

Remote Freelance Service Management System Web Application

*A Project Submitted in Partial Fulfillment of the Requirements for the
Degree of
Bachelor of Science in Computer Science and Engineering*

by

Md Raihan Alam

ID: CSE 06607759

Supervised by: **Md Samiul Islam**
Lecturer



Department of Computer Science and Engineering
STAMFORD UNIVERSITY BANGLADESH

December 2022

Abstract

The main objective of Remote Freelance Service Management System is to provide best remote Work and Service experience for local and international customers and freelancers. Remote work is very popular and smart work in now and future. We managed to make a system where user can just need to verify profile and wallet and make portfolio as a freelancer and post job as a customer.

We developed a system that make the work super easy for both freelancer and customer. The system business model will skip spammers and will make secured payments and job management. Optimized database model will help users to interact dynamically with the system. Latest tools & technologies will make this system faster and reliable and scalable.

Approval

The project report “Remote Freelance Service Management System Web Application” submitted by Md. Raihan Alam ID:CSE06607759, to the Department of Computer Science & Engineering, has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science (B.Sc.) in Computer Science & Engineering and as to its style and contents.

Board of Examiner's Name, Signature and Date:

.....

(Board Member 1)

Date:

(Board Member 2)

Date:

(Board Member 3)

Date:

Supervisor's Signature and Date:

.....

Supervisor Name

Date:

Declaration

We, hereby, declare that the work presented in this Thesis Project is the outcome of the investigation performed by us under the supervision of Md Samiul Islam, Lecturer, Department of Computer Science & Engineering, Stamford University Bangladesh. We also declare that no part of this Project and thereof has been or is being submitted elsewhere for the award of any degree or Diploma.

Signature and Date:

.....
Student Name: Md Raihan Alam
Date: 27.11.2022

Dedicated to

To my mother and father, specially to my father who was my super power and motivation of work.

Acknowledgement

First and foremost, we are grateful to Almighty creator without whose blessing, this project would not have been successful. We would like to express our sense of gratitude to Md. Samiul Islam, Lecturer of the Department of Computer Science & Engineering, Stamford University Bangladesh, for his unconditional guidance, insights, immense patience, unlimited encouragement and inspirations throughout this whole project. He is the one who supported us every step of the way in every possible way. Without him, this project would not have come to fruition. He provided the valuable technical advice and pointed us in the right directions which were much required for a project like this. We are truly grateful for having a teacher like him as our supervisor. We specially owe thanks to all the teachers of the faculty of Computer Science & Engineering, Stamford University Bangladesh for their help, valuable suggestions and discussions. Moreover, we would like to thank all our friends who have supported and helped us throughout this project. Last of all we would also like to thank our parents who have walked along with us in every step of our life and always encouraged and inspired us to do greater things in our life.

Table of Contents

List of Figures	vii
1. Introduction	1
1.1 Motivation	1
1.2 Objective	1
1.3 Problems and Definition	1
1.4 Proposed System	2
1.5 Market & Scope	2
2. Planning & Analysis.....	3
2.1 Planning	3
2.2 Requirement Analysis	4
2.3 System Analysis	5
2.4 Feasibility Study	7
2.5 Tools & Technology	8
3. System Design.....	9
3.1 Use Case Diagram.....	9
3.2 Entity Relationship Diagram.....	17
3.3 Database Schema Diagram	18
3.4 Data Flow Diagram	22
4. Implementation	23
4.1 Configuration	23
4.2 Interfaces	24
5. Conclusion.....	52
5.1 Limitations	52
5.2 Future Work	52
References	55

List of Figures

2.1.1	Seven phases of system development life cycle	3
2.3.1	Model View Template (MVT) Architecture.....	6
3.1.1	Account & User System Use Cases	9
3.1.2	Portfolio and Job Post System Use Cases.....	10
3.1.3	Real-time Chatting System Use Cases.....	11
3.1.4	Closed Wallet System Use Cases	12
3.1.5	Work & File Sharing System Use Cases.....	13
3.1.6	Search & Filtering System Use Cases.....	14
3.1.7	Dashboard & Monitoring System Use Cases.....	15
3.1.8	Notification System Use Cases	16
3.2.1	Entity Relationship Diagram.....	17
3.3.1	Database Schema Part 1	18
3.3.2	Database Schema Part 2	19
3.3.3	Database Schema Part 3	20
3.3.4	Database Schema Part 4.....	21
3.4.1	5 Level Data Flows Diagram	22
4.2.1	Interface for public user as landing page	24
4.2.2	Interface for user creation with a role	25
4.2.3	Interface for user signin	25
4.2.4	Interface for customer homepage showing portfolios.....	26
4.2.5	Interface for freelancer home showing job posts	26
4.2.6	Interface for user profile and profile management.....	27
4.2.7	Interface for profile information update.....	27
4.2.8	Interface for user profile picture add/change	28
4.2.9	Interface for user personal information change	28
4.2.10	Interface for old password change form for logged user	28
4.2.11	Interface for inactive wallet with activation form.....	29
4.2.12	Interface for activate wallet, options for only verified customer	29
4.2.13	Interface for user profile verification application	30
4.2.14	Interface for verified application.....	30
4.2.15	Interface for validate profile with verified icon	30
4.2.16	Interface for money deposit checkout for customer.....	31
4.2.17	Interface for payment gateway API, showing SSLCommerz API.....	31
4.2.18	Interface after deposit with message and transaction data in wallet for customer	32
4.2.19	Interface for all successful deposits for customer	32
4.2.20	Interface for all kinds of transaction for both user.....	32
4.2.21	Interface for portfolio creation for freelancer	33

4.2.22	Interface for all portfolio manage for freelancer.....	33
4.2.23	Interface for portfolio edit and multi image upload for freelancer	34
4.2.24	Interface for portfolio view with hire request and customer reviews for customer	35
4.2.25	Interface for portfolio gallery image zoom on mouse hover on image.....	36
4.2.26	Interface for profile view of freelancer for customer.....	36
4.2.27	Interface for job posts creation for customer	37
4.2.28	Interface for own job posts manage buttons for customer	37
4.2.29	Interface for job post edit and update for customer	38
4.2.30	Interface for job posts details with bidding features for freelancers.....	38
4.2.31	Interface for chatting between two user	39
4.2.32	Interface for offering a service to customer in chat interface for freelancer.....	39
4.2.33	Interface for ordering a service to freelancer in chat interface for customer	40
4.2.34	Interface for order payment checkout for customer.....	40
4.2.35	Interface for payment confirmation for customer	41
4.2.36	Interface for all kinds of order for customer	41
4.2.37	Interface for order checkout for customer.....	41
4.2.38	Interface for recent orders for freelancer	42
4.2.39	Interface for accepting order and start work checkout for customer	42
4.2.40	Interface for running work with tracker and file sharing for freelancer	43
4.2.41	Interface for complete work for customer.....	44
4.2.42	Interface for running work with work tracking for customer	44
4.2.43	Interface for review & ratings after a work is complete for customer	45
4.2.44	Interface for wallet after deposit received after a complete work for freelancer	45
4.2.45	Interface for successful payments after a complete work for customer.....	45
4.2.46	Interface for withdrawal checkout for freelancer.....	46
4.2.47	Interface for all withdrawal requests with status for freelancer	46
4.2.48	Interface for incomplete order payment checkout	46
4.2.49	Interface for notifications for notifying a actions to an user	47
4.2.50	Interface for user chat box shows connected user to chat.....	47
4.2.51	Interface for settings to change profile, personal information and password	47
4.2.52	Interface for dashboard with recent actions, balance and some statistics for customer.....	48
4.2.53	Interface for dashboard with recent actions, balance and statistics for freelancer.....	48
4.2.54	Interface for recent bids for customer	49
4.2.55	Interface for recent bids for freelancer.....	49
4.2.56	Interface for all offers for customer	49
4.2.57	Interface for running orders for both user	50
4.2.57	Interface for running orders for both user	50
4.2.59	Interface for search results by keywords shows job post for freelancer	50
4.2.60	Interface for admin panel with all database model access for admin	51

1 Introduction

1.1 Motivation

From last 10 years our local industry is growing with digitization and information technology based work. And after Covid19 pandemic peoples are looking for remote workers or remote services much more. Lot of my connections contacted with me for different kinds of digital services. But I wasn't professional for those work. And some of my connections talks with me about the remote marketplace as an opportunity. And my friends, juniors and other peoples who are actually talented for different kinds of work and they are also findings work in international marketplace or elsewhere but they are surviving with lot of complications. Also, I experienced lot of issues, spammers, payment methods, communication gaps, and lot of complexities in marketplaces. And after that I feel this problem and took it as opportunity.

I thought about a System which will solve and fulfill basic and standard needs of a freelancer and customer.

1.2 Objectives

The main goal is to develop a reliable, secured, easy marketplace for both freelancer and customer. We will cover all the modules and features to manage a work smoothly. Will try to make a friendly User Interface and good User Experience application. Will make a standard business model to secured the marketplace from spammers.

Design and develop a micro-service architecture system that will be sub system independent. An easy business model which will support all kinds of activities with the system.

1.3 Problems & Definitions

The world is growing with remote base market in booth product base industry and service based industry, and our country isn't out of this. People are looking for talents for his work quickly for different kinds of task. But in our local industry there is no successful marketplace for remote work. And our local customers don't have facilities to take freelance service from international marketplace cause of payment system and communication skills etc.

Unemployment is a big problem of our country, our people are getting degree from various fields but proper guidelines and organization or market are not capable to give them work.

Our talents are working on international marketplace long time, they are earning huge amount of foreign currency every year and every marketplace cuts a margin for their services. That means we are loosing some foreign currency every year. But we can reduce loosing this margin from our economy.

We can contribute a solution as part of this problem.

1.4 Market & Scope

From last 2020 or the COVID19 situation people became more depended on internet or devices. People are looking for remote talent for their digital work. Also in Bangladesh, now freelancing is a popular work and also demandable work in both local and international market.

Last few years there was lot of startup who are selling learning contents of technologies, that's mean huge number of talents are generating every day in the market and they are looking for this kind of market in local or international industry.

According to e-Commerce Association of Bangladesh (e-CAB), the F-commerce market is gradually expanding in Bangladesh. Business transactions with respect to the same worth Taka 7,000 crore took place in 2020 over traditional e-commerce mediums, and an astounding Taka 1,000 crore was contributed by F-commerce even as over 400,000 new entrepreneurs are engaged in F-commerce in the country and the number of entrepreneurs is increasing. [1]

Data published in Meta's advertising resources indicates that facebook had 44.70 million users in Bangladesh in early 2022, one of the top ten countries in terms of using the social media.

As per Statista, in 2020, Facebook's subscriber base in Bangladesh was 39 million even as other reports suggest capital city Dhaka alone has over 22 million Facebook users while the F-commerce market size in the country is approximately around Taka 312 crore. Further, 50 percent of Facebook stores are owned by females. [2]

Also, the e-commerce industry is forecasted 2.1 billion in 2020, 2.5 billion in 2021, 2.9 in 2022 and in 2023 this is to hit \$3.1 billion, if there is provided a stable internet connection and an easy online payment system. [3]

Also small e-commerce and F-commerce are becoming a regular part of humans life. Where they commerce will need more freelance work to grow their business. Like product design, graphics design, digital marketing, content writing, web design & development etc. Which can be a big lead for future market.

1.5 Proposed Systems

- A Reliable Marketplace
- Basic Accounts & Profile Management
- Manual Freelancer & Customer Verification
- Impressive Portfolio and Job Content Management
- Breakthrough Communication
- Easy Offer and Order Management
- Easy Work Process & File Management
- Closed Wallet System with Virtual Coin
- Cash Deposit & Withdrawal Process
- Automatic Payment Process
- Automatic Notifying System
- Report & Review Collection
- Search and Filtering System
- Business Intelligence flavored dashboard.

2 Planning & Analysis

2.1 Planning

A plan to make an application which will cover all the business model and requirements. We decided to make some micro application to cover all the functionalities.

- Account & User Management Application
- Content Management Application
- Messaging Application
- Wallet Application
- Work Application
- Search Application
- Notification Application
- Dashboard Application

As our project strategy we decided to choose Agile Model to get successful, with a great product.

Agile SDLC model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product. Agile Methods break the product into small incremental builds. These builds are provided in iterations. Each iteration typically lasts from about one to three weeks.

Agile will be a great model for this project. We will follow the extreme programming approach to solve all problems. Requirement gathering and fast development is the main goal in extreme programming. A big project comes with an easy solution with agile. And that is one man army project, and I think that will be a great model for this.

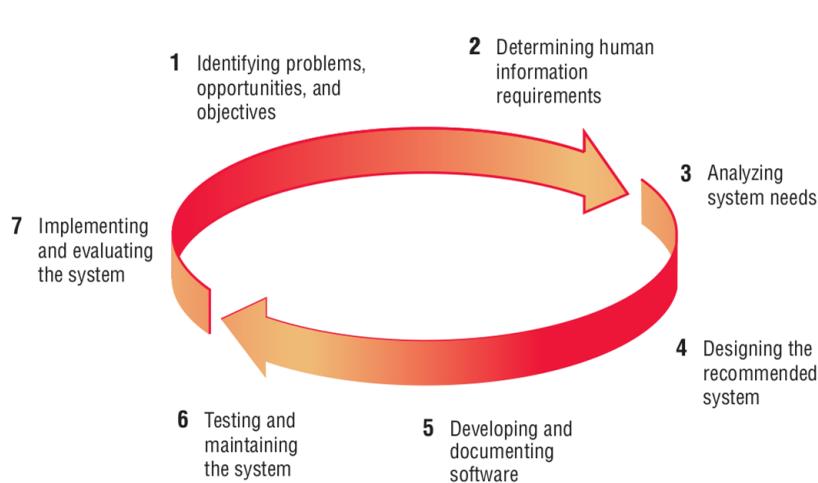


Figure 2.1.1 Seven phases of system development life cycle.

2.2 Requirements Analysis

Requirements for all User:

- In this application user can Sign-up & Sign-in.
- Need email verification before 1st Sign-in.
- Need to Complete Profile providing some valuable information.
- Have to complete wallet information for wallet.
- Have to apply for profile verification with a simple email.
- After profile verification user can interact with this system.

Requirements for Freelancer:

1. Can browse posted jobs by his expertise area.
2. Can create a portfolio.
3. Can bid on a customer job post.
4. Can send message to customer.
5. Will receive important notification from customers.
6. Will send service offer to customer.
7. Can accept or cancel order.
8. Can send file in a running work.
9. Can send files to the customer in an ongoing order.
10. Can send update in work track to customer.
11. Will confirm after work is done.
12. Will received payment from customer.
13. Can give a feed-back about a customer.
14. Freelancer can withdraw money from his account.

Requirements for Customer:

1. Can post his job/task.
2. Can browse freelancer portfolio by searching their job information.
3. Can send messages to connected freelancer.
4. Can send hire request to a freelancer by his post.
5. Can deposit money in wallet.
6. Can place order on a offer from freelancer.
7. Can confirm a order by confirming payment from wallet.
8. Will receive important notification from freelancer.
9. Can send and download files from a running work.
10. Can report about bad experience in a running work.
11. Will confirm work is done and payment will be success automatic.
12. Can give review to a freelancer work.
13. Can monitor all job status from dashboard.

2.3 System Analysis

In our system there will be three actor one is Freelancer, Customer and Admin.

Admin have the superpower in this system, he can access the admin panel, where every data model and data will show and he can interact all kinds of CRUD action. Admin can also give a sub power of him by giving different kinds accessibility and permissions.

Also, admin can set profile role to the user and can give set of rules for them. Admin will verify user's profile, withdrawals etc.

Freelancer and Customer have limitations of interactions dynamically decided by business models and logic.

We decided to use Python as a core programming language, in back-end we will use Django as a web framework. And JavaScript for front-End interaction. Bootstrap as UI framework . PostgreSQL for database.

There are lot of back-end framework but we choose Django.

Why Django is best from others? [4]

With Django, you can take web applications from concept to launch in a matter of hours. Django takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source.

- **Ridiculously fast** - Django was designed to help developers take applications from concept to completion as quickly as possible.
- **Fully loaded** - Django includes dozens of extras you can use to handle common web development tasks. Django takes care of user authentication, content administration, site maps, RSS feeds, and many more tasks — right out of the box.
- **Reassuringly secure** - Django takes security seriously and helps developers avoid many common security mistakes, such as SQL injection, cross-site scripting, cross-site request forgery and clickjacking. Its user authentication system provides a secure way to manage user accounts and passwords.
- **Exceedingly scalable** - Some of the busiest sites on the planet use Django's ability to quickly and flexibly scale to meet the heaviest traffic demands.
- **Incredibly versatile** - Companies, organizations and governments have used Django to build all sorts of things — from content management systems to social networks to scientific computing platforms.

What does Django Structure Look Like? [5]

In a traditional data-driven website, a web application waits for HTTP requests from the web browser (or other client). When a request is received the application works out what is needed based on the URL and possibly information in POST data or GET data. Depending on what is required it may

then read or write information from a database or perform other tasks required to satisfy the request. The application will then return a response to the web browser, often dynamically creating an HTML page for the browser to display by inserting the retrieved data into placeholders in an HTML template. Django web applications typically group the code that handles each of these steps into separate files:

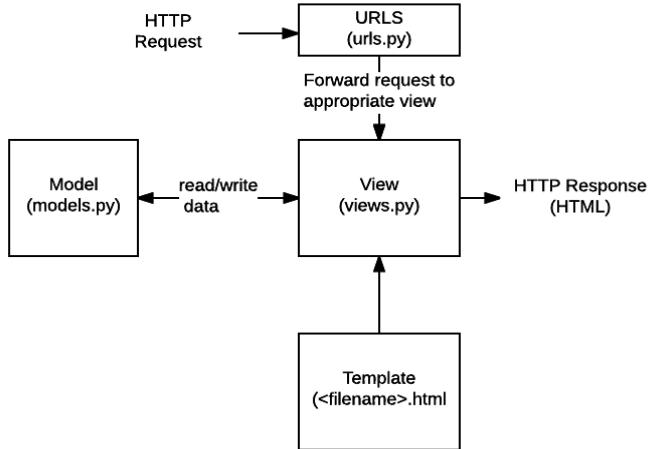


Figure 2.3.1 – Model View Template (MVT) Architecture. [5]

- **URLs:** While it is possible to process requests from every single URL via a single function, it is much more maintainable to write a separate view function to handle each resource. A URL mapper is used to redirect HTTP requests to the appropriate view based on the request URL. The URL mapper can also match particular patterns of strings or digits that appear in a URL and pass these to a view function as data.
- **View:** A view is a request handler function, which receives HTTP requests and returns HTTP responses. Views access the data needed to satisfy requests via *models*, and delegate the formatting of the response to *templates*.
- **Models:** Models are Python objects that define the structure of an application's data, and provide mechanisms to manage (add, modify, delete) and query records in the database.
- **Templates:** A template is a text file defining the structure or layout of a file (such as an HTML page), with placeholders used to represent actual content. A *view* can dynamically create an HTML page using an HTML template, populating it with data from a *model*. A template can be used to define the structure of any type of file; it doesn't have to be HTML

In chat module we used ASGI server for asynchronous programming.

ASGI (*Asynchronous Server Gateway Interface*) is a spiritual successor to WSGI, intended to provide a standard interface between async-capable Python web servers, frameworks, and applications. Where WSGI provided a standard for synchronous Python apps, ASGI provides one for both asynchronous and synchronous apps, with a WSGI backwards-compatibility implementation and multiple servers and application framework. [6]

And this is the actual architecture of our system.

2.4 Feasibility Study

It was decided that the proposed system will developed as a web based system that be operated by freelancers and customers. The very first phase in developing any life cycle is preliminary investigation. In the preliminary study we examine the project feasibility. This project has been tested in the following areas of feasibility:

- **Operational Feasibility**
- **Technical Feasibility**
- **Financial Feasibility**
- **Market Feasibility**
- **Organizational Feasibility**

Operational Feasibility:

- ✓ Microservice based architecture makes system independent.
- ✓ The system is designed in such a way that it is easy to operate.
- ✓ Easy business model makes the system more reliable.

Technical Feasibility:

- ✓ The system has very simple structure and easy to understand body.
- ✓ Impressive User Interface and Good User Experience
- ✓ The system is platform independent and browser independent.
- ✓ The system can be expanded if so decided.
- ✓ Latest technologies makes this system more faster and secured.

Financial Feasibility:

- ✓ Closed Wallet System will handle all financial services.
- ✓ Every withdrawal will margin 9% of withdrawal amounts.
- ✓ Our remote idea will reduce office cost and others cost.

Market Feasibility:

- ✓ There is no successful freelance service marketplace in our local industry.
- ✓ Our latest technologies and services will attract customer and freelancers.

Organizational Feasibility:

- ✓ We can work as freelancer in our marketplace as startup.
- ✓ And we can operate all kinds of action from our home.
- ✓ We can grow up our system remotely by developers.

2.5 Tools & Technologies

Programming Languages -

Python 3.8

Javascript

Web Framework -

Django version 4.0.2

Channels -

ASGI/Channels version 3.0.4

Database -

PostgreSQL

Cache Memory -

Redis

UI Framework -

Bootstrap 4.0

Security -

reCaptcha

Text Editor -

CKEditor

JavaScript Libraries -

Jquery 3.2.1

Ajax

Moment.js

Popper

Slick

Font Script -

Font Awesome

Google Font API

Version Controlling System -

Github

Payment Gateway -

SSLCommerz

Operating System -

Linux - Ubuntu

3 System Design

3.1 Use Case Diagram

There are some major interactions with the system as a user. We showed here booth customer and freelancer as actor. This use case shows that user can signup and then he needs to verify his email by a verification link from his given email. Without verifying email user can't sign-in. After a successful step, user can see his home page, profile, verification. Without verification user can't interact with other features.

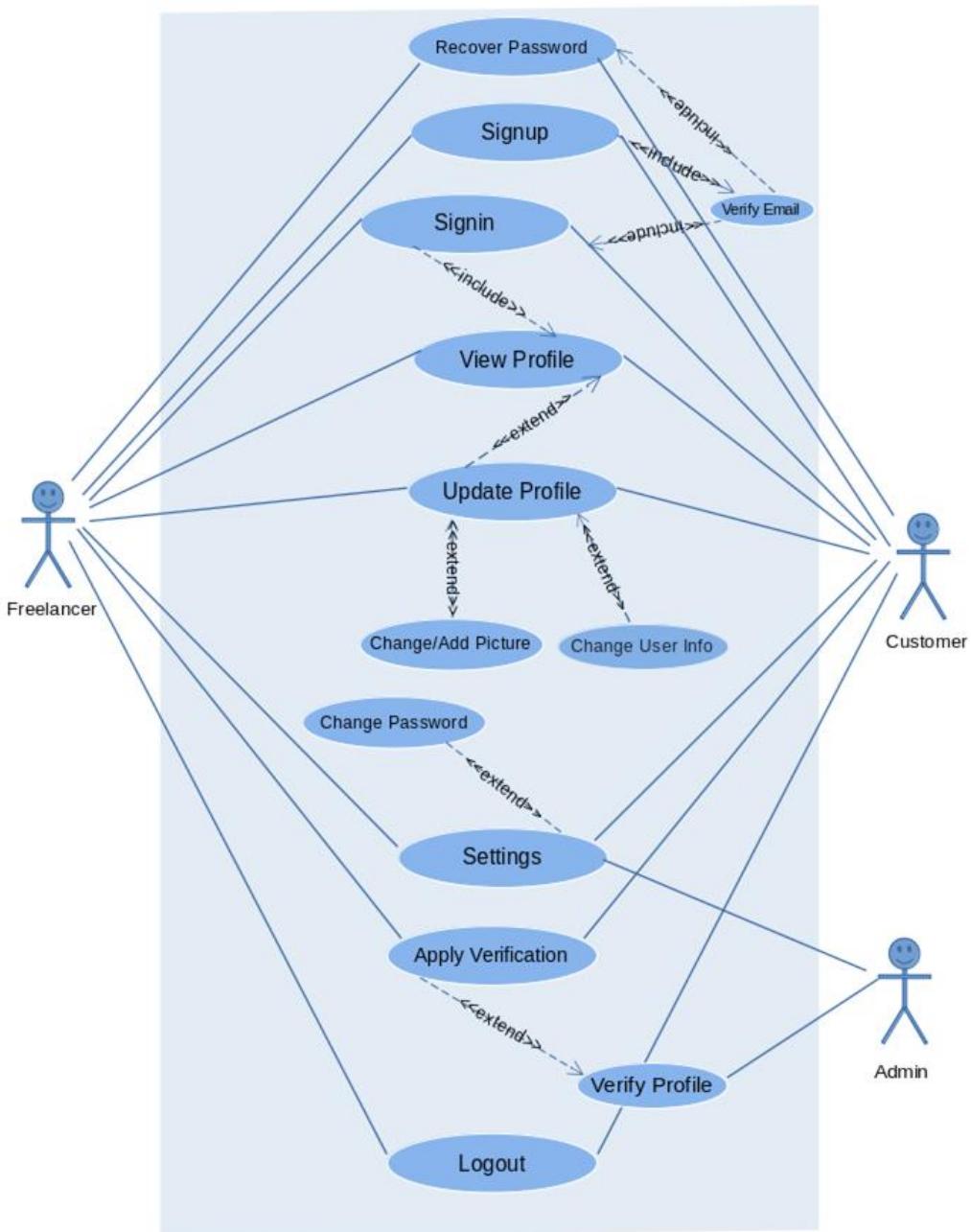


Figure 3.1.1 – Account & User Management System Use Cases.

This is Main App which will allow and manage all action for portfolio creation design and job posting content. Which core system as follows Content Management System. In this system freelancer can bid on a customer post also customer can send hire request by freelancer port. In this system portfolio defined as Port.

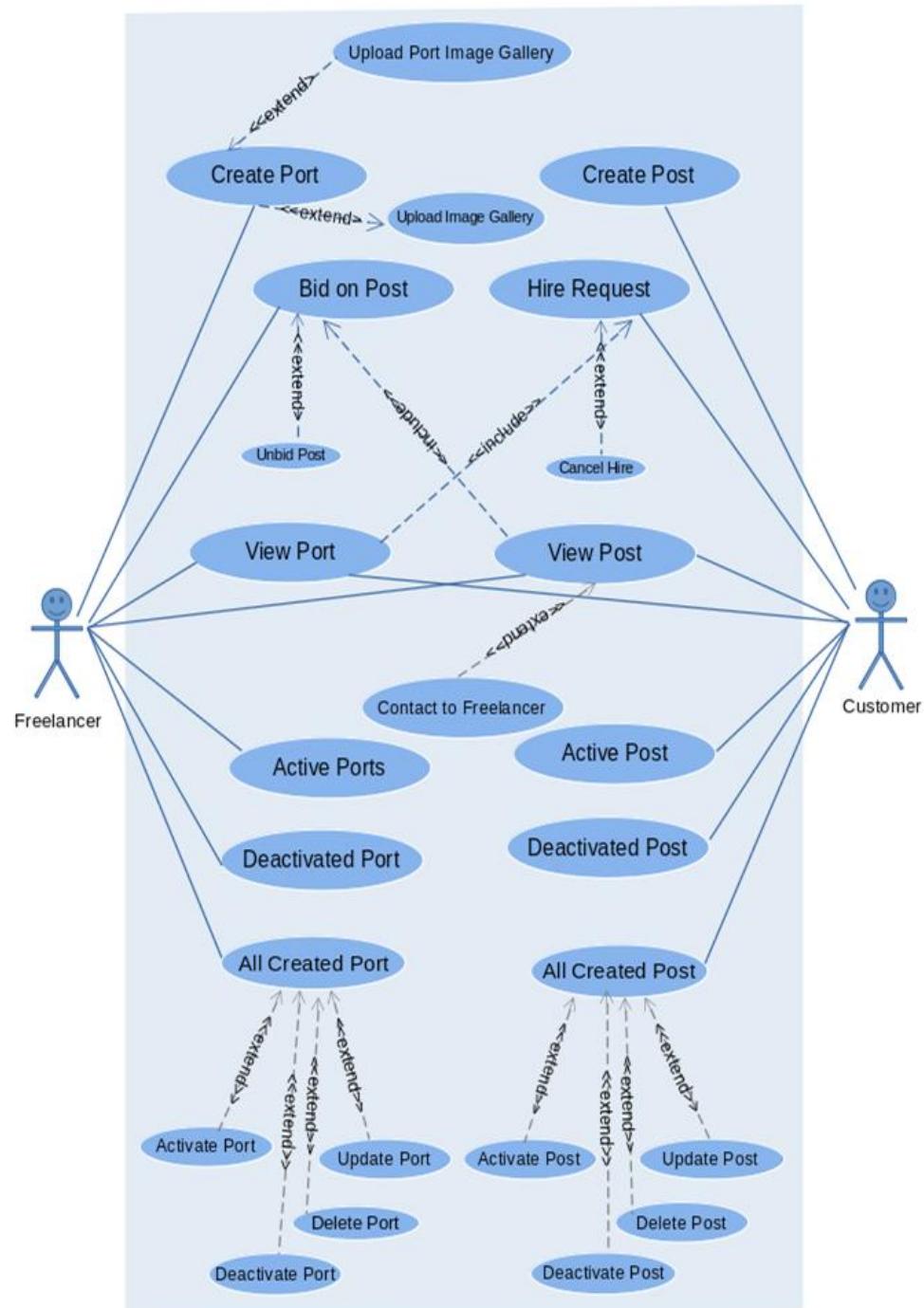


Figure 3.1.2 – Portfolio and Job Post Management System Use Cases.

This is a chat system between two User. This app use Web Socket for communication between two User asynchronously. Web Socket is a computer communications protocol, providing full-duplex communication channels over a single TCP connection. Here Web Socket connects two user port in a channel using Django channel. In Django channel Redis will work as a message broker and cache data store memory.

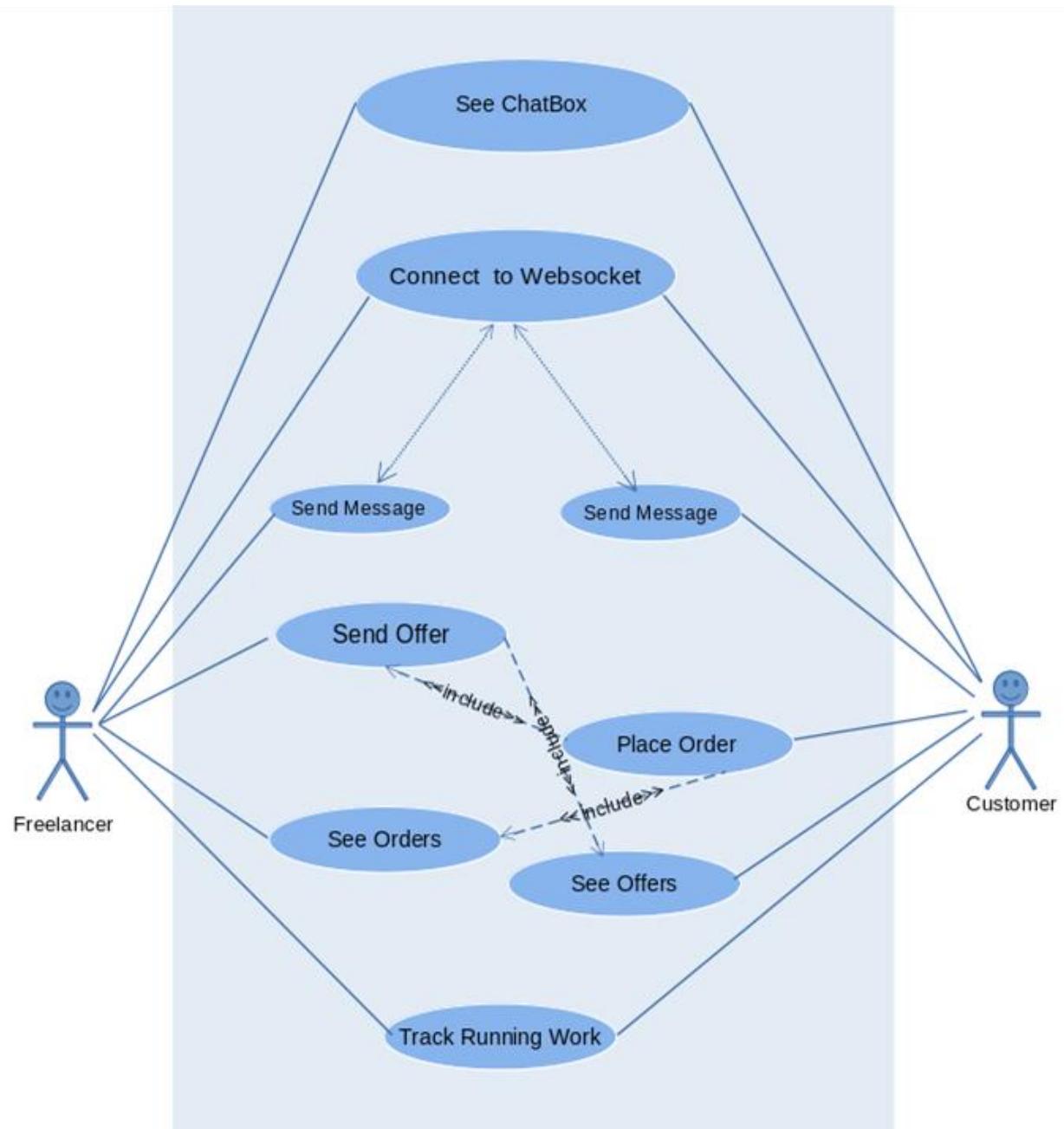


Figure 3.1.3 – Real-time Chatting System Use Cases.

This is our Wallet App which will cover all kinds of transactions in this system, first customer will deposit money to his wallet and then he can place order to freelancer. And the tricky point is every order will be just an order payment confirmation not the actual payment to freelancer.

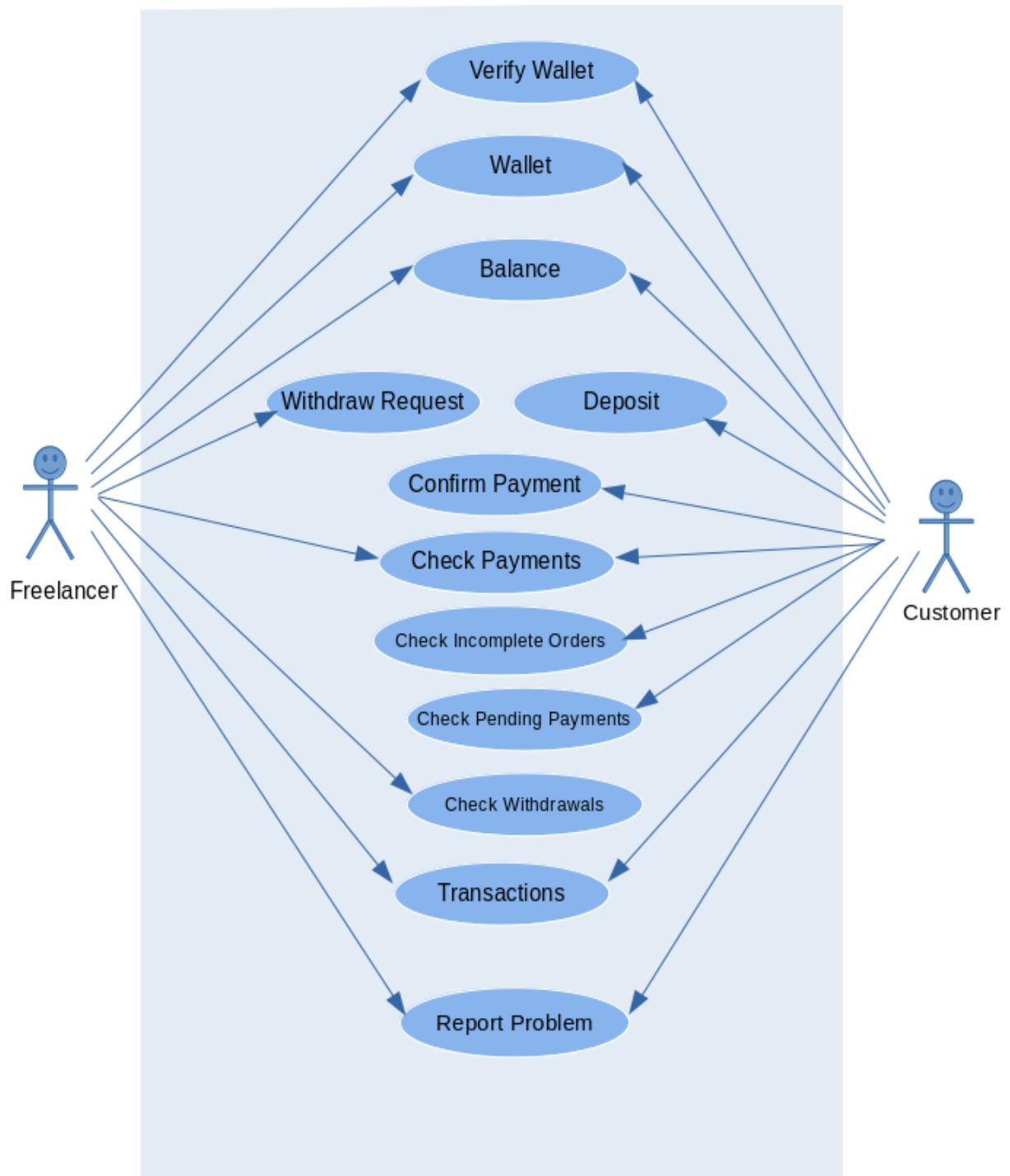


Figure 3.1.4 – Closed Wallet System Use Cases.

This is work app, this diagram shows a full cycle from sending offer to complete work. With lots of feature works will be comfortable and time consuming. An work tracking process will make communication more faster, freelancer can give his work update about done percentage and customer can track his work easily.

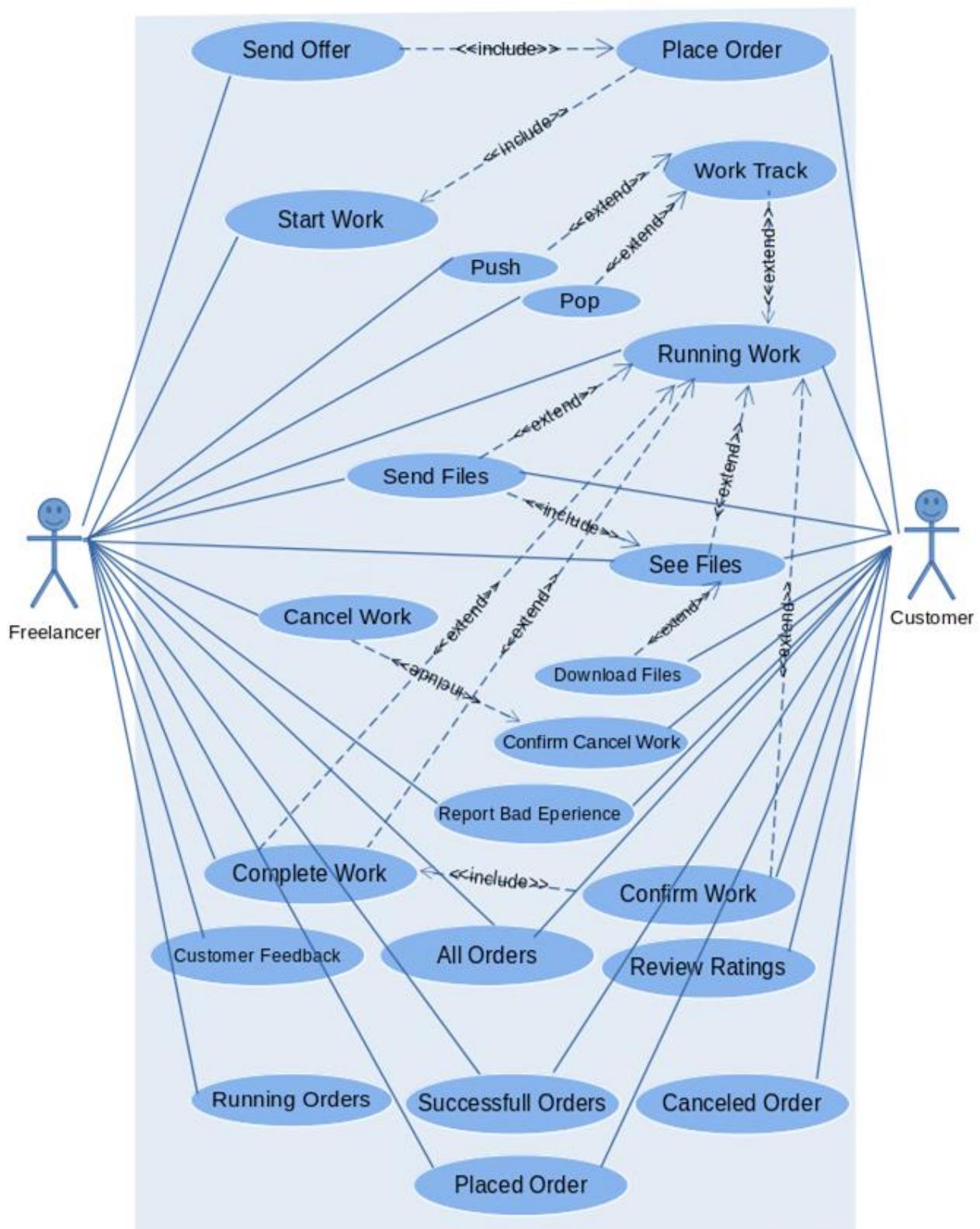


Figure 3.1.5 – Work & File Sharing System Use Cases.

This use case shows search and filtering feature in this application. This system search process different type data for both freelancer and customer, freelancer search will show jobs and customer search will show portfolio of freelancers. Also they can filter by categories.

Admin can analysis about user search history keywords, location, browser etc. And by analysis about keyword the core business can grow up easily.

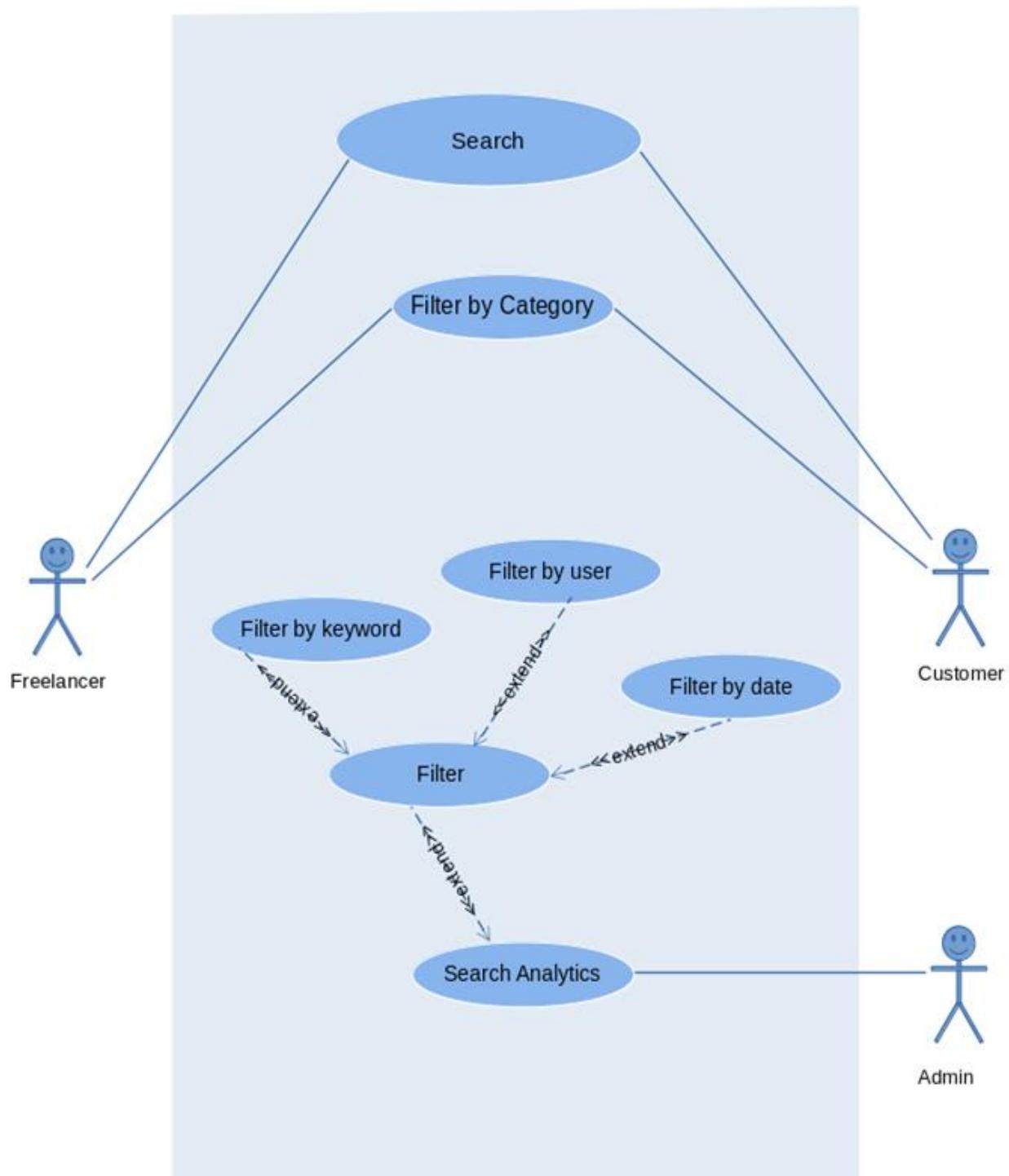


Figure 3.1.6 – Search & Filtering System Use Cases.

Dashboard is amazing workspace for both freelancer and customer. From dashboard user can manage and monitor all kinds of major actions. There are some Graph, Charts for both role. Also, can see his previous history and analytics in this app. This use case shows some major actions with this app.

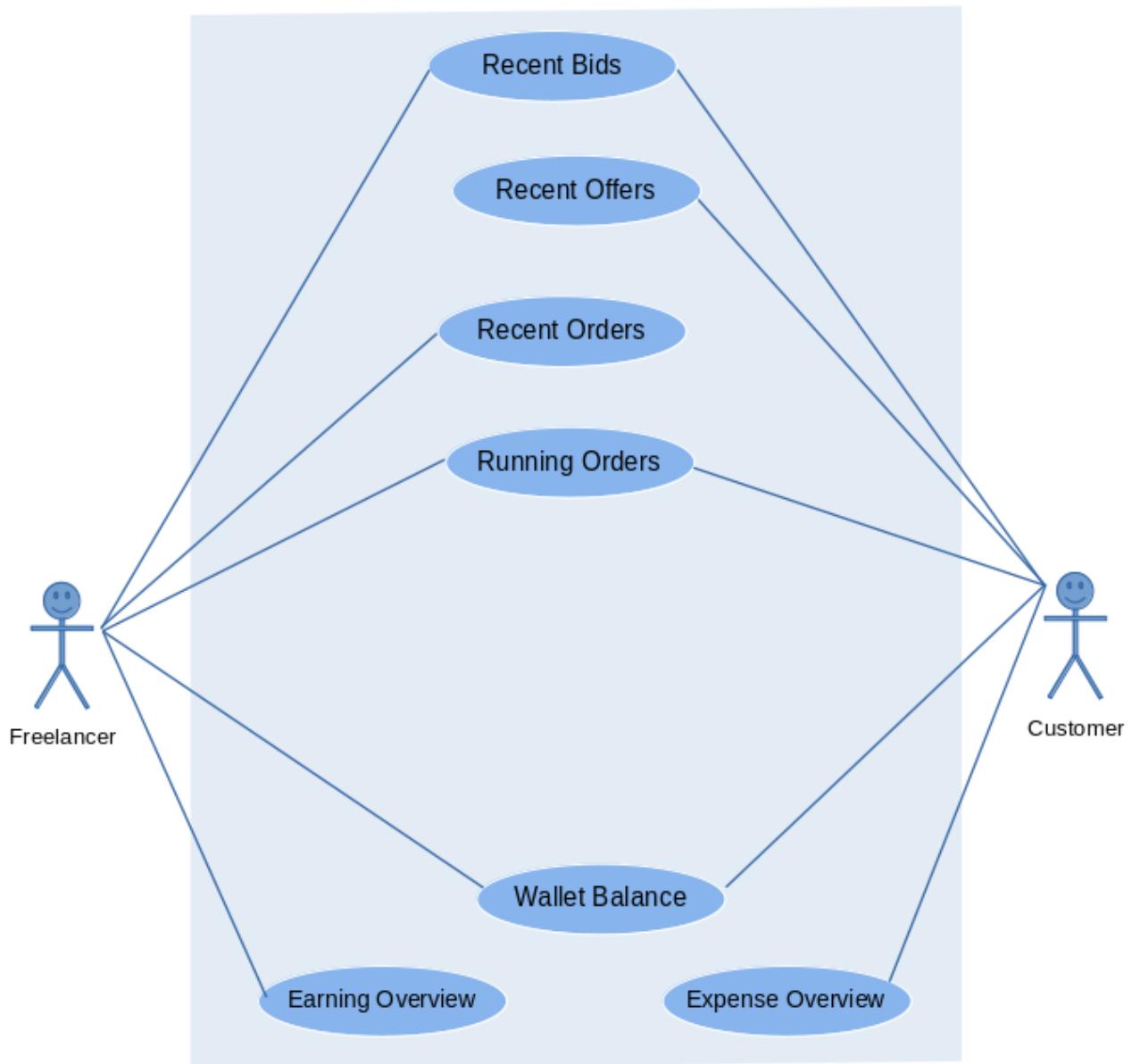


Figure 3.1.7 – Dashboard and Monitoring System Use Cases.

This is notification app which will notify all kinds of action between two user. This helps user about actions to him by other user. This system has lot of back-end functionalities, to make this real time we followed lot of mechanisms. Some of this back-end handled by Django, we just followed their structure.

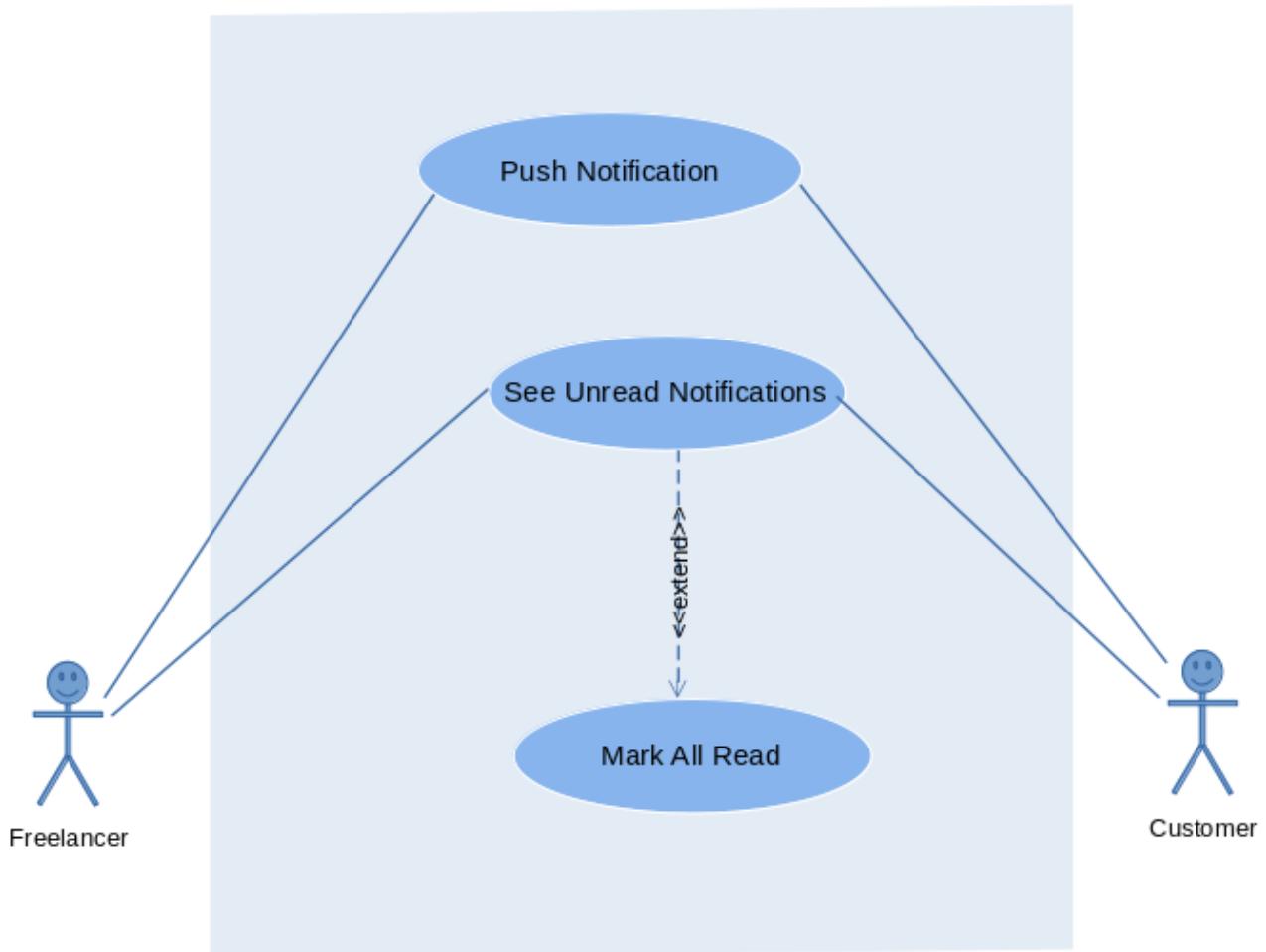


Figure 3.1.8 – Notification System Use Cases

3.2 Entity-Relationship Diagram

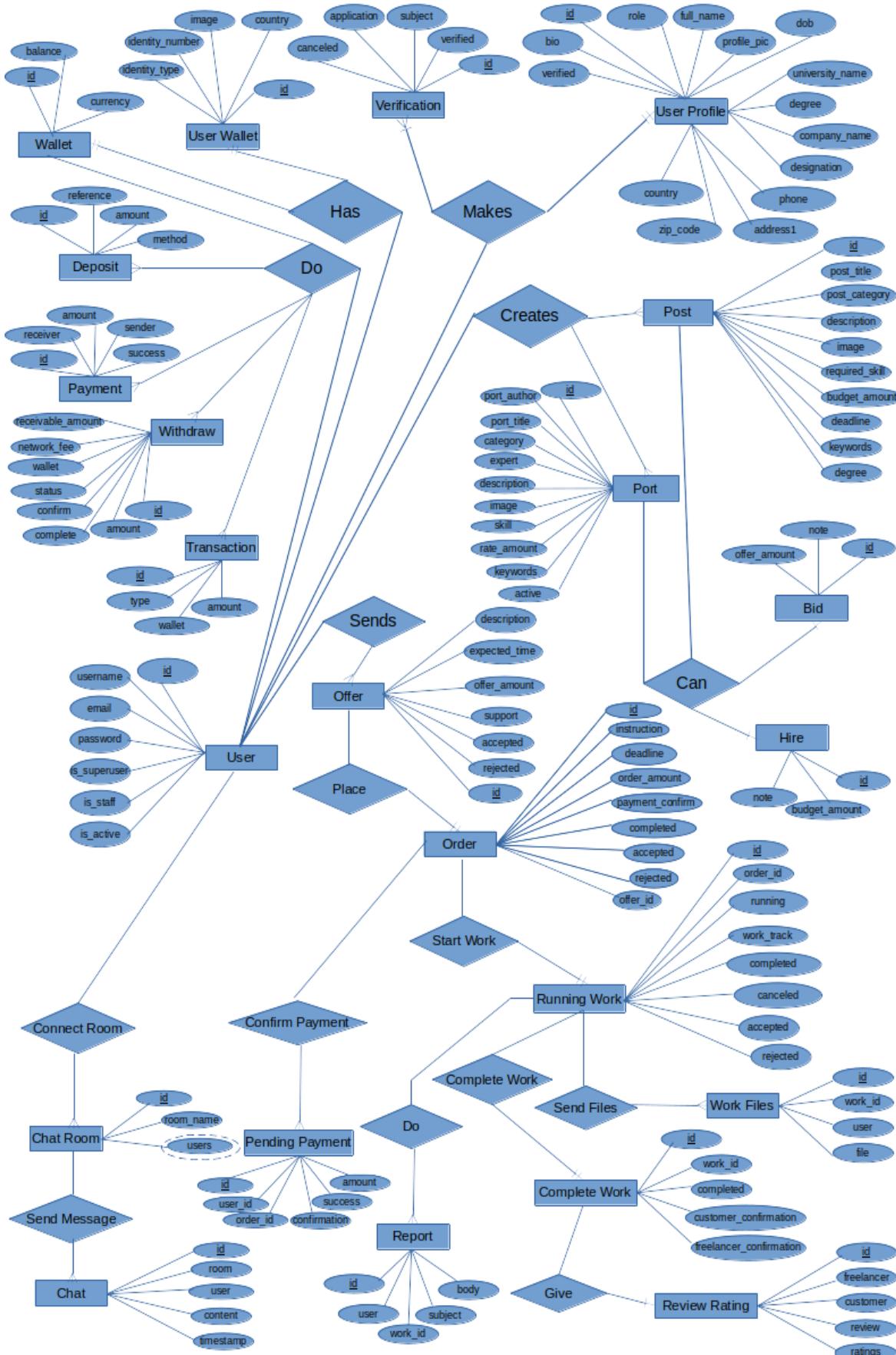


Figure 3.2.1 - Entity Relationship Diagram of the total system.

3.3 Schema Diagram

This database schema tables shows relationship of 12 tables including base entities of this system. Every line's shows relationship with each other tables. From user table the black line relates with other Schema part tables.

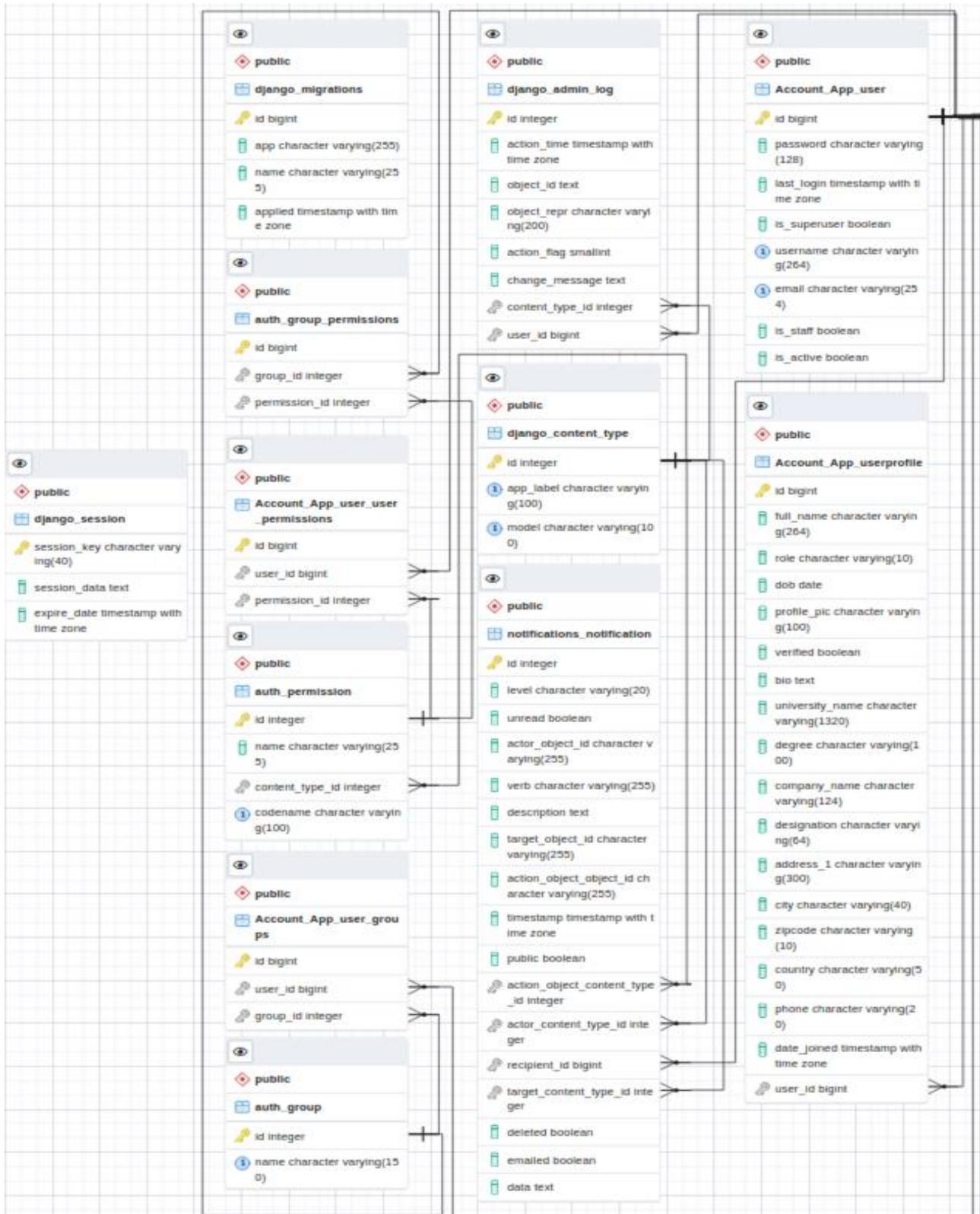


Figure 3.3.1 – Database Schema Part 1

Here showing 12 tables including Verification, Post, Port, Port Image Gallery, Bid, Hire, Chat Room, Chat, Wallet tables exist with business relationship with each other. Black border in top shows tables relation with user tables.

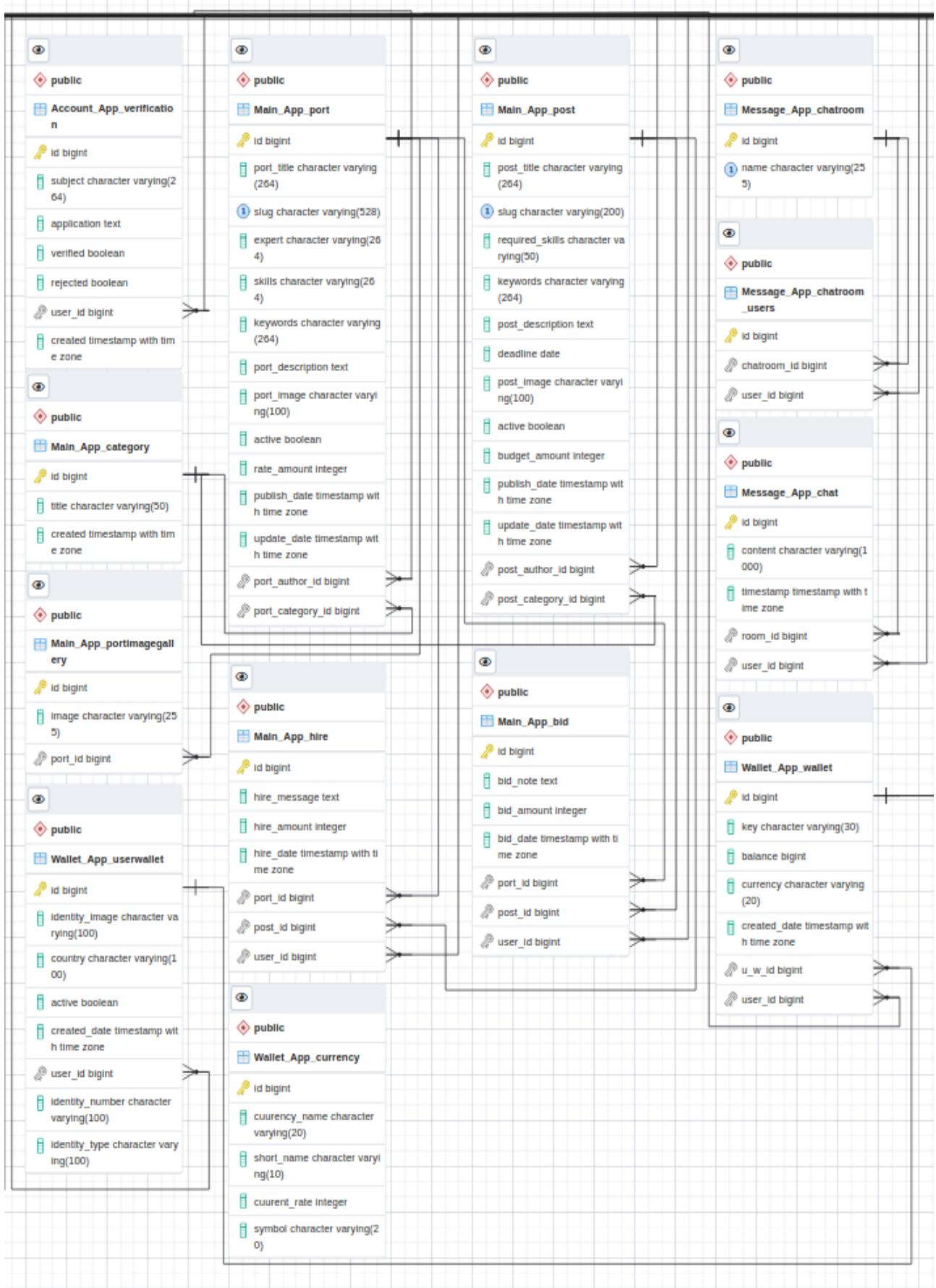


Figure 3.3.2 – Database Schema Part 2

Here, 7 table showing relation with user table in top black border and line with others. Schema of Offer, Order, Deposit, Pending Payment, Running Work, Payment, Report tables.

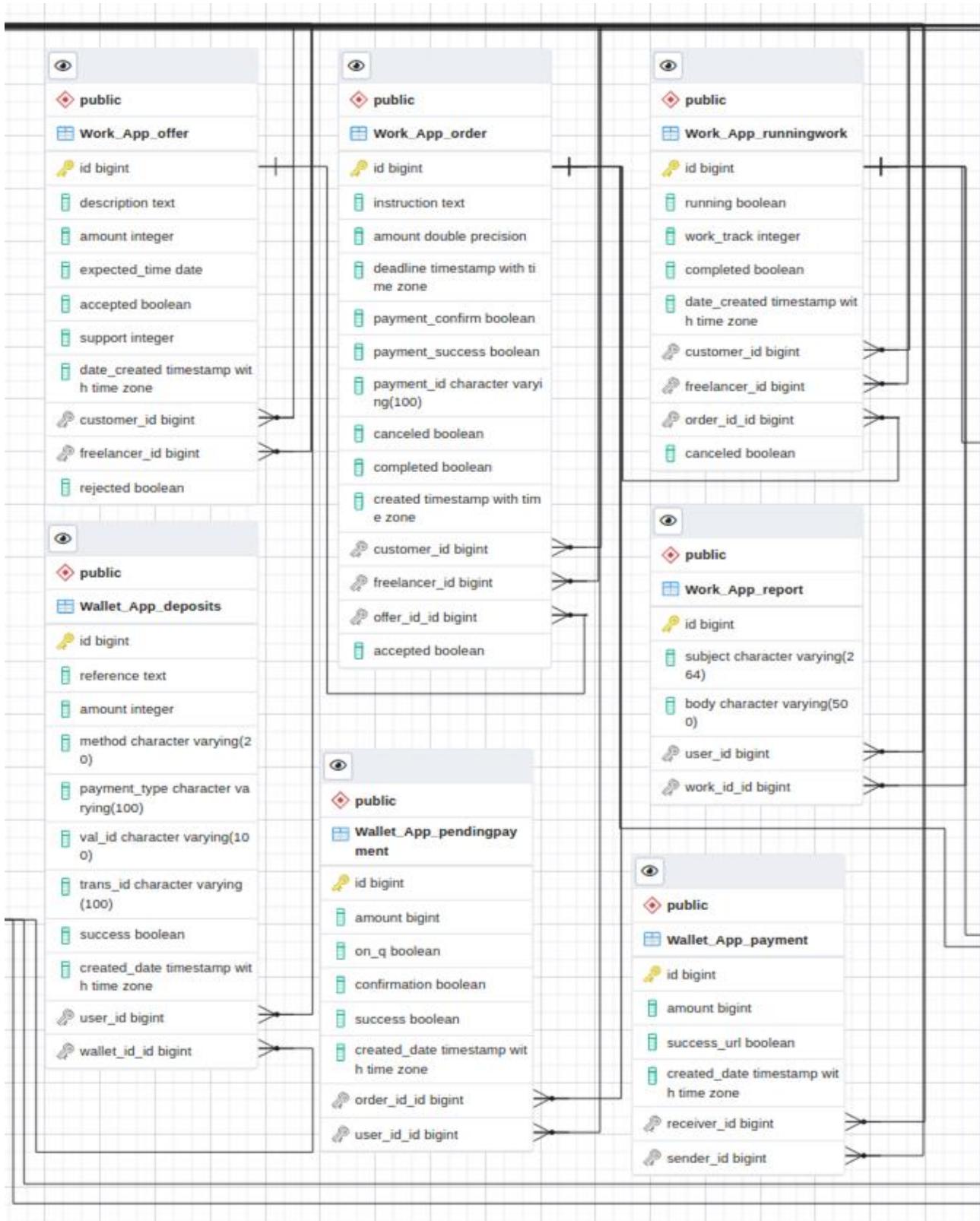


Figure 3.3.3 – Database Schema Part 3

Here showing 7 tables Database Schema, including Running work Files, Review Writing, Search History, Complete Work, Canceled Work, Transaction, Withdraw Request.

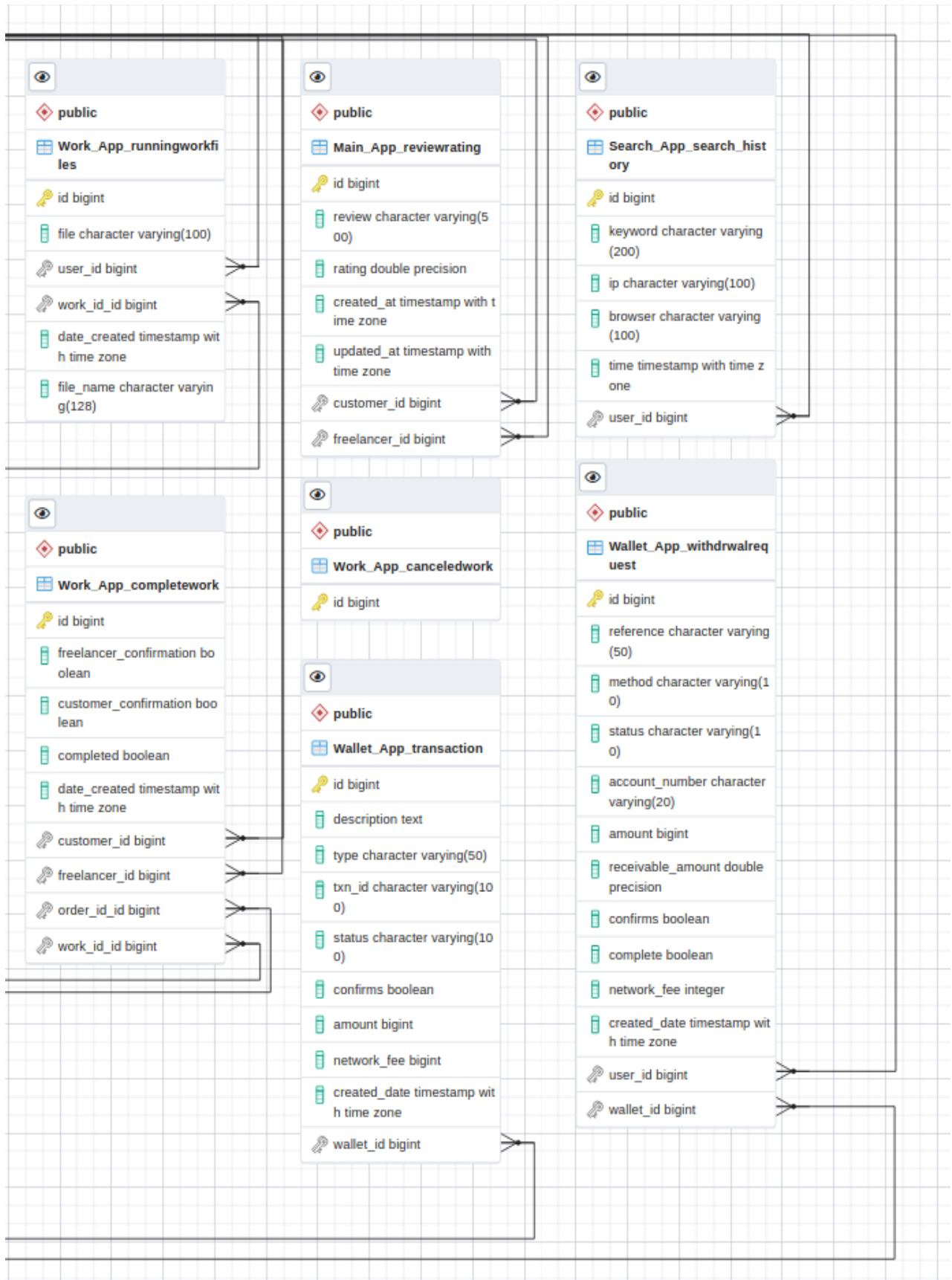


Figure 3.3.4 – Database Schema Part 4

3.4 Data Flow Diagram

Here this DFD shows 5 level complete flow from Account Creation to complete transaction & money withdraw.

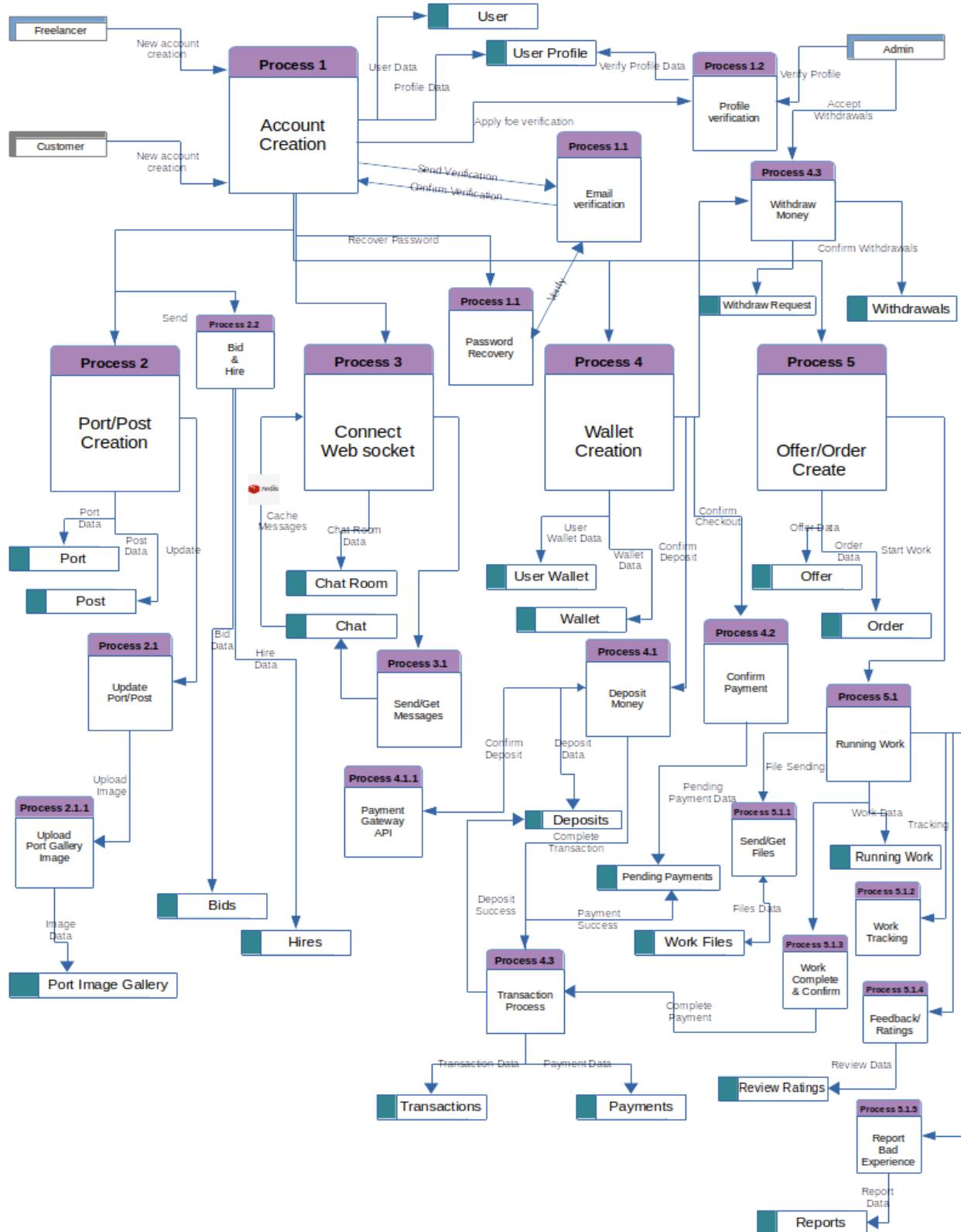


Figure 3.4.1 – 5 Level Data Flows Diagram

4 Implementation

4.1 Configuration

In this application we will use Django 4.0 as server side framework and PostgreSQL as database. So first ensure that python 3.8 and Django and PostgreSQL and a browser in your device.

Open terminal and open VS Code or other code editor.

Write those commands in terminal to start this project -

```
mkdir Project Folder  
django-admin startproject FSMS  
python -m venv env  
pip install django  
pip install pillow
```

Put full project in this directory

```
pip freeze > requirements.txt  
pip install -r requirements.txt
```

and install others module if necessary.

Alright, next start the virtual environment from terminal as our development environment.

```
source env/bin/activate
```

Open PGAdmin Make a database in PostgreSQL name ‘FSMS’ or you can configure other name from settings.

Lets make database migration for PostgreSQL

```
python manage.py makemigrations  
python manage.py migrate  
python manage.py runserver
```

Congratulations your server will be running!

Open Chrome/Firefox/Safari or any other safe browser to open running server.

You can find your running server in your local server, type this url in your browser address bar
<http://127.0.0.1:8000/> and you will see the homepage of this system.

Note: There may be dependencies on python packages, please install those packages in virtual environment.

4.2 Interface

The screenshot displays the amarKaz.io landing page, which is designed for digital freelancers and customers. The top section features a banner with a photograph of multiple laptops and a central text area: "Independent marketplace for digital work, for digital people." Below this are two buttons: "Join As Freelancer" and "Join As Customer". A large green button at the bottom of the banner says "POST YOUR PROJECT".

Process & Features

- Complete your profile and apply for verification, get a verified profile.
- Verify your wallet and deposit money from popular payments methods.
- Post your job and contact with interested freelancer by their bid.
- Make port and get order request from customer and bid on your interested job.
- Realtime chat using a message app, message each other asynchronously.
- Discuss and make a trust and deal, send offer and place order and confirm payment.
- Track on your work, and countdown with a realtime deadline counter.
- Complete work and confirm work, payment will be handled automatically.
- Monitor your all works and business flow from a analytical dashboard.
- Withdraw money from your selected bank or payment method.

Marketplace & Services

Programming Tech	Digital Marketing	Design & Arts	Others
<ul style="list-style-type: none"> Web Design Web Development App Development Software Development 	<ul style="list-style-type: none"> Search Engine Optimization Social Media Marketing Email Marketing Content Marketing 	<ul style="list-style-type: none"> UI & UX Design Graphics Design Video & Animation 	<ul style="list-style-type: none"> Content Writing Data Entry IT Consultancy

Top Ports

This section shows a grid of 10 service listings, each with a thumbnail, provider name, service description, price, and "Save" and "View" buttons.

- ReactJS for Web Development** by **tanmarmahend**: I will develop your reactjs web app as react js developer... ₹ 200.00
- Content Writer** by **meraz**: I will be your blog content, web content, and SEO writer... ₹ 200.00
- DEVELOP Android & iOS Flutter** by **shashankr007**: I will build a mobile app using flutter for both ios and android... ₹ 200.00
- odoo** by **rahulshan**: I will install, configure and make custom odoo modules... ₹ 300.00
- Online Developments** by **subrahm100**: I will design awesome website or landing page... ₹ 150.00
- GROW YOUR BRAND ON SOCIAL MEDIA** by **shantibd**: I will do digital marketing management and social media mark... ₹ 150.00
- MongoDB Express ReactJS NodeJS** by **mizanrulhan**: I will develop mern stack web application... ₹ 200.00
- Premium Shopify Website** by **sumonmujumdar**: I will create your premium shopify ecommerce store website... ₹ 200.00
- Business Logo & Branding** by **alaminbappy**: I will design your business logo and branding minimalist uni... ₹ 150.00
- Web Design** by **shafin**: I will do web design in figma, psd website design, or UI UX ... ₹ 150.00

Our Clients

This section shows a row of six placeholder boxes for client logos.

Services

- Solution
- Credit Card
- Mobile App

Partners

- Integration Partner
- Invoice Generator
- Money Withdraw

About

- About amarKaz.io
- Fees
- Investors
- Careers

Help

- Customer Care
- Contact Us
- Mail Us

Follow Us

- Facebook
- Twitter
- LinkedIn

All rights reserved by amarKaz | © Copyright 2021

Figure 4.2.1 – Interface for public user as landing page

Create New Account

Role
 Customer Freelancer

Username
Email
Password
Confirm Password

I'm not a robot 
reCAPTCHA
Privacy - Terms

[Signup](#)

Already have an account? [Signin](#)

Figure 4.2.2 – Interface for user creation with a role.

Signin Your Account

Email
Password

[Signin](#)

Don't have an account? [Signup](#)
Forgot your password? [Recover](#)

Figure 4.2.3 – Interface for user signin.

The screenshot shows the amarKaz.io customer homepage. At the top, there's a navigation bar with the logo 'amarKaz.io' and icons for home, notifications, messages, and account. A search bar and user profile 'tarikulsujon' are also present. Below the header is a large, colorful illustration of two people working at desks with laptops, surrounded by clouds and digital icons.

All Categories **Filter**

React JS for Web Development by tanvirmahmud (₹ 200.00) - I will develop your reactjs web app as reactjs developer... **Save** **View**

Content Writer by meraz (₹ 200.00) - I will be your blog content, web content, and SEO writer... **Save** **View**

DEVELOP Android & iOS by shatilsarower (₹ 200.00) - I will build a mobile app using flutter for both ios and and... **Save** **View**

Flutter Best Performance and Scalability by raihanalam (₹ 300.00) - I will install, configure and make custom odoo modules... **Save** **View**

Odoo Developments by ashrafulislam (₹ 150.00) - I will design awesome website or landing page... **Save** **View**

Previous 1 2 Next

Figure 4.2.4 – Interface for customer homepage showing portfolio.

The screenshot shows the amarKaz.io freelancer home interface. The top navigation bar includes the logo 'amarKaz.io', a search bar, and user profile 'tanvirmahmud'. A sidebar on the left shows 'freelancer : tanvirmahmud' and filtering options for 'All Categories' and 'Filter'.

Build me a C# ASP .net Chatting web site using SQL server as backend

⚡ Web Developer, ASP .NET

Deadline: Jan. 31, 2023

Budget: ₹ 15000 **Bid**

⌚ tarikulsujon ⚓ Nov. 19, 2022, 5:07 p.m.

Data Entry Operator work is available.

⚡ Data Entry Operator

Deadline: Jan. 25, 2023

Budget: ₹ 10000 **Bid**

⌚ zahinoly ⚓ Nov. 19, 2022, 5:01 p.m.

Figure 4.2.5 – Interface for freelancer home showing job posts.

Profile

Add Picture

Verified X

Role	freelancer
Username	sentukhan

[Change Profile Info](#)

[Change Personal Info](#)

Your profile is not set. [Complete Profile Here](#)

Fullname

Email sentukhan@gmail.com

Bio

Company

Designation

University

Degree

Phone

Address

Zip Code

City

Country

Figure 4.2.6 – Interface for user profile and profile management.

Profile Information

Full Name

About yourself...

Address

Zipcode

City

Country*

Date of birth*

mm / dd / yyyy

Update

Figure 4.2.7 – Interface for profile information update.

The screenshot shows the 'Change Profile Picture' section of the amarKaz.io application. At the top, there's a purple header bar with the amarKaz.io logo, a search bar, and a user profile icon for 'sentukhan'. Below the header, the main content area has a light gray background. It features a form for changing a profile picture. The first field is labeled 'Profile pic' with a placeholder 'Browse... No file selected.' A green 'Change Picture' button is located below this field.

Figure 4.2.8 – Interface for user profile picture add/change.

The screenshot shows the 'Personal Info' section of the amarKaz.io application. The interface is similar to Figure 4.2.8, with a purple header bar and a light gray content area. Under 'Personal Info', there are two input fields: one containing 'tarikulsujon' and another containing 'tarikulsujon@gmail.com'. A red 'Submit Change' button is at the bottom.

Figure 4.2.9 – Interface for user personal information change.

The screenshot shows the 'Change Old Password' form of the amarKaz.io application. The header and basic layout are consistent with the previous figures. The form includes fields for 'Old password*', 'New password*', and 'New password confirmation*'. Below these fields is a list of validation rules: 'Your password can't be too similar to your other personal information.', 'Your password must contain at least 8 characters.', 'Your password can't be a commonly used password.', and 'Your password can't be entirely numeric.'. A red 'Change Password' button is at the bottom.

Figure 4.2.10 – Interface for old password change form for logged user.

The screenshot shows the Amar-wallet interface. At the top, there is a purple header bar with the logo "amarKaz.io" and several icons (home, notifications, messages, etc.). A search bar is located on the right side of the header. Below the header, the main content area has a light gray background. On the left, there is a sidebar with a green button labeled "Withdraw Money". The main body contains several sections: a yellow banner at the top stating "Your wallet is not active please complete wallet form.", a message box on the right saying "Your wallet isn't active you can't interact with those options, please fill the form properly and activate your wallet.", another message box below it saying "Your profile isn't verified, please verify profile first.", and a large form on the right for activating the wallet. The form includes fields for "Identity type*" (set to "National Identity"), "Identity Number", "Upload Image File*" (with a "Browse..." button), "Country*", and a green "Activate Wallet" button.

Figure 4.2.11 – Interface for inactive wallet with activation form.

This screenshot shows the Amar-wallet interface after the wallet has been activated. The top header and sidebar are identical to Figure 4.2.11. The main content area now displays a different set of options. On the left, there is a sidebar with a green button labeled "Deposit Money". The main body contains several sections: a yellow banner at the top stating "Your wallet is not active please complete wallet form.", a message box on the right saying "Your wallet isn't active you can't interact with those options, please fill the form properly and activate your wallet.", another message box below it saying "Your profile isn't verified, please verify profile first.", and a large form on the right for activating the wallet. The form includes fields for "Identity type*" (set to "National Identity"), "Identity Number", "Upload Image File*" (with a "Browse..." button), "Country*", and a green "Activate Wallet" button.

Figure 4.2.12 – Interface for activate wallet, options for only verified customer.

The screenshot shows a web-based application interface for profile verification. At the top, there is a purple header bar with the logo 'amarKaz.io' and several icons (home, notifications, messages, etc.). A search bar is located on the right side of the header. Below the header, the main content area has a title 'Profile Verification'. A message box displays the text 'Your application is pending, please wait for while.' In the center, there is a form field labeled 'Subject' containing the text 'Application for profile verification'. Below it is another field labeled 'Application...' which is currently empty. At the bottom of the form is a green 'Apply' button.

Figure 4.2.13 – Interface for user profile verification application.

This screenshot shows the same application interface after the profile has been verified. The purple header bar and search bar are visible at the top. The main content area features a title 'Profile Verification'. A message box at the top says 'You got verified 🎉 Congratulations.' Below this, a green message box says 'Your application is verified.' The central form field now contains the text 'Subject: Application for profile verification' and the 'Application...' field is also empty. The 'Apply' button remains at the bottom.

Figure 4.2.14 – Interface for verified application.

This screenshot shows the profile validation interface. The purple header bar and search bar are present. The main content area has a title 'Profile'. On the left, there is a user profile section with a placeholder for a profile picture, a 'Change Picture' button, and a 'Verified' status with a green checkmark icon. To the right, there is a table displaying various profile details:

Fullscreen	Shatil Sarwoer
Email	shatilsarower@gmail.com
Bio	I am professional App Developer working with Flutter. With a versatile knowledge of cross platform app development. Working as freelance App developer in the market.
Company	Amarkaz
Designation	App Developer
University	Daffodil International University
Degree	BSC in CSE
Phone	01932432433
Address	Malibagh
Zip Code	1212
City	Dhaka
Country	BD

Figure 4.2.15 – Interface for validate profile with verified icon.

Deposit > Checkout

Reference*

Order for

Amount*

10000

Method*

SSLCommerz

Save Deposit Form

Reference:
Order for
Method:
sslcommerz

Deposit Total = ₢ 10000.00

Confirm Deposit

Figure 4.2.16 – Interface for money deposit checkout for customer.

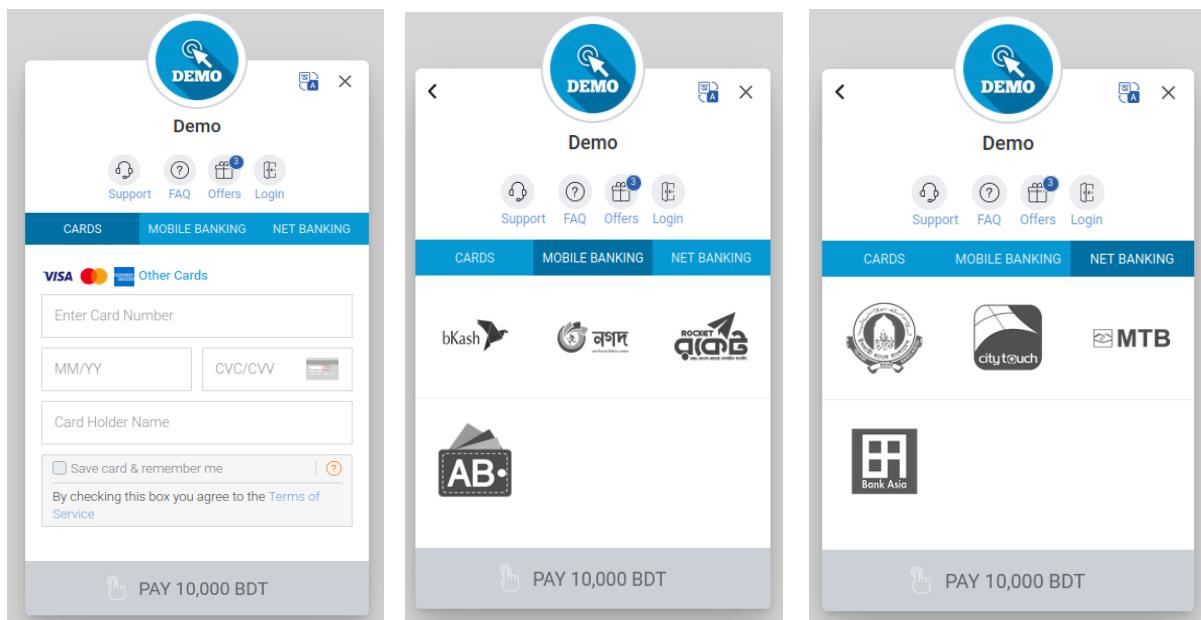


Figure 4.2.17 – Interface for payment gateway API, showing SSLCommerz API.

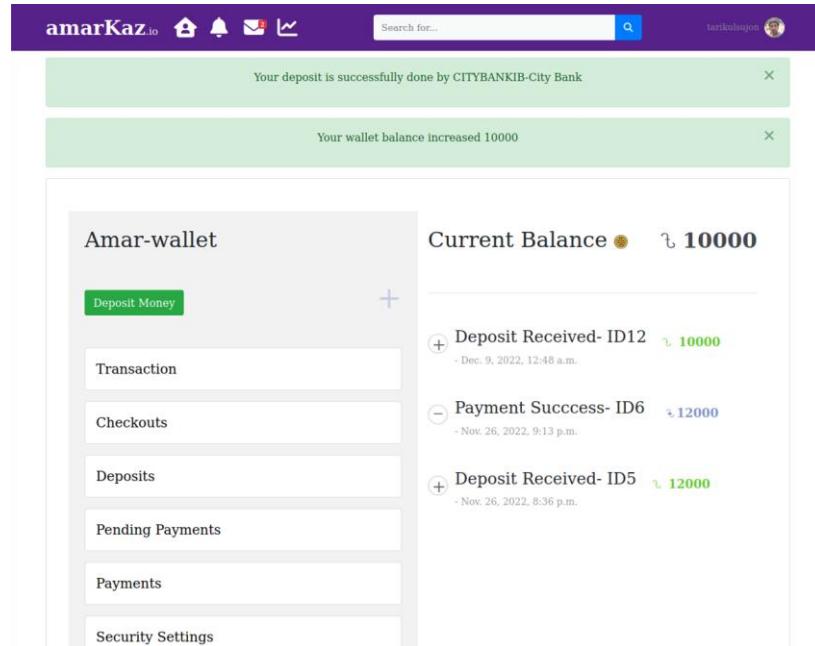


Figure 4.2.18 – Interface after deposit with message and transaction data in wallet for customer.

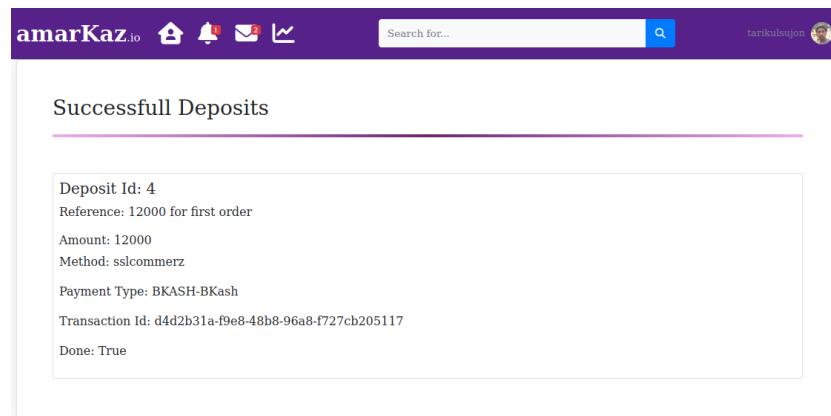


Figure 4.2.19 – Interface for all successful deposits for customer.

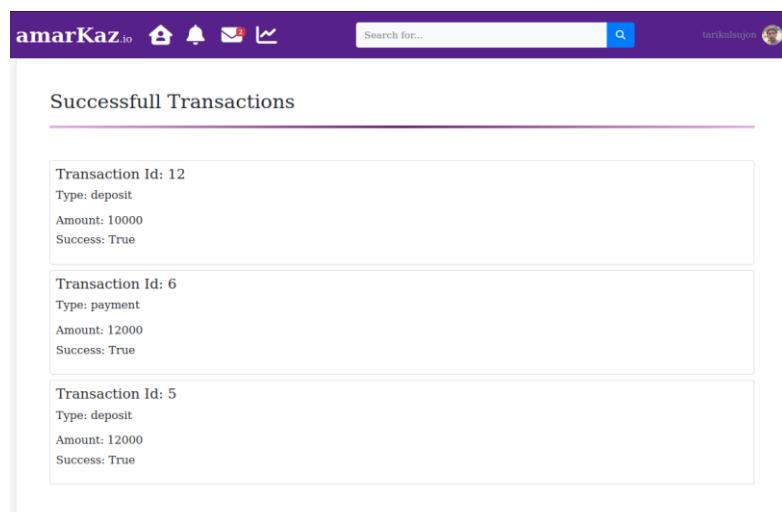


Figure 4.2.20 – Interface for all kinds of transaction for both user.

amarKaz.io    

Create Port

Title

Expertise on

Skills

Styles Format **B** *I* U **S** ← → ∞ ∞ | | | | | | A A Ω Ω Source

Browse... No file selected.

₹ Per hour ↑ ↓

Keywords

[Create Port](#)

The header features the amarKaz logo with ".io" in smaller font below it. To the right are four icons: a house, a bell, an envelope with a red '1' notification, and a document. A search bar with placeholder text "Search for..." and a magnifying glass icon is positioned next to the icons. On the far right, the user's name "tanvirmahmud" is displayed with a small profile picture.

All Ports

I will develop your reactjs web app as react js developer

Active: True

Hourly Rate: 200

Hire Requests: 0

Published: Nov. 26, 2022, 5:30 p.m. | Updated On: Nov. 26, 2022, 5:30 p.m.

[View Port](#)

[Edit Port](#)

[Deactivate](#)

[Delete](#)

Figure 4.2.22 – Interface for all portfolio manage for freelancer.

Edit Port

Web Development

I will develop your reactjs web app as react js developer

Web Development

HTML, CSS, Bootstrap, Javascript, React, Website, App

Styles - | Format - | **B** *I* **U** **S** ← → | ☰ ☱ | ☰ ☱ ☲ ☳ | A- A+ | ☺ ☻ | ☷ ☸ | ☹ ☺ Source

Hello,
I am experienced React.js Developer. If you are looking for React developer that fits in your budget then your search is over. I'll help you with your Web Application front end using React.js.
I can also help you to create React UI using Html, CSS or React UI library (Reactstrap)

charges are not fixed - It can change according to the complexity of the task or number of pages.

My skills:

Reactjs
Hooks
Reactstrap
Redux

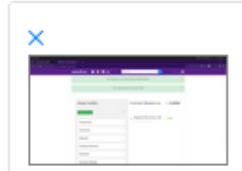
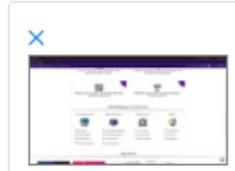
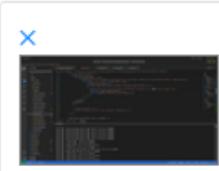
Hero Image*

Browse... No file selected.

200

HTML, CSS, Bootstrap, Javascript, React, Website, App

Update



Browse... No file selected.

Upload Image

Figure 4.2.23 – Interface for portfolio edit and multi image upload for freelancer.

amarKaz.io    

Search for... 

tarikulsujon 

I will install, configure and make custom odoo modules

Hourly Rate: ₦ 300



About Freelancer

 **rathanalam** 
Odoo Developer
[Contact this Freelancer](#)

Select your post:
Build me a C# ASP .net Chatting

Write about your work...*

Budget amount...*

[Send Hire Request](#)

About This Port

Hi this Raihan,

I am a Professional Odoo Developer with extensive knowledge of designing and developing odoo applications. I am capable of making odoo app from scratch or customize the existing module. I am currently working as a odoo developer in a odoo certified company. Here are some of the critical skills I love to share,

- Odoo version (12,13,14,15) community version with Customization
- Expertise in Purchase, Sales, Inventory and Accounting Module
- Customized and efficient helpdesk module development
- Different approval management Systems like Sales orders, Purchase Order, Invoices.
- Automation things like Recurring invoice and others
- Elegant form designing and backend development
- Clean code and testing release
- Customizable Report templates with data representation
- Modern wizard and report Generate on Pdf and excel sheet
- Database design and excellent query optimization in case of extensive data

Besides odoo development, i have also skilled in deployment in odoo.

- Microsoft Azure VM deployment service with Odoo
- AWS service
- Installed and set up locally
- Configure and set up the server in a linux based system
- Real IP configuration and live.

Customer Reviews

 2 reviews

zahinoly  5.0	Dec. 9, 2022, 5:46 p.m.
Excellent Work, Good behavior. I am satisfied.	
tarikulsujon  5.0	Dec. 12, 2022, 10:39 p.m.
Nice work, fast delivery. Great.	

Figure 4.2.24 – Interface for portfolio view with hire request and customer reviews for customer.

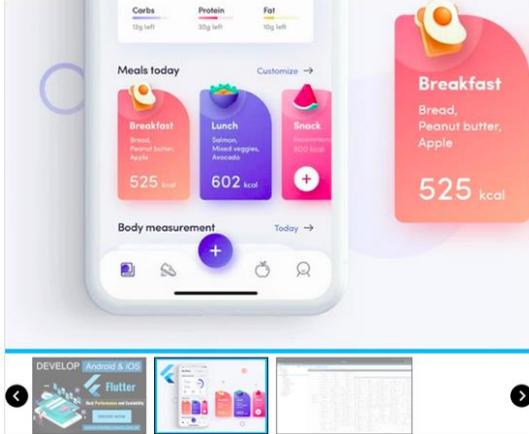
amarKaz.io    

Search for... 

zahinoly 

I will build a mobile app using flutter for both ios and android

Hourly Rate: ₴ 200



About Freelancer

 **shatilsarower** 
App Development
[Contact to Freelancer](#)

Select your post:
 

Write about your work...*

Budget amount...*

[Send Hire Request](#)

About This Port

Are you looking for a Flutter app for your business, product, service, or brand? Then you are at the right place - here you will get a great Flutter app with Amazing Design & Great Performance.

Figure 4.2.25 – Interface for portfolio gallery image zoom on mouse hover on image.

amarKaz.io    

Search for... 

zahinoly 

Profile Information

 raihanalam	Worked 11	Ratings 5.0	Response 25minutes
---	---------------------	-----------------------	------------------------------

Verified 
Country BD

We are team of Professional Programmers with expertise in Website, Mobile App and Desktop App Development. We have experts for almost all the modern Development tools and languages.


raihanalam
I will do django, python, react js websites as a full stack ...
₴ 300.00
[Save](#) [View](#)


raihanalam
I will install, configure and make custom odoo modules...
₴ 300.00
[Save](#) [View](#)

Figure 4.2.26 – Interface for profile view of freelancer for customer.

The screenshot shows the 'Create Post' form on the amarKaz.io platform. At the top, there's a purple header bar with the website name 'amarKaz.io' and various icons. A search bar and a user profile for 'tarikulsujon' are also present. The main form area has a light gray background.

Category*: A dropdown menu with a placeholder 'Please select an item in the list.' and options like 'Job Title' and 'Skill Required'.

Description*: A rich text editor with a toolbar at the top containing buttons for styles, formats, and other document functions. Below the toolbar is a large text input area.

Upload Image File*: A file upload field with a 'Browse...' button and a message 'No file selected.'

Budget: A text input field with a placeholder '₹'.

Keywords: A text input field.

Deadline*: A date input field with a placeholder 'mm/dd/yyyy'.

Action Buttons: A green button labeled 'Post Work' at the bottom right of the form.

Figure 4.2.27 - Interface for job posts creation for customer

The screenshot shows a list of job posts under the heading 'All Post's'. Each post is displayed in a card-like format.

Post 1: Build me a C# ASP .net Chatting web site using SQL server as backend
 Active: True
 Offer: 15000
 Bids: 0
 Published: Nov. 19, 2022, 5:07 p.m. | Updated On: Nov. 19, 2022, 5:07 p.m.
 Buttons: View Post, Edit Post, Deactivate, Delete

Post 2: Need a web developer to make a website for my restaurant.
 Active: True
 Offer: 10000
 Bids: 0
 Published: Nov. 19, 2022, 2:49 p.m. | Updated On: Nov. 19, 2022, 2:49 p.m.
 Buttons: View Post, Edit Post, Deactivate, Delete

Figure 4.2.28 - Interface for own job posts manage buttons for customer.

Edit Post

Web Development

Build me a C# ASP .net Chatting web site using SQL server as backend

Web Developer, ASP .NET

Styles | Format | **B** *I* U ~~S~~ ← → | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | <img alt="italic icon

amarKaz.io    



rahanalam 

Need a web developer to make a website for my restaurant.

Budget: ₹ 10000
Deadline: Feb. 20, 2023

⚡ Web Developer & Web Designer
🕒 Nov. 19, 2022, 2:49 p.m.

Hello,
I have a restaurant and I want to develop a website for my website with some online order feature. People can place order and I will take payment and order from customer.
Interested freelancer please start bidding, I will contact asap.

[Attachments](#)

Bids:

 [tanjirmahmud](#)
\$15000
Hello, I am interested in your project.
Nov. 26, 2022, 8:06 p.m.

About Client



tarikulsujon 

I will do django, python, react
Send note...*

Bid amount...*

[Bid This Job](#)

Figure 4.2.30 - Interface for job posts details with bidding features for freelancers

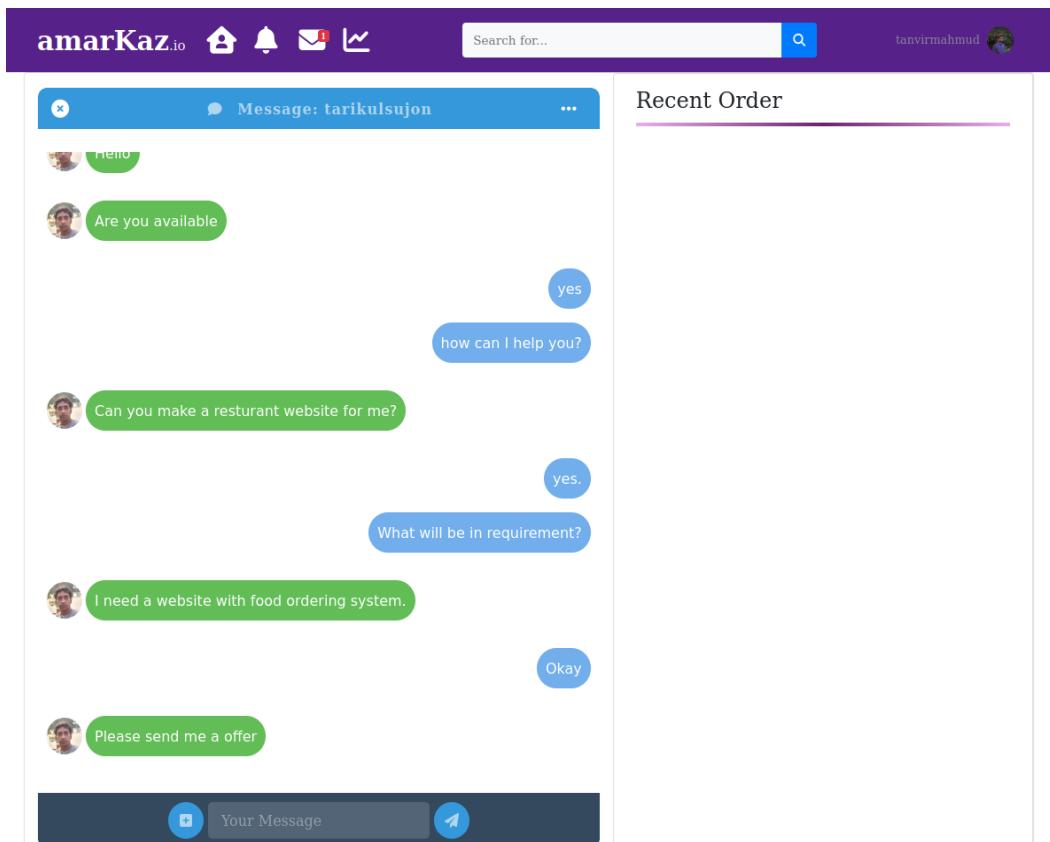


Figure 4.2.31 - Interface for chatting between two user.

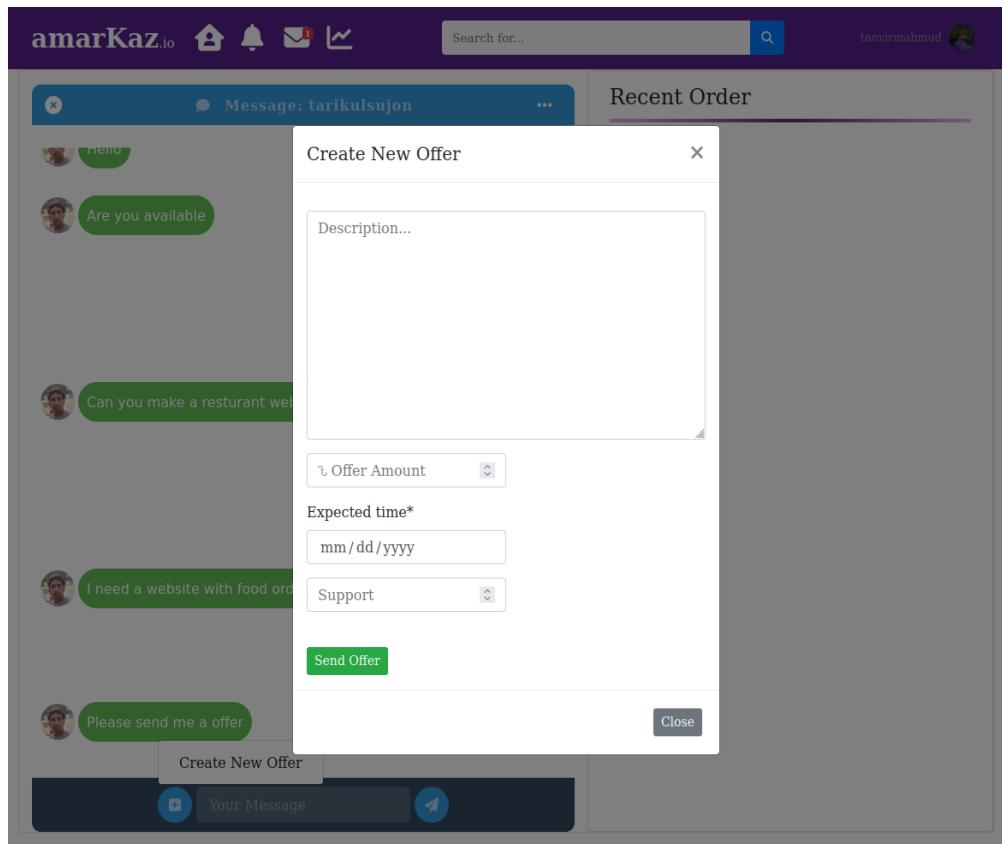


Figure 4.2.32 - Interface for offering a service to customer in chat interface for freelancer.

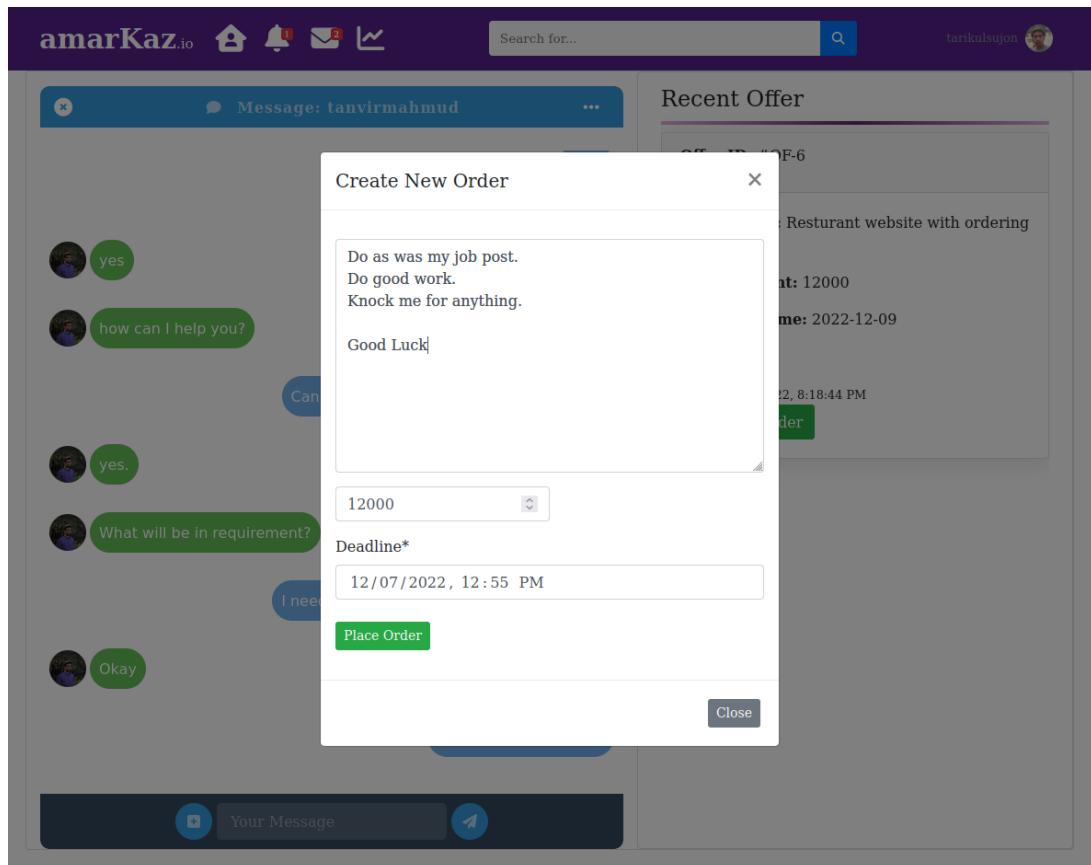


Figure 4.2.33 - Interface for ordering a service to freelancer in chat interface for customer.

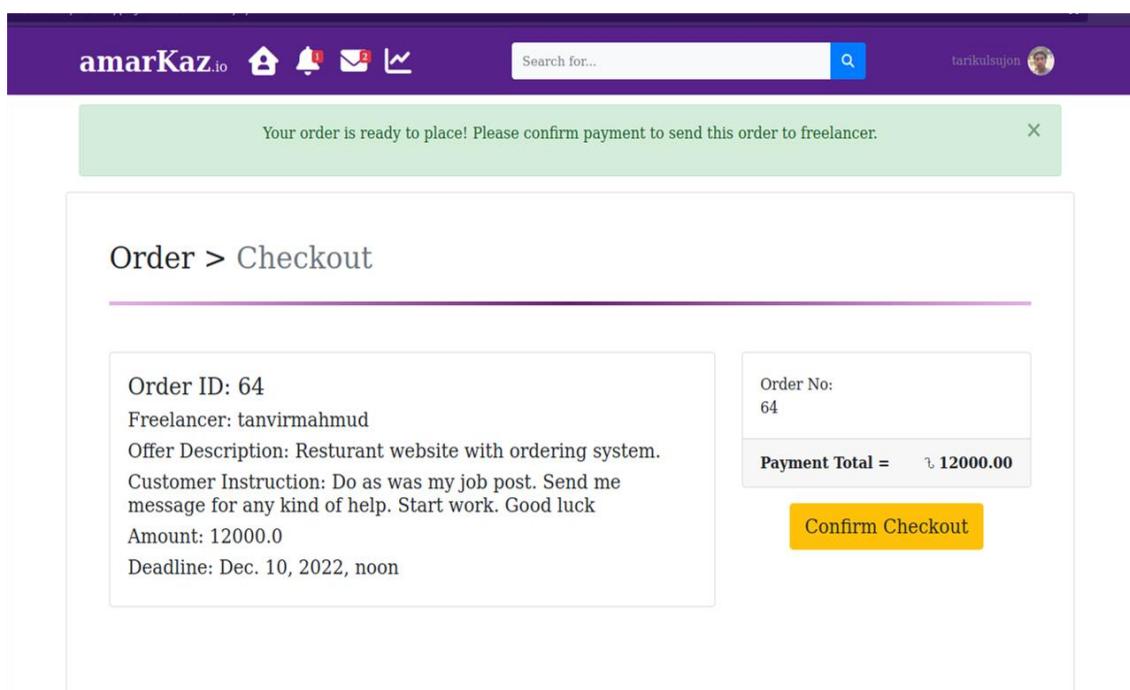


Figure 4.2.34 - Interface for order payment checkout for customer.

The screenshot shows the amarKaz.io customer interface. At the top, there is a purple header bar with the website name "amarKaz.io" and icons for home, notifications (1), messages (2), and account. A search bar with placeholder text "Search for..." and a magnifying glass icon is also present. On the right side of the header, the user's name "tarikulsujon" and profile picture are shown. The main content area has a light gray background. It displays a message: "Your payment will be pending, after your task is completely done then the freelancer will receive money to his account." To the right of this message, the amount "Amount: 12000.0 ₪" is listed, followed by a checked checkbox labeled "Confirmation" and a green "Confirm Payment" button.

Confirm Payment

Your payment will be pending, after your task is completely done then the freelancer will receive money to his account.

Amount: 12000.0 ₪

Confirmation

Confirm Payment

Figure 4.2.35 - Interface for payment confirmation for customer.

The screenshot shows the amarKaz.io customer interface. The top navigation bar is identical to Figure 4.2.35. The main content area is titled "All Orders". It displays a single "Placed Order" card. The order details are: Order ID: 4, Deadline: Dec. 10, 2022, noon, Accepted: False, and Status: False. Below the details is a yellow "Cancel Order" button.

Figure 4.2.36 - Interface for all kinds of order for customer.

The screenshot shows the amarKaz.io customer interface. The top navigation bar is identical to Figure 4.2.35. The main content area is titled "Pending Payments". It displays a single card with the following payment details: Payment Id: 3, Order Id: 3, Amount: 500, Confirmation: True, and Success: False.

Figure 4.2.37 - Interface for order checkout for customer.

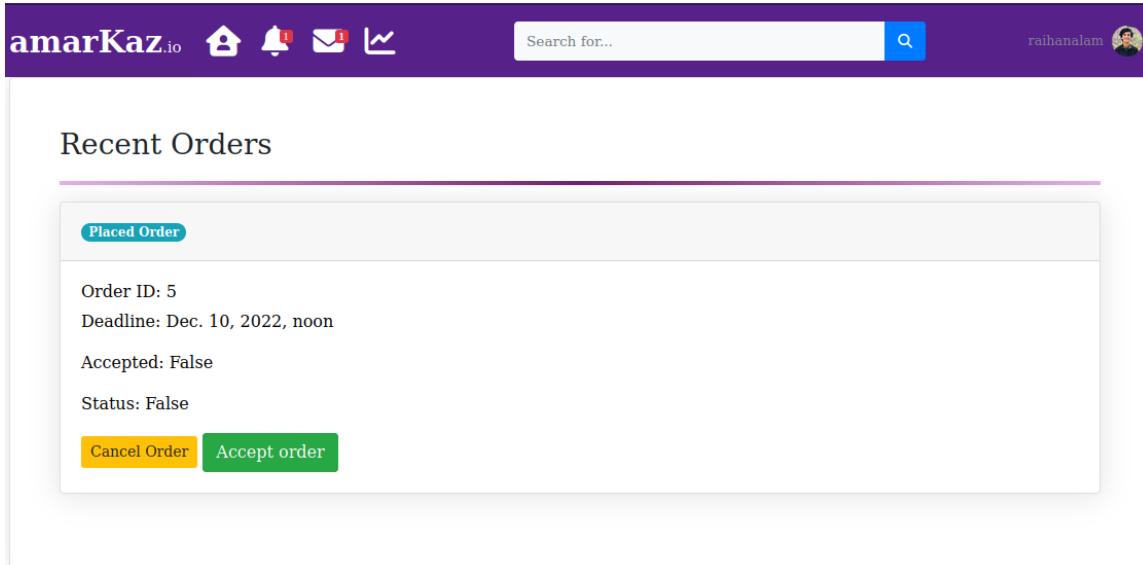


Figure 4.2.38 - Interface for recent orders for freelancer.

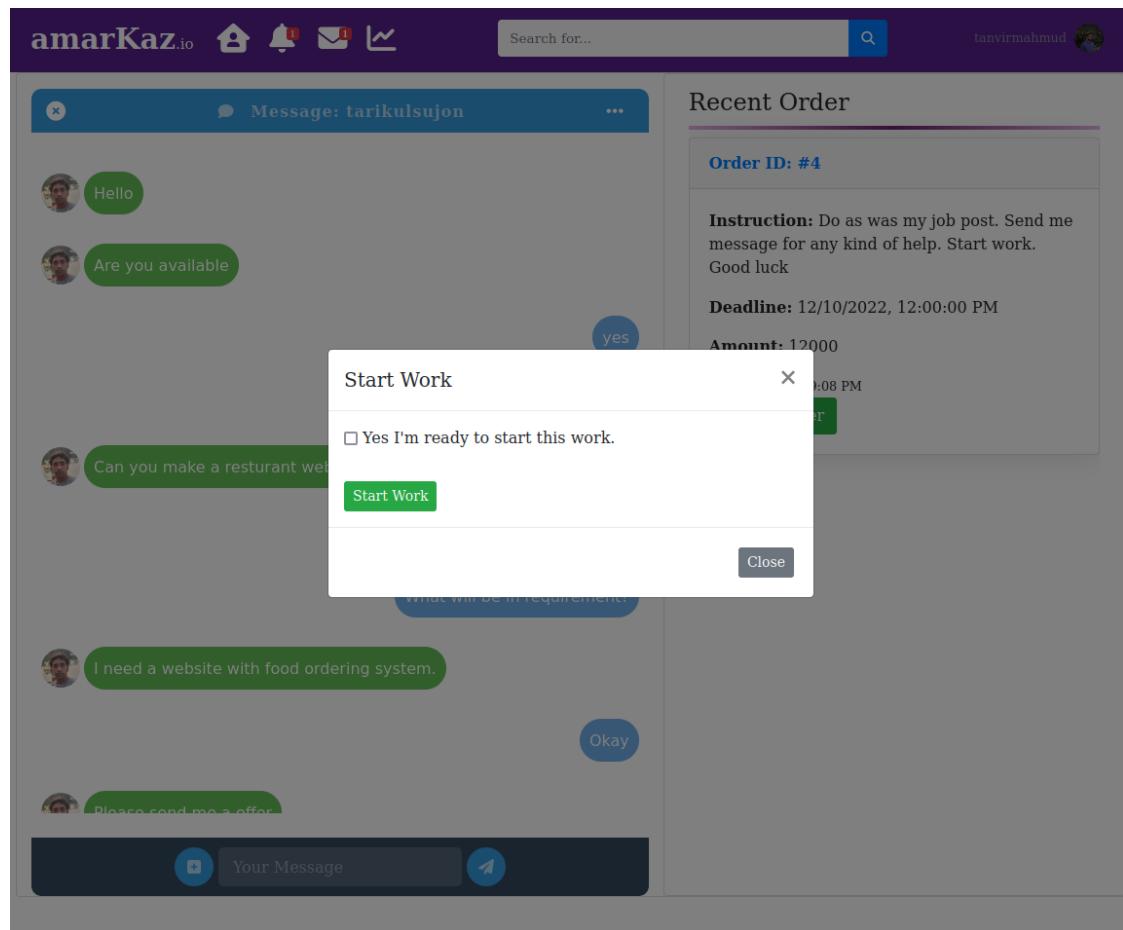


Figure 4.2.39 - Interface for accepting order and start work checkout for customer.

Running work where user will act for file sharing, deadline counter, complete, cancel button and order information for freelancer, the view is dynamic for different actor.

The screenshot shows the amarKaz.io platform interface for a freelancer named tanvirmahmud. The top navigation bar includes icons for home, notifications, messages, and a search bar. The main content area is titled "Running Work".

ON: Order-4

Freelancer: tanvirmahmud
Customer: tarikulsujon

Work History

Running: True
Track: 0
Status: False
Work Started: Nov. 26, 2022, 9:05 p.m.

Order History

Instruction: Do as was my job post. Send me message for any kind of help. Start work. Good luck
Deadline: Dec. 10, 2022, noon
Amount: 12000.0

Offer History

Description: Restaurant website with ordering system.
Expected Time: Dec. 9, 2022
Support: 5

Work Status

Deadline is :
Dec 10, 2022 12:12:00
Time left:
13 days, 15 hours, 5 minutes, 10 seconds

Tracker 0%

Pop **Push**

Work File: **Browse...** No file selected.
Send

Complete Work **Cancel Work** **Report Bad Experience**

Work Files

tanvirmahmud
Nov. 26, 2022, 9:06 p.m.
v6.13.0-dist.zip
Download

Figure 4.2.40 - Interface for running work with tracker and file sharing for freelancer.

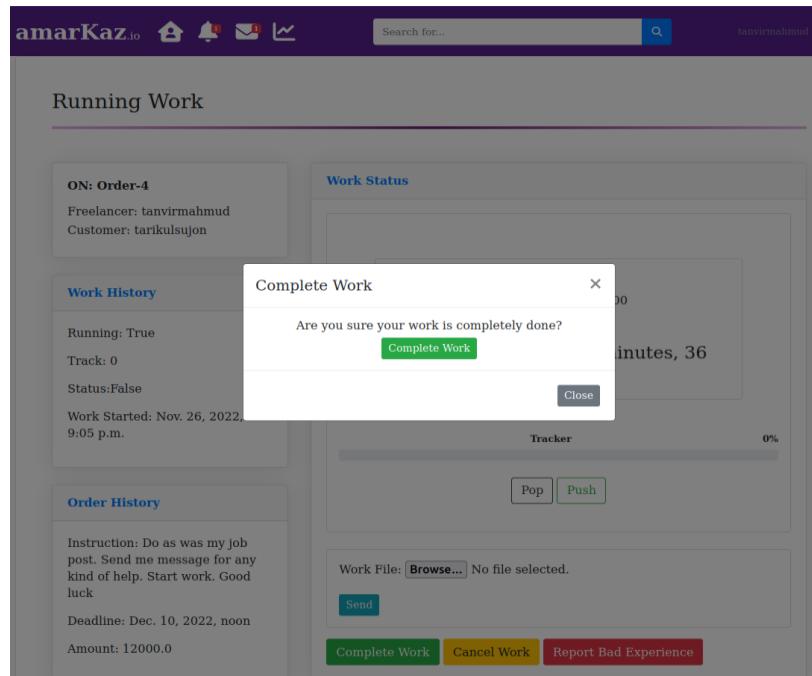


Figure 4.2.41 - Interface for complete work for customer.

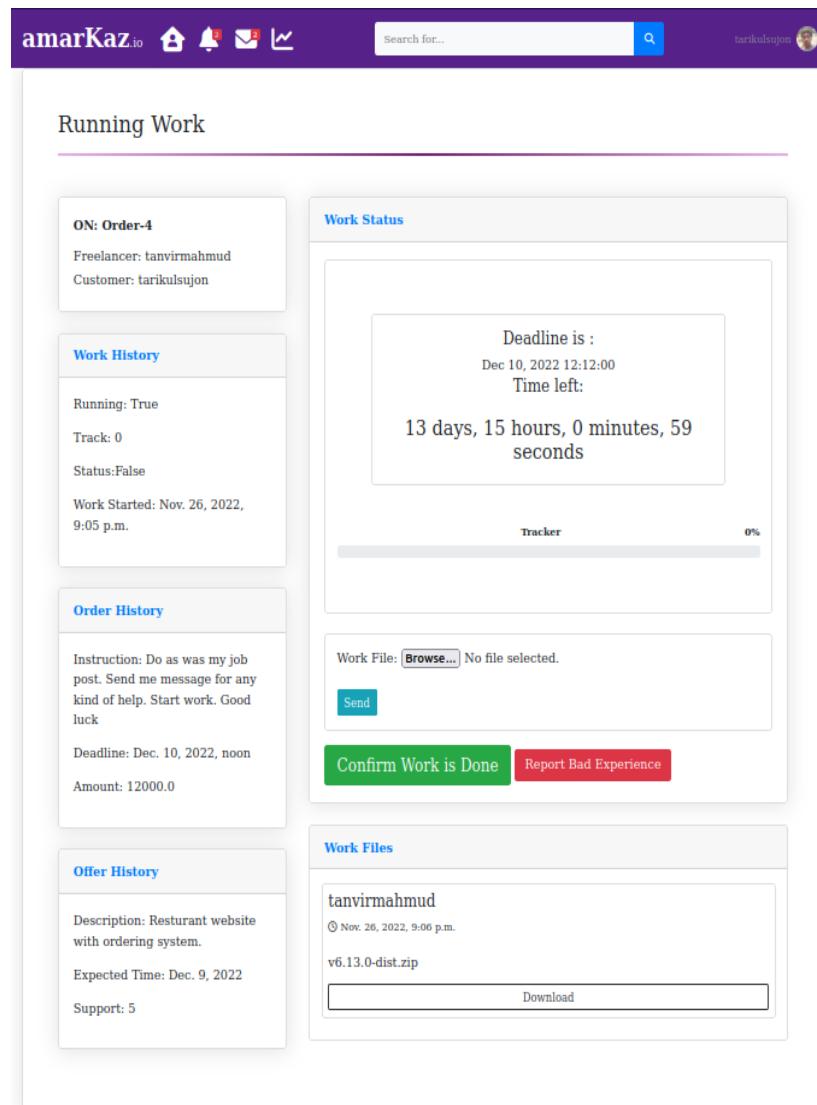


Figure 4.2.42 - Interface for running work with work tracking for customer.

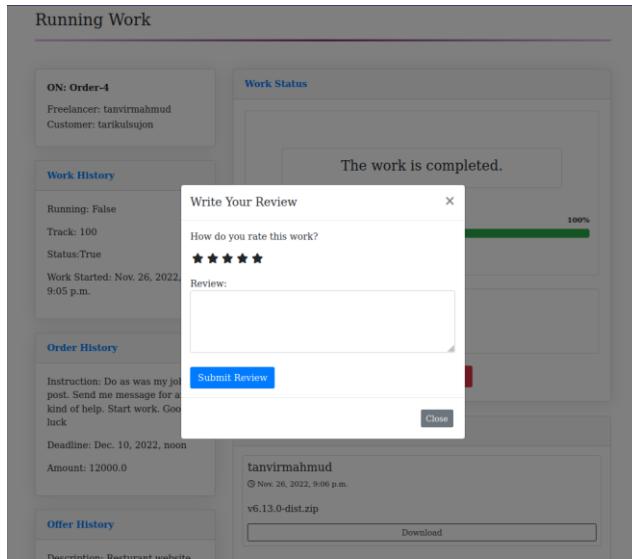


Figure 4.2.43 - Interface for review & ratings after a work is complete for customer.

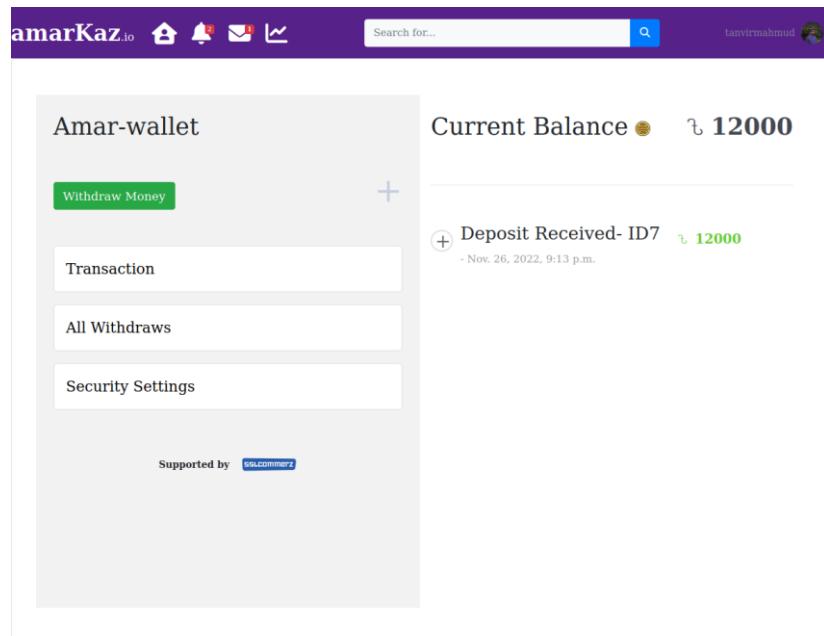


Figure 4.2.44 - Interface for wallet after deposit received after a complete work for freelancer.

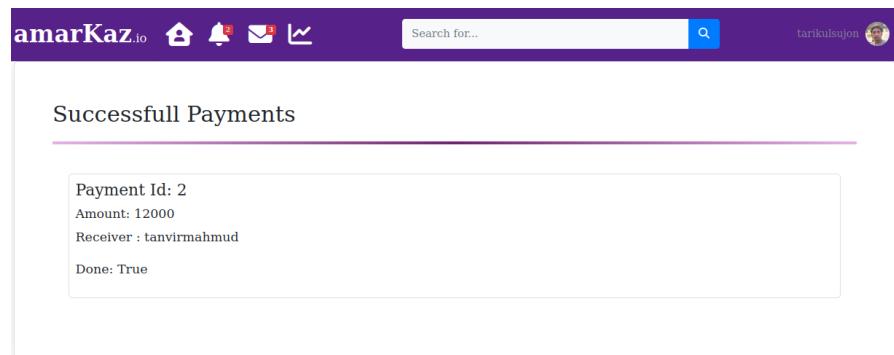


Figure 4.2.45 - Interface for successful payments after a complete work for customer.

The screenshot shows the 'Withdraw > Request' page. On the left, there's a form with fields for Reference (Withdrawal 1000), Method (Bank), Account number (6727365723), and Amount (10500). A 'Save Withdraw Form' button is at the bottom. On the right, a summary box shows: Reference: Withdrawal 1000, Method: bank, Withdrawal Amount: 10500, Network Fee: 945.00, and Receivable Amount = ₮ 9555.00. A 'Send Withdrawal Request' button is also present.

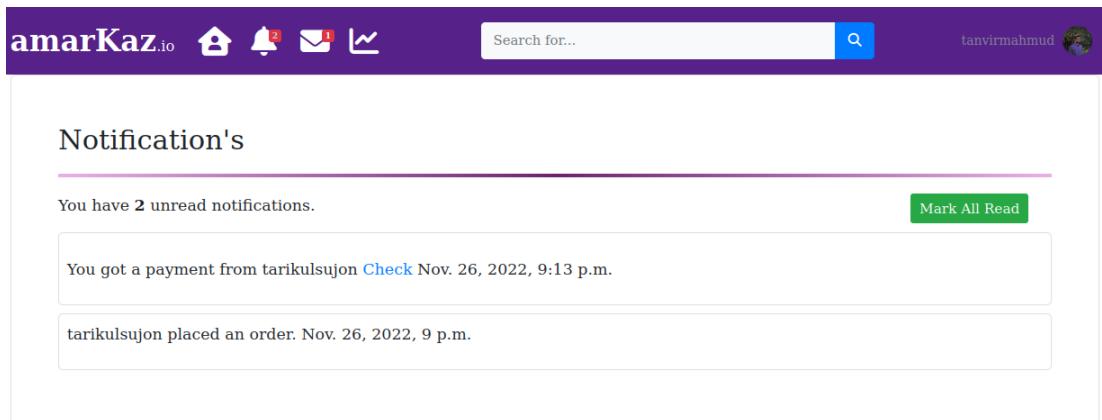
Figure 4.2.46 - Interface for withdrawal checkout for freelancer.

The screenshot shows the 'All Withdrawal Requests' page. It displays a table with columns: Name, Method, Acc. Number, Amount, Receivable, Fee, and Status. One entry is shown: tanvirmahmud, bank, 6727365723, ₮ 10500, ₮ 9555.0, 945, pending. Below the table, it says 'Showing 5 out of 25 entries' and has a navigation bar with buttons for Previous, 1, 2, 3, 4, 5, and Next.

Figure 4.2.47 - Interface for all withdrawal requests with status for freelancer.

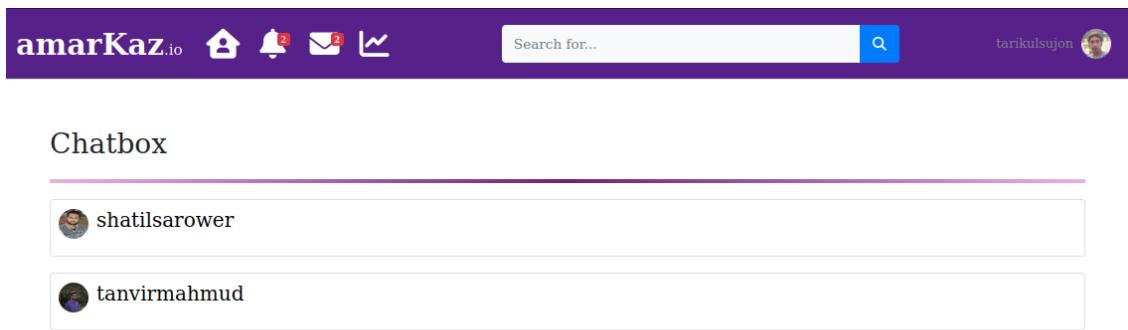
The screenshot shows the 'Incomplete Order' page. It lists two orders: Order ID: 8 (Amount: ₮ 555.0, Date: Dec. 10, 2022, 5:02 p.m.) and Order ID: 7 (Amount: ₮ 5600.0, Date: Dec. 10, 2022, 4:56 p.m.). Each order has 'Complete Payment' and 'Cancel Order' buttons.

Figure 4.2.48 - Interface for incomplete order payment checkout.



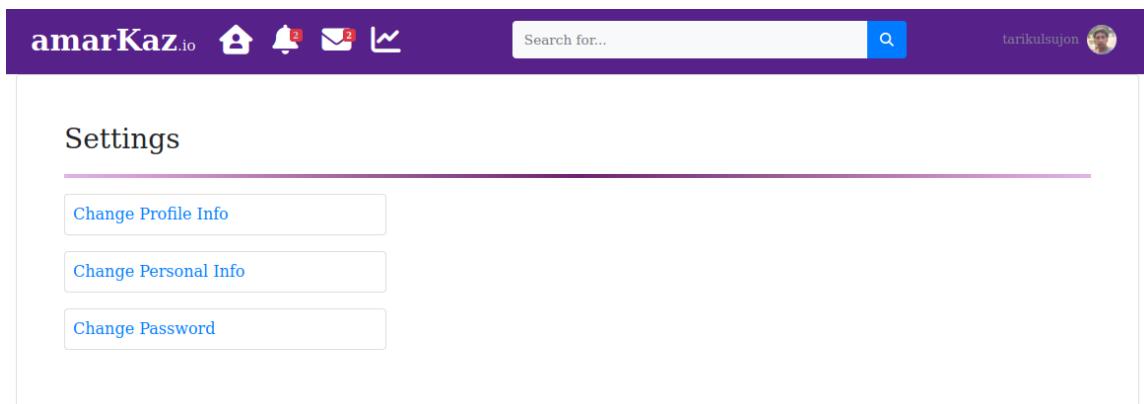
The screenshot shows the notification interface of the amarKaz.io application. At the top, there is a purple header bar with the website logo 'amarKaz.io' and several icons for navigation and notifications. A search bar is located in the center of the header. On the right side of the header, the user's name 'tanvirmahmud' is displayed next to a small profile picture. Below the header, the main content area has a light gray background. The title 'Notification's' is centered at the top of this area. A message states 'You have 2 unread notifications.' To the right of this message is a green button labeled 'Mark All Read'. Two notifications are listed in separate boxes: 'You got a payment from tarikulsujon' with a timestamp 'Nov. 26, 2022, 9:13 p.m.' and 'tarikulsujon placed an order.' with the same timestamp.

Figure 4.2.49 - Interface for notifications for notifying a actions to an user .



The screenshot shows the chatbox interface of the amarKaz.io application. The top navigation bar is identical to Figure 4.2.49. The main content area is titled 'Chatbox'. It displays two users connected to the chat: 'shatilsarower' and 'tanvirmahmud', each with a small profile picture next to their names.

Figure 4.2.50 - Interface for user chat box shows connected user to chat.



The screenshot shows the settings interface of the amarKaz.io application. The top navigation bar is identical to the previous figures. The main content area is titled 'Settings'. It contains three buttons for managing account information: 'Change Profile Info', 'Change Personal Info', and 'Change Password', all in blue text.

Figure 4.2.51 - Interface for settings to change profile, personal information and password.

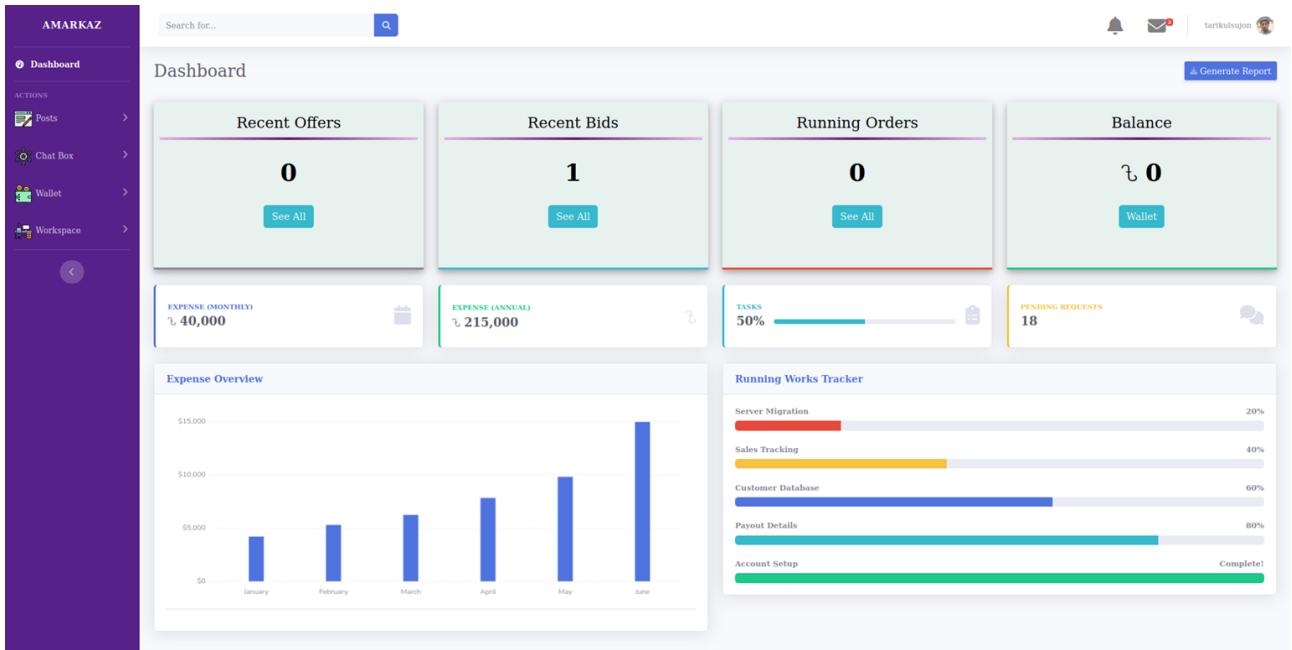


Figure 4.2.52 - Interface for dashboard with recent actions, balance and some statistics for customer.

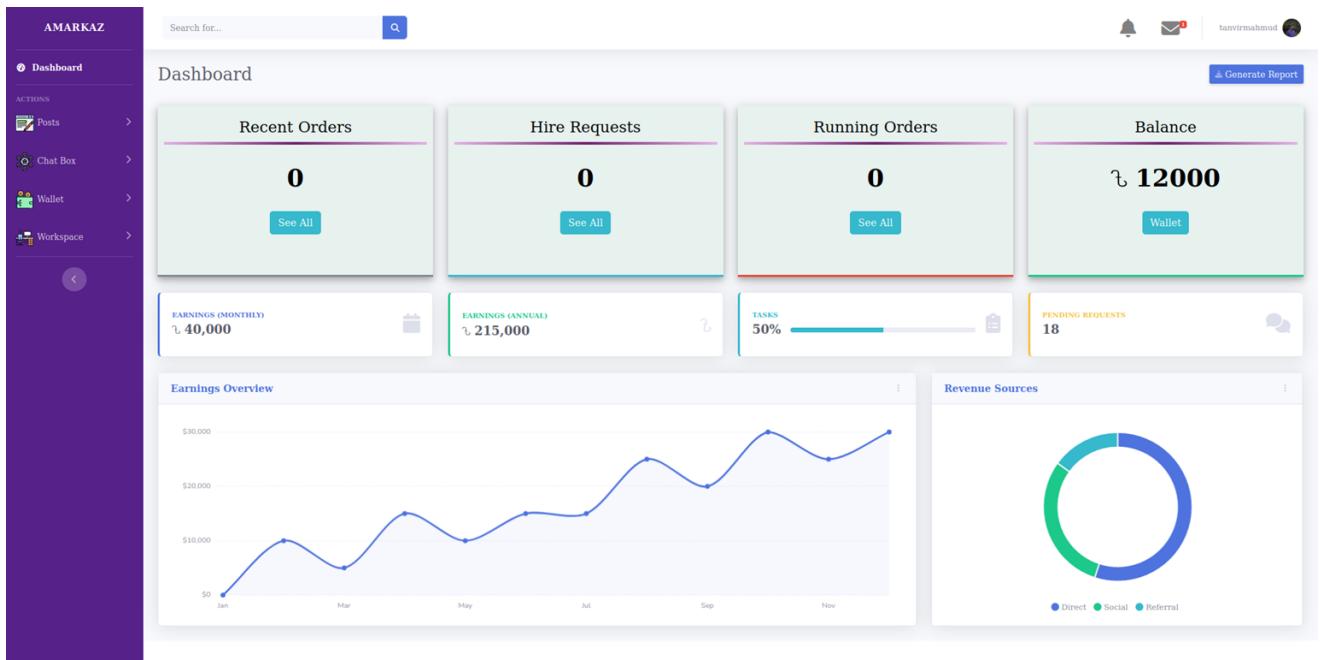


Figure 4.2.53 - Interface for dashboard with recent actions, balance and statistics for freelancer.

The screenshot shows the amarKaz.io platform interface for a customer. At the top, there's a purple header bar with the logo 'amarKaz.io' and several icons (home, notifications, messages, etc.). A search bar with placeholder 'Search for...' and a magnifying glass icon is on the right. The user profile 'tarikulsujon' is also visible. Below the header, the title 'Recent Bids' is centered. Two bid cards are displayed in a grid:

- User: raihanalam**
Post: Need a web developer to make a website for my restaurant.
Note: Hello, I am experienced about your work.
Amount: 15000
[See Portfolio](#) [Contact to Freelancer](#) [Reject](#)
- User: tanvirmahmud**
Post: Need a web developer to make a website for my restaurant.
Note: Hello, I am interested in your project.
Amount: 15000
[See Portfolio](#) [Contact to Freelancer](#) [Reject](#)

Figure 4.2.54 - Interface for recent bids for customer.

The screenshot shows the amarKaz.io platform interface for a freelancer. The top bar is identical to the customer view. The user profile 'raihanalalm' is shown. Below the header, the title 'Recent Hire Requests' is centered. One request card is visible:

- User: tarikulsujon**
Post: Need a web developer to make a website for my restaurant.
Note: Hello can you please check this?
Amount: 10000
[Send Offer](#)

Figure 4.2.55 - Interface for recent bids for freelancer.

The screenshot shows the amarKaz.io platform interface for a customer. The top bar is identical. The user profile 'tarikulsujon' is shown. Below the header, the title 'All Offers' is centered. One offer card is visible:

- Offer ID: #OF-11**
Recent
Description: One complete Restaurant website with ordering system.
Offer Amount: 10000
Expected Time: Dec. 31, 2022
Support: 5
Time: Dec. 10, 2022, 9:55 p.m.
[Place An order](#)

Figure 4.2.56 - Interface for all offers for customer.

The screenshot shows the amarKaz.io interface. At the top, there is a purple header bar with the logo 'amarKaz.io' and various icons for navigation and notifications. A search bar is located in the center of the header. On the right side of the header, there is a user profile for 'raihanalam' with a small profile picture. Below the header, the main content area has a title 'Running Orders'. Underneath this, there is a card titled 'Running Order' containing the following details:

- Order ID: 5
- Deadline: Dec. 10, 2022, noon
- Accepted: True
- Status: False

Figure 4.2.57 - Interface for running orders for both user.

The screenshot shows the amarKaz.io interface with a purple header bar. The main content area displays a search result for 'Filter for category 'Web Design'' with a total of 'result 2'. Two service posts are listed:

- ashrafulislam**
I will design awesome website or landing page...
₹ 150.00
[Save](#) [View](#)
- shafin**
I will do web design in figma, psd website design, or UI UX ...
₹ 150.00
[Save](#) [View](#)

Figure 4.2.58 - Interface for filter results by categories shows ports for customer .

The screenshot shows the amarKaz.io interface with a purple header bar. The main content area displays a search result for 'Search for 'website'' with a total of 'result 2'. Two job posts are listed:

- tarikulsujon** Need a web developer to make a website for my restaurant.
Nov. 19, 2022, 2:49 p.m.
₹ 10000 [Bid](#)
- parvezsumon** Need a Full-Stack developer for my project
Nov. 19, 2022, 5:20 a.m.
₹ 20000 [Bid](#)

Figure 4.2.59 - Interface for search results by keywords shows job post for freelancer.

amarKaz Administration

Site administration

The screenshot shows the Django Admin interface with the following sections:

- ACCOUNT_APP**: User profiles, Users, Verifications.
- ACCOUNTS**: Email addresses.
- AUTHENTICATION AND AUTHORIZATION**: Groups.
- MAIN_APP**: Bids, Categories, Hires, Port image gallery, Ports, Posts, Review ratings.
- MESSAGE_APP**: Chat rooms, Chats.
- NOTIFICATIONS**: Notifications.
- SEARCH_APP**: Search histories.
- SITES**: Sites.
- SOCIAL ACCOUNTS**: Social accounts, Social application tokens, Social applications.
- WALLET_APP**: Currencies, Deposits, Payments, Pending payments, Transactions, User wallets, Wallets, Withdrawal requests.
- WORK_APP**: Complete works, Offers, Orders, Reports, Running work filess, Running works.

Recent actions

- ReviewRating object (4) Review rating
- ReviewRating object (5) Review rating
- ReviewRating object (6) Review rating
- ReviewRating object (7) Review rating
- ReviewRating object (8) Review rating
- ReviewRating object (9) Review rating
- test Bid
- test Bid
- Payment object (1) Payment
- Payment object (2) Payment

Figure 4.2.60 - Interface for admin panel with all database model access for admin.

5 Conclusion

After this successful project, we achieved our final goal. This is a huge milestone for us, we developed what was our requirements and needs. There was lots of problems, challenge and bugs and errors in our development time but we took challenge and did it. We motivate our-self “There is always a way”. There will come lot of update and optimization in our next SDLC iteration.

5.1 Limitations

User have to wait for profile verification we will manually verify every profile of our partner and freelancer. Our system is only for workspace, not for any personal interactions, user can't share there personal information like email, phone number, social media accounts etc via the messages. We will track all your messages. Can't pay or purchase something from other market by wallet money, this is a closed wallet system, means wallet for only this marketplace. Have to wait for money withdrawal requests approval admin will approve and transfer money to user bank account.

5.2 Future Work

Lot of Research & Development we make our product workable and after reaching this goal, we have another plan to make this project workable in real world. We will host this application in a cloud space, and we will buy a business domain for this. We will make marketing in our connections and we will try to find funds and ventures capital for our project to make this top job marketplace in the world.

References

- [1] F-commerce adding a new dimension to Bangladesh's digital commerce landscape.
[Online]. Available: <https://bd.apparelresources.com/business-news/retail/f-commerce-adding-new-dimension-bangladesh-digital-commerce-landscape/>
- [2] F-commerce growing exponentially
[Online] Available: <https://businesspostbd.com/front/f-commerce-growing-exponentially-2022-06-16>
- [3] E-commerce would be a \$3b industry by 2023
[Online] Available: <https://archive.dhakatribune.com/business/2020/11/03/e-commerce-market-will-be-3-billion-big-by-2023>
- [4] Django was invented to meet fast-moving newsroom deadlines, while satisfying the tough requirements of experienced web developers.
[Online] Available: <https://www.djangoproject.com/start/overview/>
- [5] Django Tutorial Part 5: Creating our home page
[Online] Available: https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Home_page
- [6] ASGI is a spiritual successor to WSGI, the long-standing Python standard for compatibility between web servers, frameworks, and applications.
[Online] Available: <https://asgi.readthedocs.io/en/latest/introduction.html>