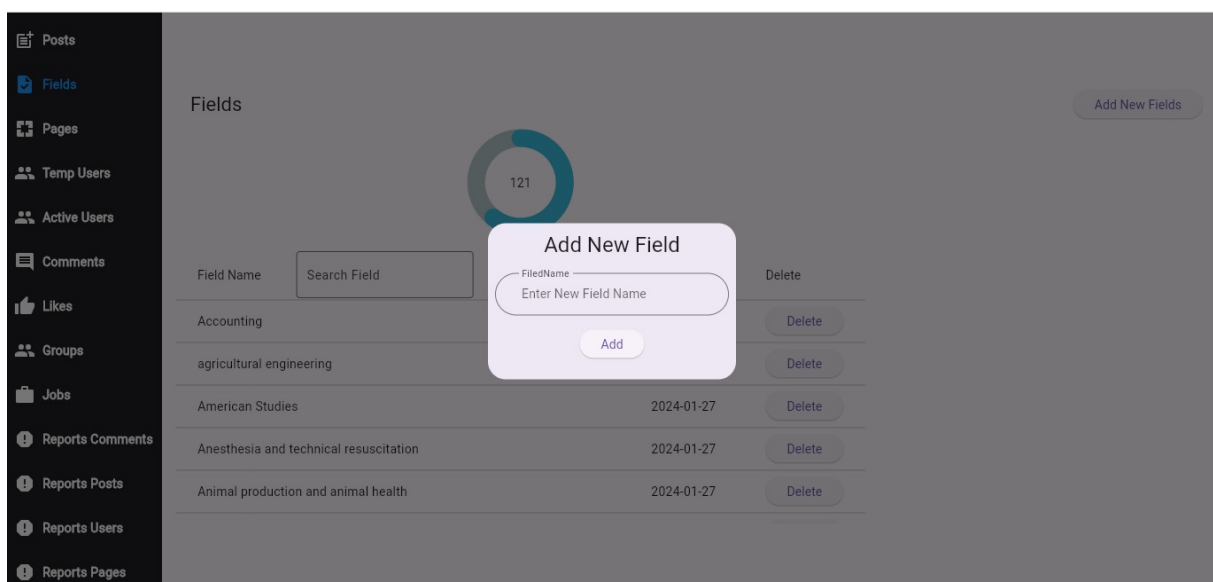


Figure 5.155: Add Field

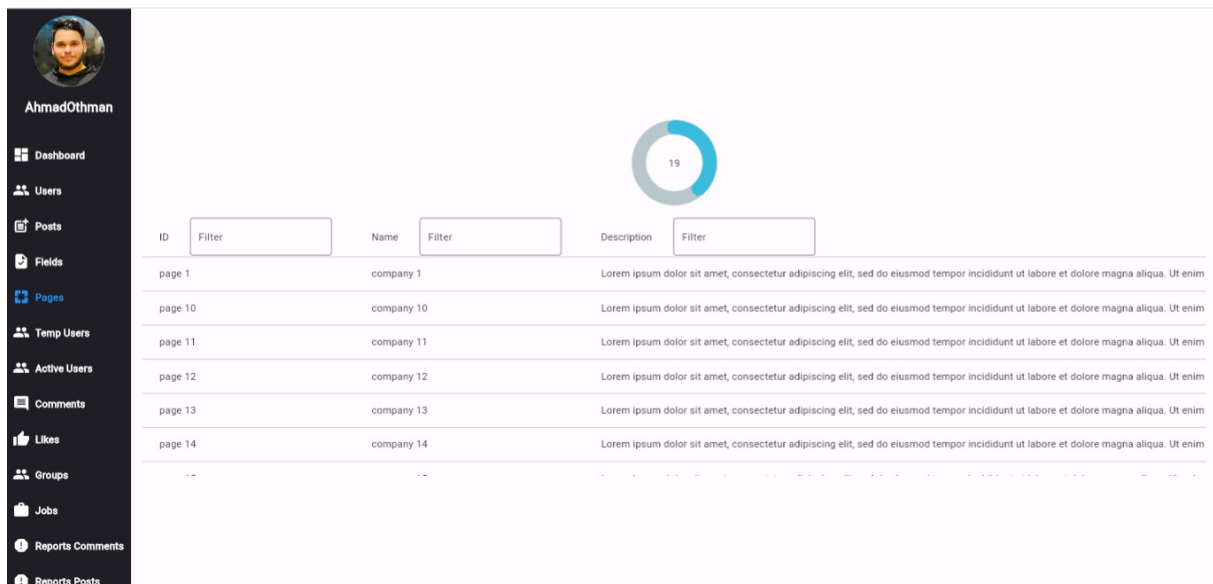


Figure 5.156: Show All Pages

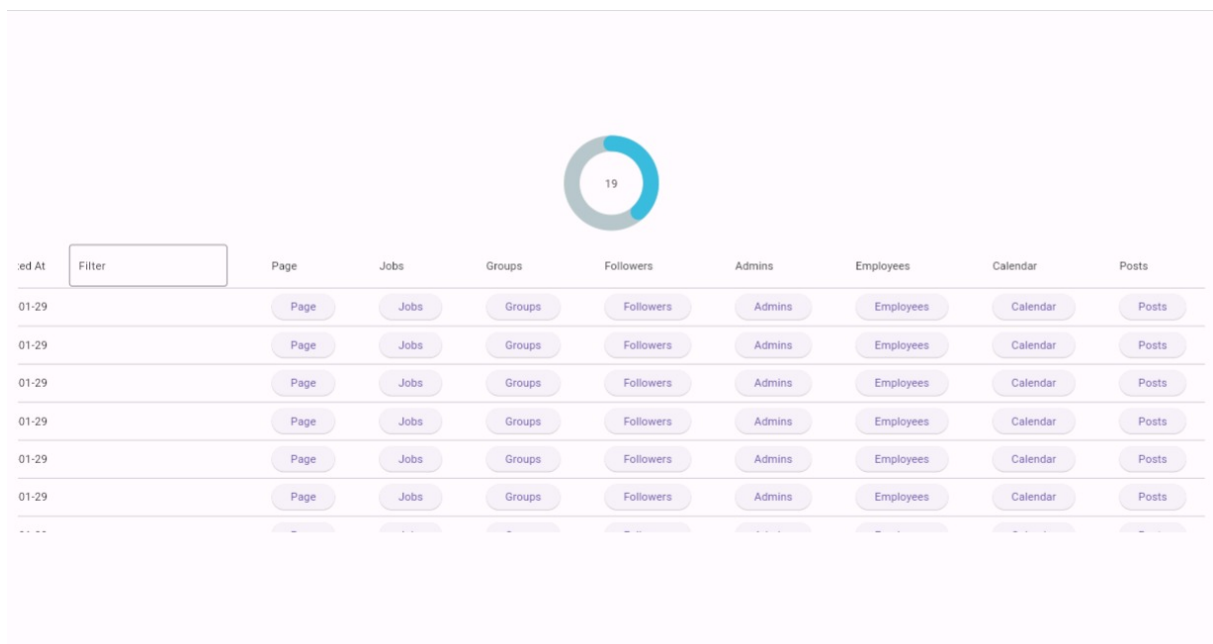


Figure 5.157: Show All Pages

← Page Jobs

19

Job ID	Filter	Page ID	Filter	Title	Filter	Fields	Filter	Description	Filter
1		page 1		Job Title 1		Other		Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim a	
2		page 1		Job Title 2		Other		Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim a	
3		page 1		Job Title 3		Other		Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim a	
4		page 1		Job Title 4		Other		Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim a	
5		page 1		Job Title 5		Other		Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim a	
6		page 1		Job Title 6		Other		Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim a	
7		page 1		Job Title 7		Other		Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim a	
8		page 1		Job Title 8		Other		Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim a	
9		page 1		Job Title 9		Other		Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim a	
10		page 1		Job Title 10		Other		Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim a	
11		page 1		Job Title 11		Other		Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim a	
12		page 1		Job Title 12		Other		Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim a	

Figure 5.158: Show All Jobs for page

← Page Follower

20

Username	Filter	Page Id	Filter	created At	Filter	updated At	Filter
ahmad0		page 1		2024-01-29		2024-01-29	
ahmad1		page 1		2024-01-29		2024-01-29	
ahmad2		page 1		2024-01-29		2024-01-29	
ahmad3		page 1		2024-01-29		2024-01-29	
ahmad4		page 1		2024-01-29		2024-01-29	
ahmad5		page 1		2024-01-29		2024-01-29	
ahmad6		page 1		2024-01-29		2024-01-29	
ahmad7		page 1		2024-01-29		2024-01-29	
ahmad8		page 1		2024-01-29		2024-01-29	
ahmad9		page 1		2024-01-29		2024-01-29	
ahmad10		page 1		2024-01-29		2024-01-29	
ahmad11		page 1		2024-01-29		2024-01-29	

Figure 5.159: Show All Follower for the page

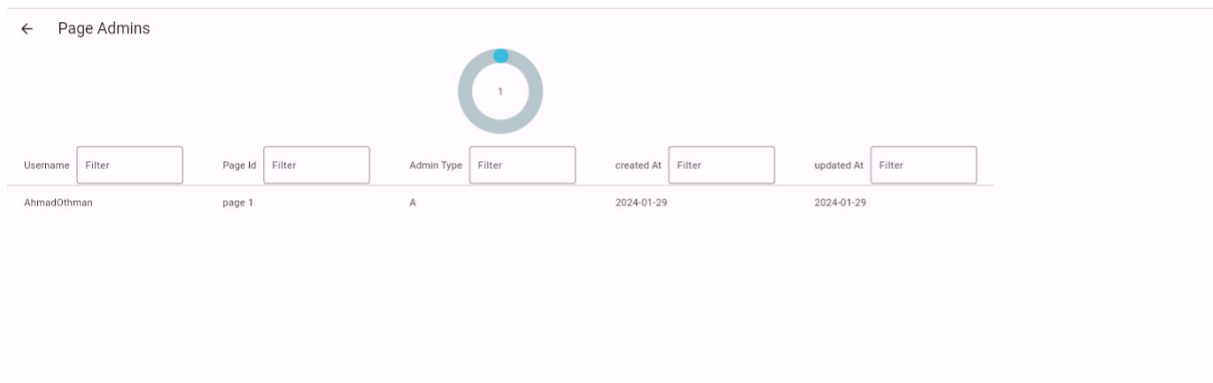


Figure 5.160: show page Admins

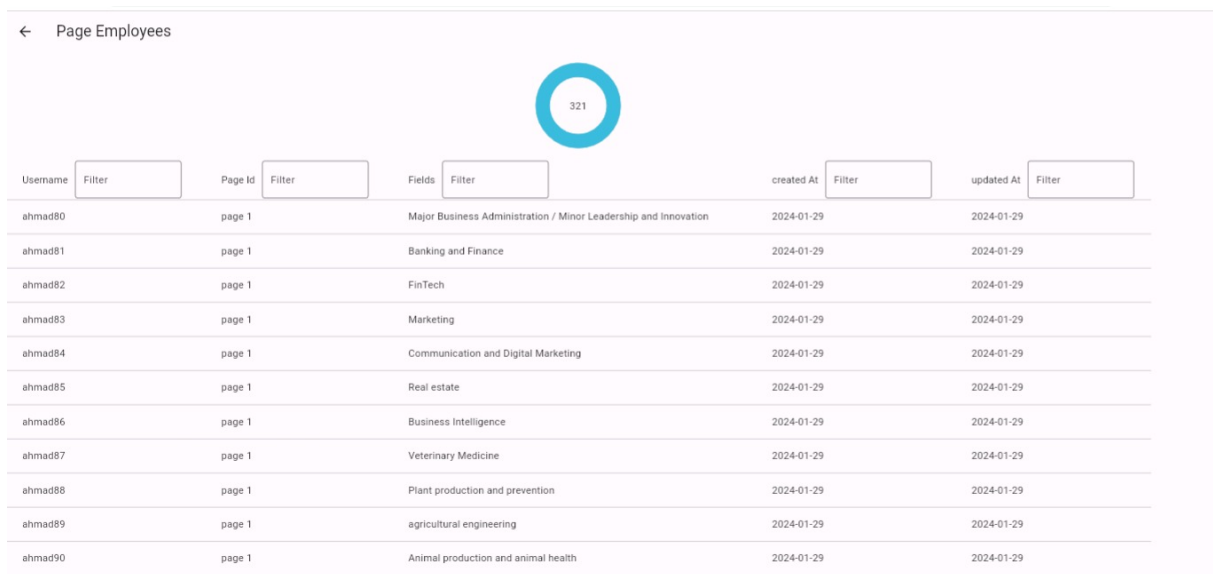


Figure 5.161: show All Employee page

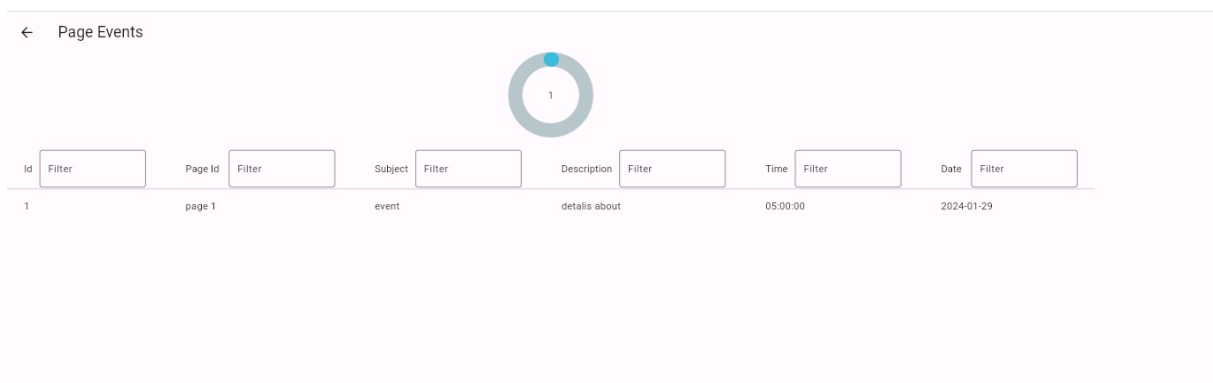



Figure 5.162: show All Events for the page



First Name	Filter	Last Name	Filter	Email	Filter	Phone	Filter	Date of Birth	Filter	Delete
ahamd		majed		dsnkjds		4792793748892		2001-12-12		<a href="#">Delete</a>

Figure 5.163: TempUser table



ID	Filter	Username	Filter
1		AhmadOthman	

Figure 5.164: Active Users



Figure 5.165: Get All Comments

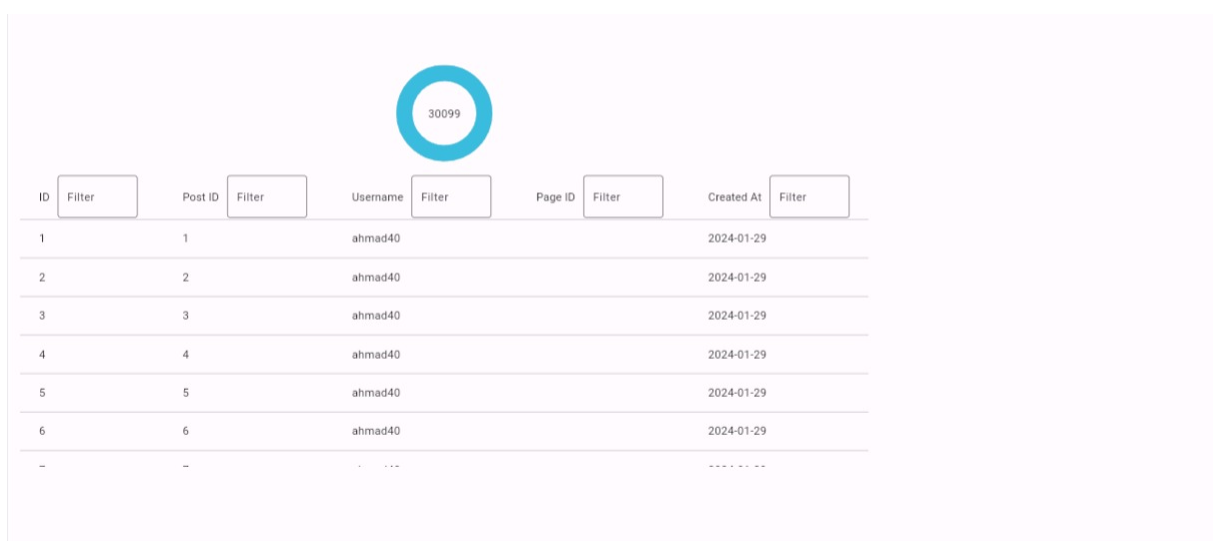


Figure 5.166: Get All Likes



Figure 5.167: Show All Jobs

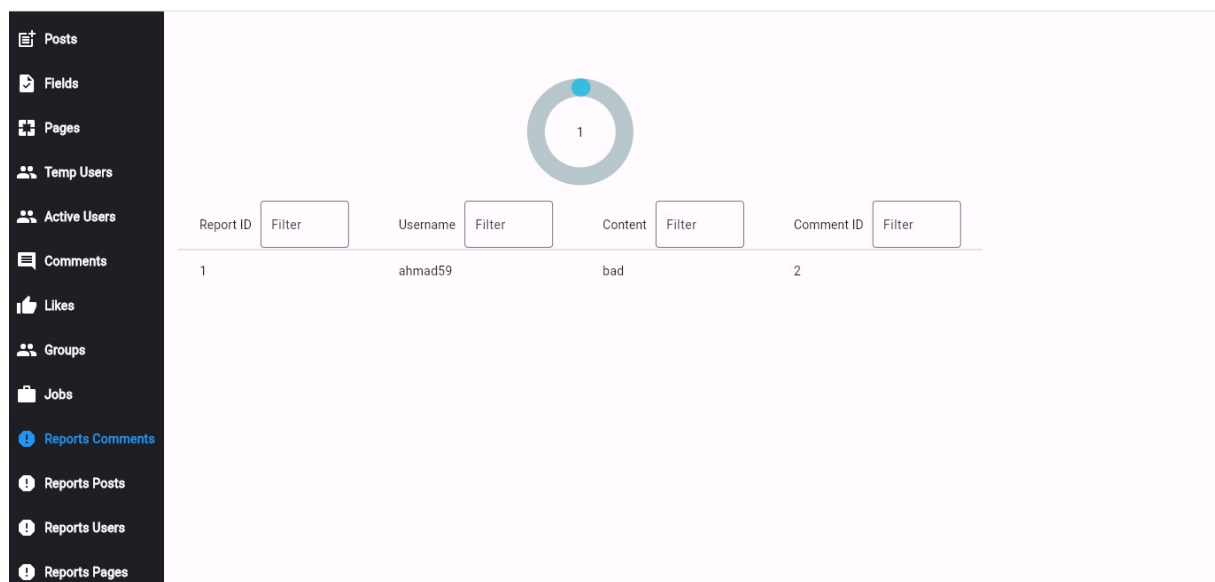


Figure 5.168: Reported Comment



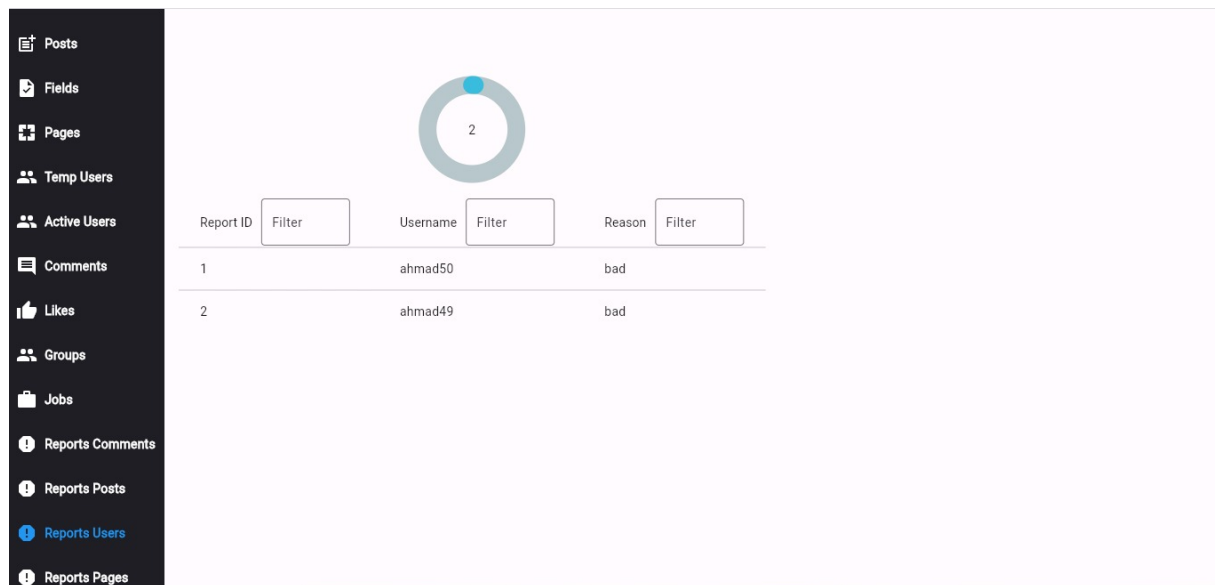


Figure 5.169: Reported Users



Figure 5.170: Reported Pages

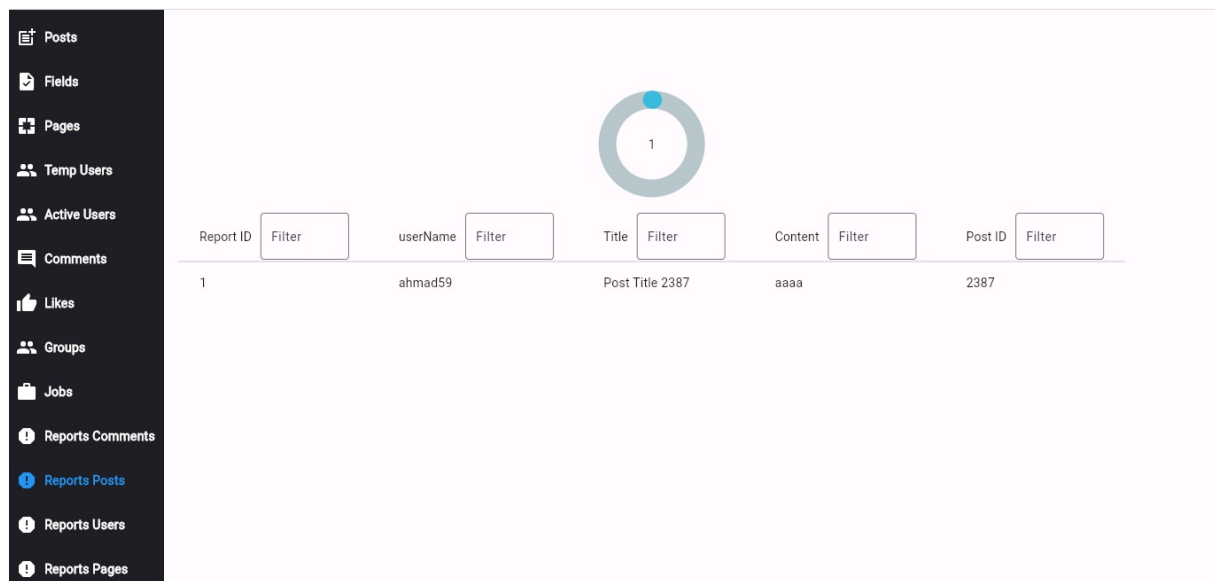


Figure 5.171: Reported Posts

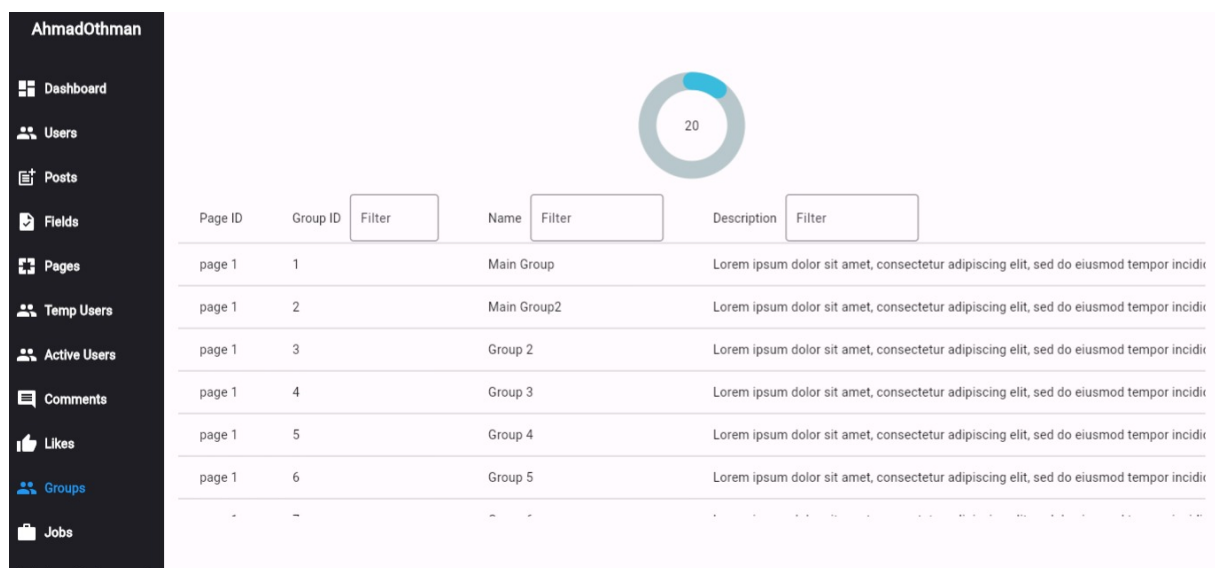


Figure 5.172: Show All Groups

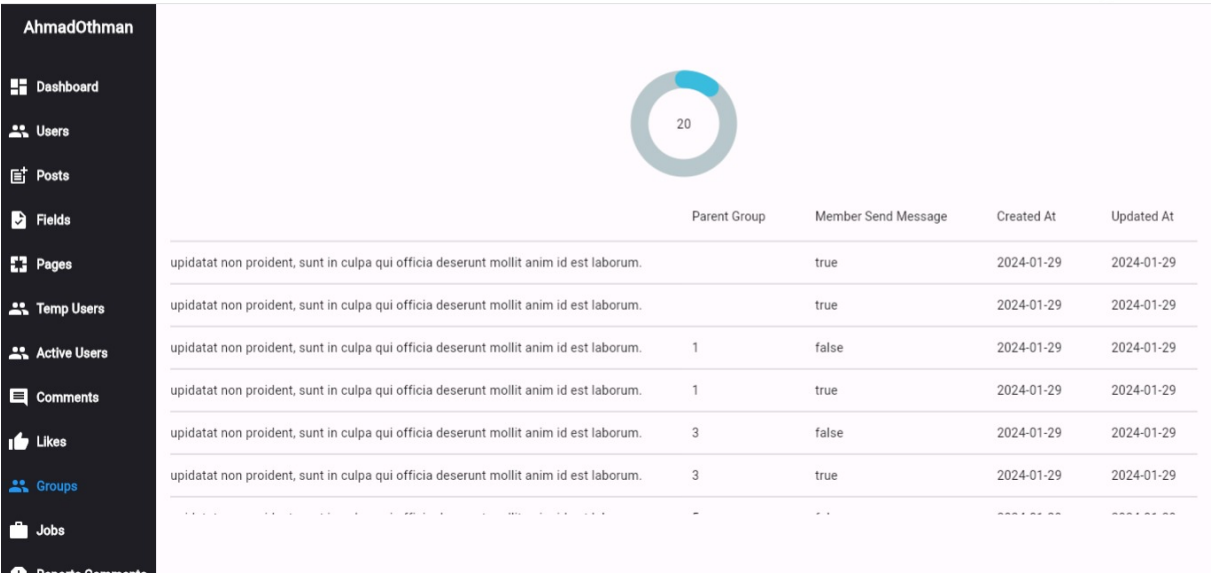


Figure 5.173: Show All Groups

# Chapter 6

## Results & Discussion

Our application provides a very large number of features that serve both users and companies/organizations, by integrating the best features found in social networking sites, business organization sites, and corporate structuring and organization sites in one app, which makes it easier for the user to organize his business and professional life, and connect a large number. Some of the features are distributed on several sites in one site, which contributes to their proper use.

The most important features that our application offers were mentioned in this report. There is a huge amount of validation and a very large number of response possibilities from the server that were not mentioned.

During the application development process, we faced several problems that made it clear that using other technologies might be more appropriate than the technologies we chose.

### **Why not to use MySQL?**

Through our experience, and after completing a big part of the project, it became clear that using MySQL is not the most appropriate option for social networking sites, as its use led to limiting the features that the app can provide, as during the work it became clear to us that using another type of Databases (NoSQL) will be better to our needs, for example in some cases we need to store a specific flag or parameter for a specific number of rows in a table ( The proportion of these rows is very small relative to the total rows of the table) and thus we need to add another column which has no value in most of the rows, or adding new table to store these different flags.

Therefore, from our experience, we do not recommend using this type of database with this type of application.

### **Why not to use Flutter?**

Flutter framework is very good in terms of building regular applications that work on both the web and mobile, but when you delve deeply into applying some technologies such as WebRTC or Server Sent Events, you will find a very large deficiency in the libraries, in addition to being full of errors.

### **Problems we encountered in WebRTC**

When using SFU Structure, connection cannot be relied upon definitively, as it sometimes happens that a user enters the meeting and cannot see the rest of the users, so he need to leave and then join back again to be able to see them. There are more than one possibility for this problem to occur, such as: no reliability on the connection itself, the server running on my personal computer cannot handle a large number of connections, the flutter builds the page before the data arrives, the data arrives but does not appear due to an error in the library itself (because the flag provided by the library indicates that the stream has arrived and is being displayed). The complexity of this problem is difficult to solve directly, so we were forced to take detours to mitigate its appearance.

# Chapter 7

## Conclusion & Future Work

### 7.1 Conclusion

The main goal of our project is to amplify the Growth for both companies/organizations and individuals. our application offer a large number of features and combine the main features of other sites such as social networking, organizing user tasks and work, providing huge advantages for companies to structure and organize their work.

### 7.2 Future Work

There are many areas that can be improved in Growify, including:

1. Add more privacy settings.
2. Add more search and filtration on screens.
3. Increase security.
4. Expand the features to include a larger number of features of other sites .
5. Development on video meeting to be reliable every time a user enters the meeting.
6. Record the meeting by recording the stream on the server itself and not on the user's device.