FREELANCERS

SENIOR PROJECT BY

HAMZA YAZBEK

Submitted to the School of Arts & Science of the Lebanese International University

In part of fulfillment of the requirements for the degree of

BACHERLOR OF SCIENCE IN COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

Supervised by: Dr. NAYEF SALEH

Spring: 2022 – 2023

Contents

About Our Project:

Welcome to our project documentation! This document provides an overview of our project, its goals, features, and technical details. We hope this guide will help you understand our project and its implementation.

Project Description:

Our project is a web-based platform designed to connect freelancers with potential clients, facilitating the hiring process and promoting collaboration. It aims to provide a seamless experience for both freelancers and clients, offering a wide range of features and functionalities to support their needs.

Key Features:

1. User Registration and Authentication: Users can create accounts, log in, and access personalized profiles.
2. Freelancer Profile Creation: Freelancers can create detailed profiles, showcasing their skills, experience, and portfolio.
3. Job Posting and Search: Clients can post job opportunities, and freelancers can search and apply for available jobs.
4. Ratings and Reviews: Users can rate and review freelancers and clients based on their experience.
5. Reporting System: Users can report issues, ensuring a safe and reliable environment for everyone.

Technologies Used:

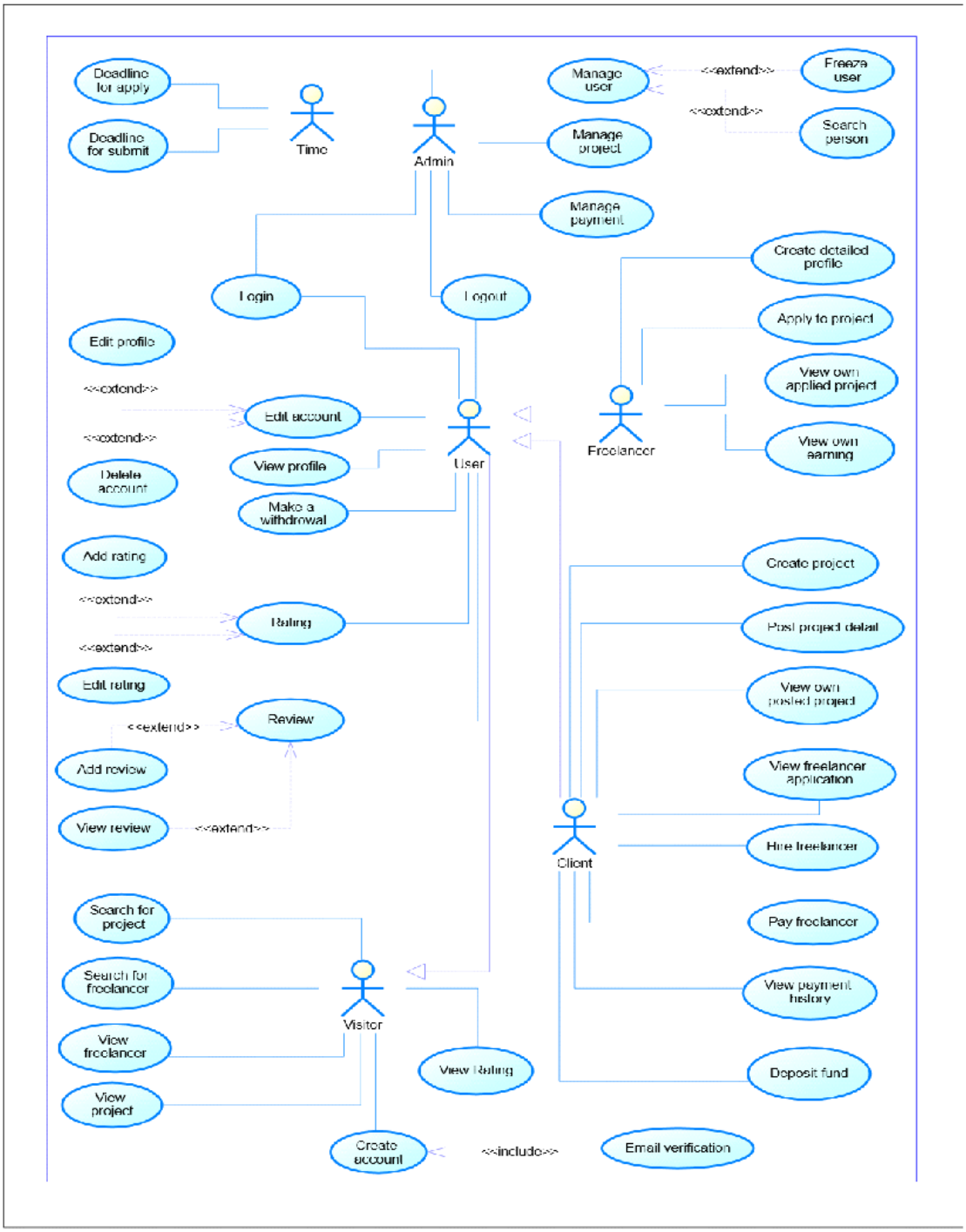
Our project is developed using modern web technologies and frameworks, including:

1. Front-end: Angular, HTML, CSS, and TypeScript.
2. Back-end: Laravel PHP framework, MySQL database.
3. RESTful APIs: Communication between the front-end and back-end is handled through RESTful APIs.
4. Authentication and Security: Token-based authentication and secure data transmission using HTTPS.

Contact Us:

If you have any questions, feedback, or issues regarding our project, please don't hesitate to contact our support team. We value your input and are committed to providing the best possible experience for our users.

USECASE DIAGRAM



NARATIVE USECASE

LOGOUT

Use-Case Name:	logout	
Use-Case ID:	logout-01	
Priority:	Medium	
Primary Actor:	User, admin	
Description:	User will logout from the System.	
Precondition:	Being logged in.	
Trigger:	To-logout.	
Typical Course of Events:	Actor Action	System Response
	-User will click on the Logout button	-Logout User from the system by deleting the access token from the database.
Alternate Courses:	Not click the Logout button.	
Conclusion:	The user is logged out of the system.	
Post-condition:	User get redirected to the login page.	

CREATE ACCOUNT

Use-Case Name:	Create account	
Use-Case ID:	Create-acount-02	
Priority:	High	
Primary Actor:	Visitor	
Description:	This use case is used to create new account as a freelancer or client.	
Precondition:	New and valid Email Address	
Trigger:	To create new account	
Typical Course of Events:	Actor Action	System Response
	-Enter details -Create account -validate email address by pressing on the send link or entering numbers that are sent	-check validation. -save account in database. -send Authentication token to the user.
Alternate Courses:	If the email is not valid or already exist in the database then system will tell user and will not save the information's in the database.	
Conclusion:	A new account will be created	
postcondition	The user can now log in to the system using their registered email address and password.	

LOGIN

Use-Case Name:	login	
Use-Case ID:	login-03	
Priority:	High	
Primary Actor:	User, Admin	
Description:	This use case is used to log in with existing account	
Precondition:	Having a valid account	
Trigger:	To login	
Typical Course of Events:	Actor Action	System Response
	-Enter username and password. -Click on the login button.	-The system verifies the username and password. -If the username and password are correct, the system logs the user into the system and send Auth Token. -The system displays an error message if the username and password are incorrect.
Alternate Courses:	If the user forgets their password, they can click on the "forgot password" link. The system will then ask for the user's email address, and send a password reset link to their email. The user can then follow the link to reset their password.	
Conclusion:	The user is logged into the system.	
postcondition	The user has access to their account and can perform various actions based on their account type.	

EMAIL VERIFICATION

Use-Case Name:	Email Verification	
Use-Case ID:	Email_Verification-04	
Priority:	High	
Primary Actor:	Visitor	
Description:	This use case is used to verify an email.	
Precondition:	Sending an email with the register.	
Trigger:	Verify the email.	
Typical Course of Events:	Actor Action	System Response
	-fill the registration form. -write the code received from email.	-send a code for the email. -waiting to receive the same code from the visitor.
Alternate Courses:	If the code that was send doesn't be send back by the visitor the it will cancel the registration process.	
Conclusion:	The Email owner will be confirmed.	
Post-condition:	The Email will be saved in the database.	

VIEW PROJECTS

Use-Case Name:	View Projects	
Use-Case ID:	View Projects-05	
Priority:	Medium	
Primary Actor:	Visitor	
Description:	The visitor will view the recruiter’s posted jobs.	
Precondition:	-Have internet. -Visit the website.	
Trigger:	View the posts in the website.	
Typical Course of Events:	Actor Action	System Response
	-Navigate to the "Projects" page. -View the Category of posted projects. -Click on a project to view its details. -View the project's details, such as its description, requirements, budget, and deadline.	-send the latest post’s information.
Alternate Courses:	None	
Conclusion:	The Email own-View the list of posted projects and their details.er will be confirmed.	
Post-condition:	The visitor is still not logged in to the system, and therefore cannot bid on or apply for any of the projects. They can, however, choose to create an account and log in to the system in order to bid on or apply for projects that they are interested in.	

SEARCH FOR PROJECT

Use-Case Name:	Search for project	
Use-Case ID:	Search for project-06	
Priority:	Medium	
Primary Actor:	Visitor	
Description:	This use case is used by any visitor to search for specific projects on the freelancing website.	
Precondition:	-Have the internet. -Visit the website.	
Trigger:	The visitor wants to search for specific projects.	
Typical Course of Events:	Actor Action	System Response
	-Navigate to the "Projects" page. -choose the category. -Enter keywords or specific project details into the search bar. -Click on the "search" button. -View the list of projects that match the search criteria. -Click on a project to view its details.	-send projects with searched tags.
Alternate Courses:	If there are no projects that match the search criteria, the system displays a message indicating that there are no results found.	
Conclusion:	The visitor is able to search for and view specific projects that match their search criteria.	
Post-condition:	The visitor is still not logged in to the system, and therefore cannot bid on or apply for any of the projects. They can, however, choose to create an account and log in to the system in order to bid on or apply for projects that they are interested in.	

VIEW FREELANCER

Use-Case Name:	View Freelancer	
Use-Case ID:	View_Freelancer-07	
Priority:	Medium	
Primary Actor:	Visitor	
Description:	This use case is used to view the profile of a specific freelancer on the freelancing website.	
Precondition:	-Have the internet. -Visit the website.	
Trigger:	To view the profile of a specific freelancer.	
Typical Course of Events:	Actor Action	System Response
	-Search for the freelancer by name or skill set. -Click on the freelancer's name or profile picture. -View the freelancer's profile, including their name, profile picture, skill set, work experience, education, and ratings/reviews.	-send the latest freelancer's information.
Alternate Courses:	If the code that was send doesn't be send back by the visitor the it will cancel the registration process.	
Conclusion:	The Email owner will be confirmed.	
Post-condition:	The Email will be saved in the database.	

SEARCH FOR FREELANCER

Use-Case Name:	Search for freelancer	
Use-Case ID:	Search for freelancer-08	
Priority:	Medium	
Primary Actor:	Visitor	
Description:	This use case is used by any visitor to search for specific freelancer on the freelancing website.	
Precondition:	-Have the internet. -Visit the website.	
Trigger:	To search for a specific freelancer on the freelancing website based on specific criteria.	
Typical Course of Events:	Actor Action	System Response
	-Navigate to the "Freelancers" page. -choose the category. -Enter keywords or specific project details into the search bar. -Click on the "search" button. -View the list of freelancers that match the search criteria. -Click on a freelancer to view his/her details.	-send freelancers with searched skills.
Alternate Courses:	If there are no freelancers that match the search criteria, the system displays a message indicating that there are no results found.	
Conclusion:	The visitor is able to search for and view specific freelancer that match their search criteria. Must be client to contact the freelancer.	
Post-condition:	The client can choose to contact the freelancer to discuss potential projects, or to invite the freelancer to bid on an existing project.	

RATING

Use-Case Name:	Rating	
Use-Case ID:	Rating-00	
Priority:	Low	
Primary Actor:	User	
Description:	User can rate another user after the job done.	
Precondition:	Having a completed project	
Trigger:	To	
Typical Course of Events:	Actor Action	System Response
	-Enter details -Create account -validate email address by pressing on the send link or entering numbers that are sent	-check validation -save account in database
Alternate Courses:	If details are invalid then system will tell user and will not save the information's in the database.	
Conclusion:	A new account will be created	
Post-condition:	User won't be able to create another account using the same email	

ADD RATING

Use-Case Name:	Add Rating	
Use-Case ID:	Add Rating-09	
Priority:	Low	
Primary Actor:	User	
Description:	This use case is used to add rating for different user.	
Precondition:	The freelancer or client has completed a project with the other party and is logged in to the system.	
Trigger:	The freelancer or client wants to rate the other party based on their experience working together on the project.	
Typical Course of Events:	Actor Action	System Response
	-Navigate to the project page or dashboard. -Find the completed project and click on the "rate" button next to the other party's name. -Enter a rating on a scale of 1-5, with optional comments or feedback. -Click on the "submit" button.	-Store this rating information in the database.
Alternate Courses:	If the project has not yet been completed, the system displays a message indicating that ratings cannot be submitted until the project is completed.	
Conclusion:	The freelancer or client is able to rate the other party based on their experience working together on the project.	
Post-condition:	The rating and feedback are saved in the system and is visible to other users who view the freelancer's or client's profile. The rated party is notified that they have received a rating and feedback from the other party. The system may use the ratings to calculate an overall rating or score for the freelancer or client.	

EDIT RATING

Use-Case Name:	Edit Rating	
Use-Case ID:	EditRating-10	
Priority:	Medium	
Primary Actor:	User	
Description:	This use case is used by a freelancer or client to edit a rating that they previously submitted for the other party.	
Precondition:	The freelancer or client has previously submitted a rating for the other party and is logged in to the system.	
Trigger:	The freelancer or client edits the rating they previously submitted.	
Typical Course of Events:	Actor Action	System Response
	-Navigate to the project page or dashboard. -Find the completed project and click on the "edit rating" button next to the other party's name. -Edit the rating on a scale of 1-5, with optional comments or feedback. -Click on the "submit" button.	-update the existed rating information.
Alternate Courses:	None	
Conclusion:	The freelancer or client is able to edit a rating that they previously submitted for the other party.	
Post-condition:	The updated rating and feedback are saved in the system and are visible to other users who view the freelancer's or client's profile.	

MAKE A WITHDROWAL

Use-Case Name:	Make a withdrawal	
Use-Case ID:	Withdraw-11	
Priority:	Medium	
Primary Actor:	User	
Description:	This use case is used by a user to withdraw money from his account.	
Precondition:	-Has logged in and has money in his account. -Payment information in the user account must be filled by valid data.	
Trigger:	To withdraw his/her money to the bank account.	
Typical Course of Events:	Actor Action	System Response
	-Navigate to the Wallet page. -Enter the amount that you want to withdraw and send the request.	-confirm the withdrawal information. -if the information's are valid will transfer the amount of the money to the user bank account. -if the information is not valid will send error message
Alternate Courses:	If there is no valid bank account provided by the user or the user doesn't have enough credits in his/her account to transfer the withdrawal request will be rejected.	
Conclusion:	The user can withdraw money from his account to a bank account.	
Post-condition:	The money that has been withdrawn will be subtracted from the total amount this user has in his/her account.	

VIEW RATING

Use-Case Name:	View Rating	
Use-Case ID:	View-Rating-12	
Priority:	Low	
Primary Actor:	Visitor	
Description:	This use case is used by a visitor, freelancer, or client to view the ratings and feedback that have been submitted for a specific freelancer or client.	
Precondition:	-The freelancer or client has completed at least one project on the platform and has received ratings and feedback from other users. Trigger: The visitor, freelancer, or client wants to view the ratings and feedback that have been submitted for a specific freelancer or client.	
Trigger:	The visitor, freelancer, or client wants to view the ratings and feedback that have been submitted for a specific freelancer or client.	
Typical Course of Events:	Actor Action	System Response
	-Navigate to the freelancer or client's profile page. -Click on the "ratings" or "feedback" tab. -View the ratings and feedback that have been submitted by other users.	-send the recent rating data to the website.
Alternate Courses:	If the freelancer or client has not completed any projects or has not received any ratings or feedback, the system displays a message indicating that there are no ratings or feedback available.	
Conclusion:	The visitor, freelancer, or client is able to view the ratings and feedback that have been submitted for a specific freelancer or client.	
Post-condition:	The ratings and feedback are displayed on the freelancer or client's profile page and are visible to other users who view the page	

EDIT ACCOUNT

Use-Case Name:	Edit account	
Use-Case ID:	Edit-account-13	
Priority:	Medium	
Primary Actor:	User	
Description:	This use case is used by a user to edit their account information, including their personal details and account settings.	
Precondition:	The user is logged in to the system and has the necessary permissions to edit their account.	
Trigger:	The user wants to edit their account information.	
Typical Course of Events:	Actor Action	System Response
	-Navigate to the account settings page. -Edit the desired fields, such as name, email address, password, or payment information. -Click on the "save" button.	-Update the new information in the database.
Alternate Courses:	If the information entered is invalid, the system displays an error message and prompts the user to enter valid information.	
Conclusion:	User can edit his/her account.	
Post-condition:	The account information will be updated.	

EDIT PROFILE

Use-Case Name:	Edit profile	
Use-Case ID:	Edit-profile-14	
Priority:	Medium	
Primary Actor:	User	
Description:	This use case is used by a freelancer or client to edit their profile information, including their bio, skills, portfolio, and other details that are visible to other users.	
Precondition:	The user is logged in to the system and has the necessary permissions to edit their profile	
Trigger:	The user wants to edit their profile information.	
Typical Course of Events:	Actor Action	System Response
	--Navigate to the profile page. -Click on the "edit" or "update" button. -Edit the desired fields, such as bio, skills, portfolio, or other details. -Click on the "save" or "update" button.	-Update the new information in the database.
Alternate Courses:	If the information entered is invalid, the system displays an error message and prompts the user to enter valid information.	
Conclusion:	User can edit his/her profile.	
Post-condition:	The account information will be updated.	

DELETE ACCOUNT

Use-Case Name:	Delete account	
Use-Case ID:	Delete account-15	
Priority:	Medium	
Primary Actor:	User	
Description:	This use case is used by a user to delete their account from the system.	
Precondition:	The user is logged in to the system and has the necessary permissions to edit their profile	
Trigger:	The user wants to permanently delete their account.	
Typical Course of Events:	Actor Action	System Response
	-Navigate to the account settings page. -Click on the "delete account" button. -Confirm the deletion by entering their password or other identifying information. -Click on the "delete" or "confirm" button.	-If the identity information is valid the account will be deleted from the database. -if not delete process will be rejected.
Alternate Courses:	If the user decides not to delete their account, they can cancel the deletion process by clicking on the "cancel" or "back" button.	
Conclusion:	The user is able to delete their account from the system.	
Post-condition:	All associated data (such as project history, ratings, and feedback) is permanently removed from the system.	

MANAGE PROJECT

Use-Case Name:	Manage Project	
Use-Case ID:	Manage Project-16	
Priority:	High	
Primary Actor:	Admin	
Description:	This use case is used by the admin to manage projects on the freelancing website, including creating new projects, editing existing projects, and deleting projects.	
Precondition:	The admin is logged in to the system and has the necessary permissions to manage projects.	
Trigger:	The admin wants to manage a project on the freelancing website.	
Typical Course of Events:	Actor Action	System Response
	-Select the project to manage. -Edit the project details, such as project description, budget, deadline, or other information. -or delete the entire project. -Click on the "save" or "update" button.	-Update the post information in the database.
Alternate Courses:	None	
Conclusion:	The admin is able to manage projects on the freelancing website, including editing existing projects, and assigning projects to freelancers or teams of freelancers.	
Post-condition:	The updated project information is saved in the system and is visible to other users who view the project details. If the project is deleted, all associated data (such as freelancer or team of freelancers assigned to the project, project history, ratings, and feedback) is permanently removed from the system.	

MANAGE PAYMENT

Use-Case Name:	Manage Payment	
Use-Case ID:	Manage Payment-17	
Priority:	High	
Primary Actor:	Admin	
Description:	This use case allows the admin to manage the payment of projects and freelancers.	
Precondition:	The admin has access to the system and is logged in.	
Trigger:	The admin chooses to manage the payment.	
Typical Course of Events:	Actor Action	System Response
	-Admin selects "Manage Payment" from the admin dashboard. -System displays a list of ongoing projects and freelancers. -Admin selects a project or a freelancer to manage payment. -Admin selects a payment option (e.g., approve, decline, refund).	-System displays the payment details and options to manage payment. -System processes the selected payment option and updates the payment status. -System sends a notification to the concerned parties (e.g., freelancer, client) regarding the payment -status update.
Alternate Courses:	-If the admin encounters any issues with the payment processing, the system displays an error message and suggests contacting the technical support team.	
Conclusion:	The admin manages the payment of a project or a freelancer, and the payment status is updated accordingly.	
Post-condition:	The concerned parties (e.g., freelancer, client) are notified of the payment status update, and the payment information is updated in the system.	

MANAGE USER

Use-Case Name:	Manage User	
Use-Case ID:	Manage User-18	
Priority:	High	
Primary Actor:	Admin	
Description:	This use case allows the admin to manage users, including searching for a user and freezing a user account.	
Precondition:	The admin has access to the system and is logged in.	
Trigger:	The admin chooses to manage a user.	
Typical Course of Events:	Actor Action	System Response
	-selects "Manage User" from the admin dashboard. -enters the user's information and selects "Search". -selects a user from the list to manage. -selects a user account option (e.g., freeze, unfreeze, delete).	-displays a search bar to search for a user. -The system displays a list of users that match the search criteria. -The system displays the user's details and options to manage the user account. -The system processes the selected user account option and updates the user account status.
Alternate Courses:	-If the admin encounters any issues with the user account processing, the system displays an error message and suggests contacting the technical support team.	
Conclusion:	The admin manages a user account, and the user account status is updated.	
Post-condition:	The concerned user is notified of the user account status update, and the user account information is updated in the system.	

CREATE DETAILED PROFILE

Use-Case Name:	Create detailed profile	
Use-Case ID:	Create-profile-19	
Priority:	High	
Primary Actor:	freelancer	
Description:	This use case allows a freelancer to create a detailed profile that highlights their skills and experience.	
Precondition:	The freelancer is registered and logged in.	
Trigger:	The freelancer chooses to create a detailed profile.	
Typical Course of Events:	Actor Action	System Response
	-selects "Create Detailed Profile" from their dashboard. -Freelancer fills out the data (name, skills, experience, education, etc...) and uploads their resume and portfolio. -Freelancer selects "Submit" to save the profile.	-The system verifies the uploaded information and displays a preview of the profile for the freelancer to review. -The system saves the profile in the database and sends a confirmation message to the freelancer.
Alternate Courses:	If the freelancer encounters any issues with the form or uploads, the system displays an error message and suggests to try again.	
Conclusion:	The freelancer creates a detailed profile that shows their skills and experience.	
Post-condition:	The freelancer's profile is saved in the system and can be viewed by clients.	

APPLY TO PROJECT

Use-Case Name:	Apply to project	
Use-Case ID:	Apply-20	
Priority:	High	
Primary Actor:	freelancer	
Description:	This use case allows a freelancer to apply for a project posted by a client.	
Precondition:	The freelancer is registered and logged in.	
Trigger:	The freelancer chooses to apply for a project.	
Typical Course of Events:	Actor Action	System Response
	-Apply to an available project. -Freelancer fills out the data (name, skills, experience, education, etc...) and uploads their resume and portfolio. -Reviews the project details and confirms that they meet the project requirements.	-The System displays the project details, including the project description, skills required, budget, and other relevant information. -The system sends the proposal message to the client. -The system updates the freelancer's applied projects list.
Alternate Courses:	If the freelancer already applied for the project, the system displays an error message.	
Conclusion:	The freelancer applies for a project by sending a proposal message to the client.	
Post-condition:	The freelancer's proposal message is sent to the client, and the freelancer's applied projects list is updated in the system.	

VIEW OWN APPLIED PROJECT

Use-Case Name:	View own applied project	
Use-Case ID:	Applied-Projects-21	
Priority:	Medium	
Primary Actor:	freelancer	
Description:	This use case allows a freelancer to view their own applied projects list.	
Precondition:	The freelancer is registered and logged in.	
Trigger:	The freelancer chooses to view their applied projects.	
Typical Course of Events:	Actor Action	System Response
	-selects a project from the list to view more details.	-The system displays a list of the freelancer's applied projects, including the project details and application status. -The system displays the project details and the proposal message sent to the client.
Alternate Courses:	If the freelancer has never applied to a project, nothing will appear.	
Conclusion:	The freelancer can view his proposals to the projects.	
Post-condition:	The freelancer will get the details for his applications.	

Use-Case Name:	View own earning	
Use-Case ID:	View-earning-22	
Priority:	Medium	
Primary Actor:	freelancer	
Description:	This use case is used by the freelancer to view their earning history on the website.	
Precondition:	The freelancer is registered and logged in.	
Trigger:	To view earning history.	
Typical Course of Events:	Actor Action	System Response
	-select his earning process from the earning history list.	-The system will display the earning history and details that was stored in the data base
Alternate Courses:	If the freelancer doesn't earn any money yet, the earning history will be empty.	
Conclusion:	The freelancer can view earning.	
Post-condition:	The client will be able to track their earning progress.	

VIEW OWN EARNING

DEADLINE FOR APPLY

Use-Case Name:	Deadline for apply	
Use-Case ID:	Deadline-for-apply-23	
Priority:	medium	
Primary Actor:	Time	
Description:	This use case is used to set the last time for the freelancer to apply for the project	
Precondition:	The freelancer is registered and logged in.	
Trigger:	The freelancer chooses to apply for a project.	
Typical Course of Events:	Actor Action	System Response
	-Freelancer check for the deadline for applying	-System show it.
Alternate Courses:	If freelancer attempts to apply for the project after the deadline has passed, the system displays an error message.	
Conclusion:	This use case allows the system set a deadline for freelancer to apply for a project.	
Post-condition:	Once the deadline for applying has end, the system will notify the client of all the received applications and allow them to select a freelancer.	

DEADLINE FOR SUBMIT

Use-Case Name:	Deadline for submit	
Use-Case ID:	Deadline-for-submit-24	
Priority:	medium	
Primary Actor:	Time	
Description:	This use case shows the last time for the freelancer to submit the project	
Precondition:	The freelancer is registered and logged in.	
Trigger:	The freelancer applied for this project.	
Typical Course of Events:	Actor Action	System Response
	-freelancer checks the deadline for submission.	-the deadline for submission will appear.
Alternate Courses:	If the freelancer didn't submit on time, the system doesn't accept the project.	
Conclusion:	The freelancer must submit his work before the specified time.	
Post-condition:	If freelancer submits the project before the deadline, the system will accept it, else the system will reject it.	

CREATE PROJECT

Use-Case Name:	Create project	
Use-Case ID:	Create-project-25	
Priority:	medium	
Primary Actor:	Client	
Description:	This use case is used to post new project by the clients.	
Precondition:	The client is registered and logged in.	
Trigger:	The client chooses to post a project.	
Typical Course of Events:	Actor Action	System Response
	-enter project detail. -post the project.	-view the project for the freelancers.
Alternate Courses:	The client can post many projects.	
Conclusion:	This use case allows client to easily post new projects.	
Post-condition:	The admin accepts the project to be posted, and a notification will send for the client that there project has been accepted.	

POST PROJECT DETAILS

Use-Case Name:	Post project details	
Use-Case ID:	Post-project-detail-26	
Priority:	medium	
Primary Actor:	Client	
Description:	This use case allows the client to add project details before submitting the project. The client can enter all the relevant information about the project.	
Precondition:	The client is registered and logged in.	
Trigger:	The client chooses to post a project and clicks on the "Add Project Details" button.	
Typical Course of Events:	Actor Action	System Response
	- clicks on "Add Project Details" button. - enters project details. -submits the project with details.	- displays a form to enter project details. - validates the details and allows the client to proceed. - displays a confirmation message.
Alternate Courses:	If the client does not provide all the required details, the system displays an error message and prompts the client to complete the missing fields.	
Conclusion:	This use case helps to ensure that the project details are complete and accurate before the project is submitted.	
Post-condition:	Once the client submits the project with details, the system saves the project details and prepares to display the project to potential freelancers.	

View own posted project

Use-Case Name:	View own posted project	
Use-Case ID:	View-own-posted-project-27	
Priority:	low	
Primary Actor:	Client	
Description:	This use case allows the client to view the projects that they have posted on the platform.	
Precondition:	The client is registered and logged in.	
Trigger:	The client chooses to view there posted project.	
Typical Course of Events:	Actor Action	System Response
	-choose to view the project he posts.	- displays a list of all the projects that the client has posted.
Alternate Courses:	If the client has not posted any projects, the system displays a message stating that the client has no posted projects.	
Conclusion:	This use case enables the client to keep track of the projects they have posted and make any necessary changes to the project details.	
Post-condition:	Once the client has finished reviewing the posted project, the system returns to the list of posted projects.	

View freelancer application

Use-Case Name:	View freelancer application	
Use-Case ID:	View-freelancer-application-28	
Priority:	medium	
Primary Actor:	client	
Description:	This use case enables the client to view the list of freelancers who have applied for their posted project.	
Precondition:	The client is registered and logged in.	
Trigger:	The client chooses to view the freelancer applications for their posted project.	
Typical Course of Events:	Actor Action	System Response
	-view applications. -select freelancer. -reviews the freelancer's application details.	-display a list of all freelancer application who have applied for the project. -displays the freelancer's application details. -allow to accept or reject application.
Alternate Courses:	If no freelancers have applied for the project, the system displays a message stating that there are no applications for the project.	
Conclusion:	This use case provides the client with an easy way to manage the freelancer applications for their posted project.	
Post-condition:	The client may choose to accept or reject one or more freelancer applications.	

Hire freelancer

Use-Case Name:	Hire freelancer	
Use-Case ID:	Hire-freelancer-29	
Priority:	medium	
Primary Actor:	Client	
Description:	This use case allows the client to hire a freelancer for their project from the list of applications received.	
Precondition:	The client is registered and logged in.	
Trigger:	The client has reviewed the list of freelancer applications and decided to hire one.	
Typical Course of Events:	Actor Action	System Response
	-selects the freelancer they want to hire. -confirms the selection.	-prompts the client to confirm their choice. -notifies the freelancer that they have been hired. -update the project status to “in progress”.
Alternate Courses:	-if the client changes their mind before confirming the selection, they can cancel the hire process. -if the freelancer declines the offer, the client can select another freelancer.	
Conclusion:	This use case enables the client to hire a suitable freelancer for their project.	
Post-condition:	The freelancer is notified that they have been hired, and the project statues is updated to “in progress”.	

Pay frrelancer

Use-Case Name:	Pay freelancer	
Use-Case ID:	Pay-freelancer-30	
Priority:	medium	
Primary Actor:	Client	
Description:	This use case allows the client to pay the freelancer for the completed project.	
Precondition:	The client is registered and logged in, and the project has been completed and submitted by the freelancer.	
Trigger:	The client chooses to pay the freelancer.	
Typical Course of Events:	Actor Action	System Response
	- selects the project for which payment is to be made. - selects the option to pay the freelancer for the project. - confirms the payment amount and initiates the payment.	- displays the payment amount previously agreed upon during the hiring process. - deducts the payment amount from the client's account balance and adds it to the freelancer's account balance. - sends a notification to the freelancer informing them of the payment and the amount received.
Alternate Courses:	-If the payment method fails, the system displays an error message and prompts the client to select a different payment method. -If the freelancer disputes the payment or there is an issue with the completed project, the system notifies both the client and the freelancer to resolve the issue before the payment can be made.	
Conclusion:	This use case allows the client to easily and securely pay the freelancer for the completed project.	
Post-condition:	The payment is made and the project is closed, with the payment amount deducted from the client's balance and added to the freelancer's balance.	

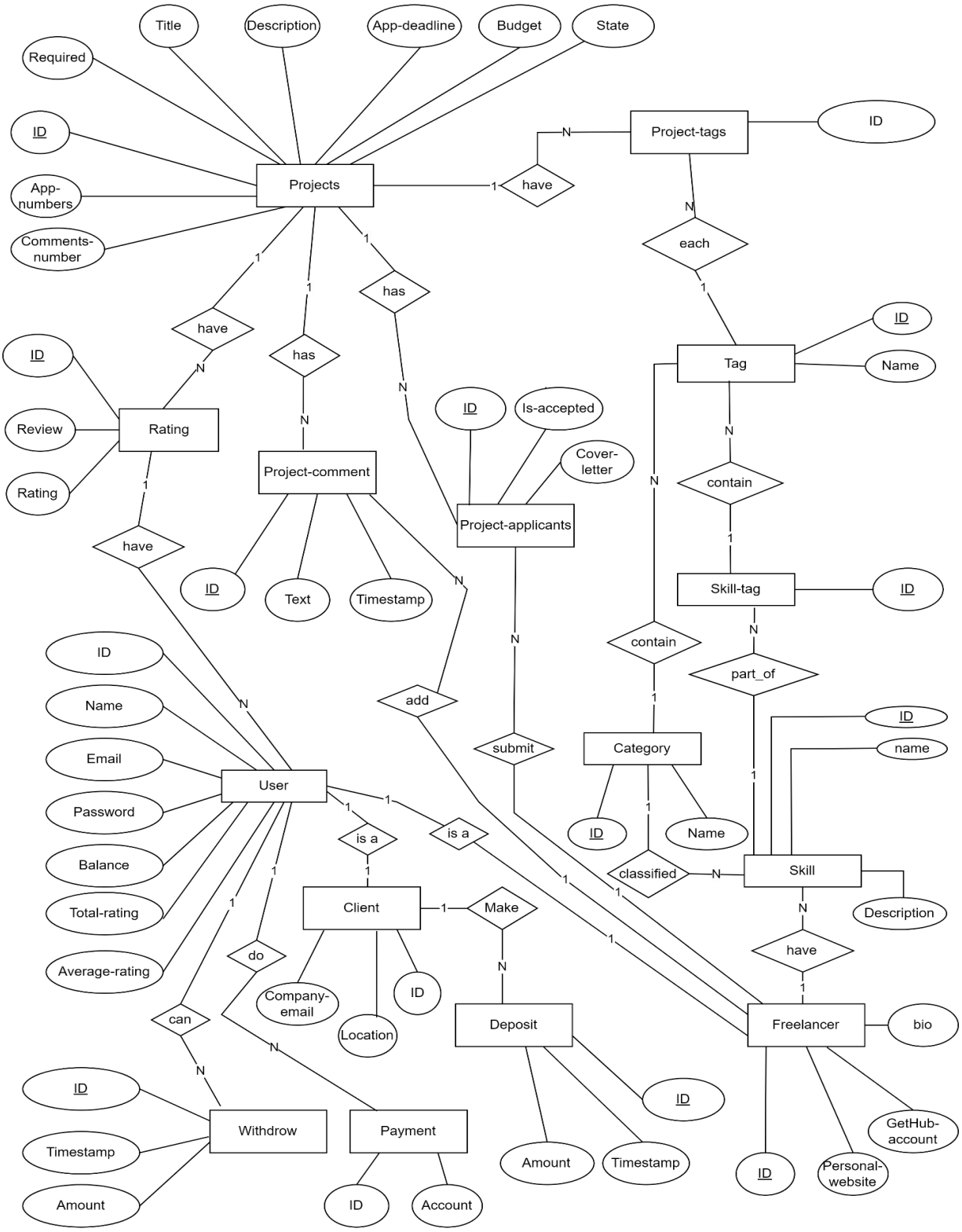
View payment history

Use-Case Name:	View payment history	
Use-Case ID:	View-payment-history-31	
Priority:	medium	
Primary Actor:	Client	
Description:	This use case allows the client to view their payment history on the platform.	
Precondition:	The client is registered and logged in.	
Trigger:	The client selects the "Payment History" option in their account settings.	
Typical Course of Events:	Actor Action	System Response
	- selects a payment from the list to view more details.	- displays a list of all payments made by the client. - displays the details of the selected payment, including the project it was for, the freelancer it was paid to, the amount paid, and the date of payment.
Alternate Courses:	If the client has not made any payments on the platform, the system displays a message indicating no payment history is available.	
Conclusion:	This use case allows the client to easily view their payment history and track their spending on the platform.	
Post-condition:	The client has reviewed their payment history and can exit the payment history or select another payment to view.	

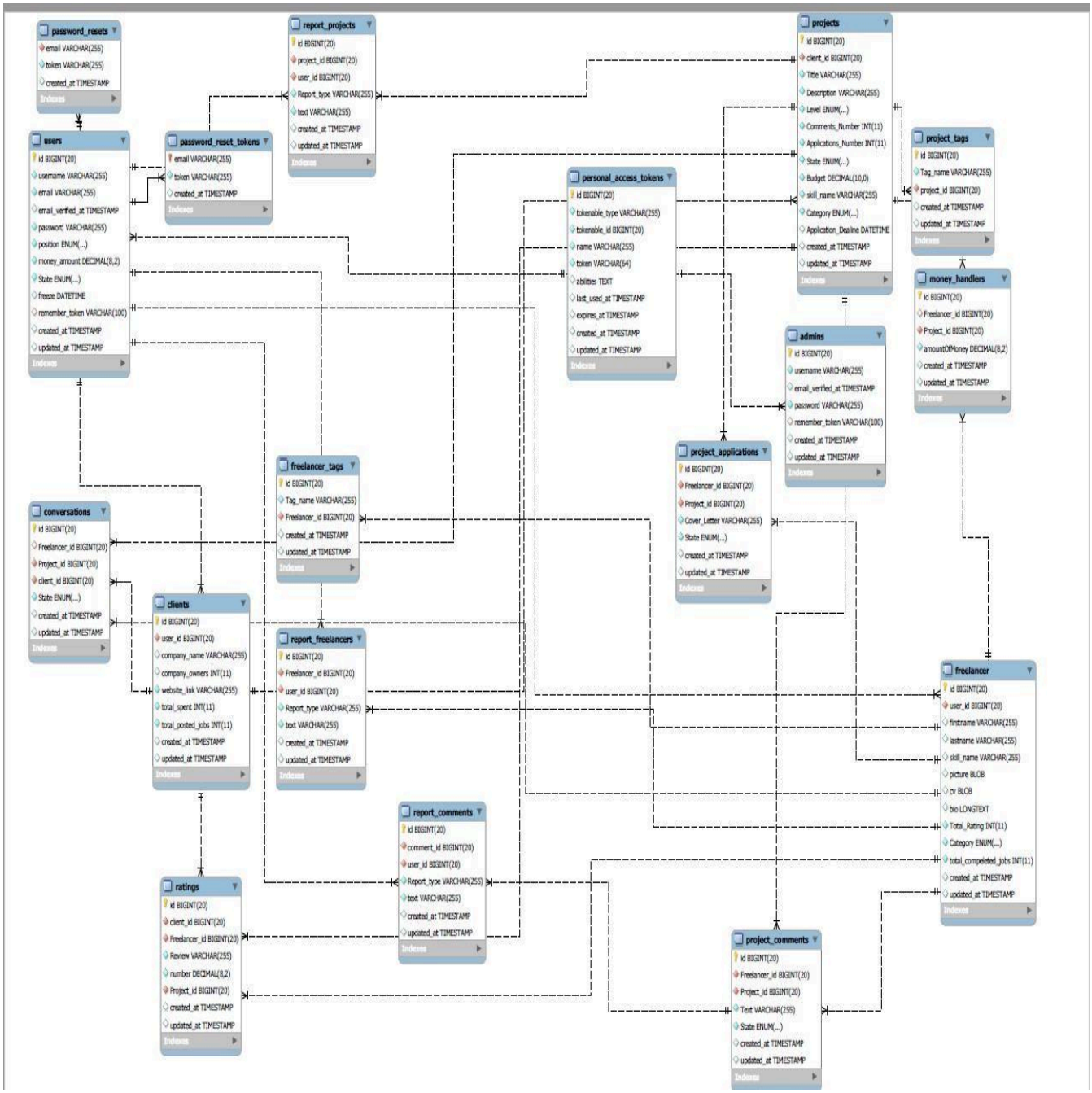
Deposit fund

Use-Case Name:	Deposit fund	
Use-Case ID:	Deposit-fund-32	
Priority:	medium	
Primary Actor:	Client	
Description:	This use case is used to allow the client to deposit fund to their account balance.	
Precondition:	The client is registered and logged in.	
Trigger:	The client chooses to deposit fund to the account.	
Typical Course of Events:	Actor Action	System Response
	-goes to the “Deposit Fund” section. -choose a payment method and enters the deposit amount. -confirms the payment.	-displays the available payment method. -verifies the payment and updates the client’s amount balance. -send a confirmation message to the client.
Alternate Courses:	If the payment method is invalid or confirmation fail, the system displays an error message.	
Conclusion:	This use case enables clients to easily deposit fund to their account balance.	
Post-condition:	The client’s account balance is updated with the deposited fund.	

ENTITY RELATION SHIP DIAGRAM

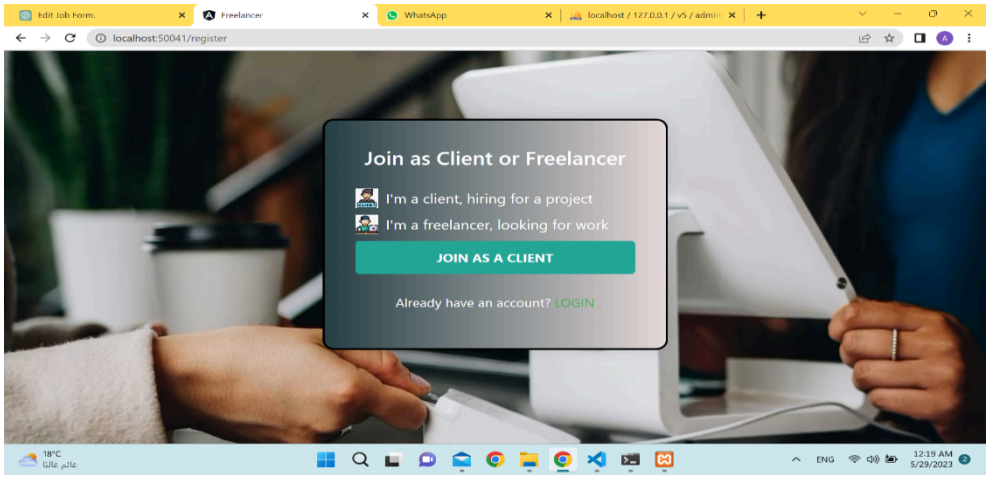

Title
Description
App-deadline
Budget
State
Required
ID
App-numbers
Comments-number
Projects
have
Project-tags
ID
each
has
have
has
ID
Review
Rating
Rating
Project-comment
ID
Text
Timestamp
ID
Is-accepted
Cover-letter
Project-applicants
Tag
ID
Name
contain
Skill-tag
ID
part_of
have
add
submit
contain
Category
ID
Name
ID
name
Skill
Description
classified
ID
Name
Email
Password
Balance
Total-rating
Average-rating
User
is a
is a
Client
Make
do
Company-email
ID
Location
Deposit
Freelancer
bio
have
can
Withdrow
Payment
ID
Timestamp
Amount
ID
Account
ID
Amount
Timestamp
ID
GetHub-account
Personal-website

CLASS DIAGRAM

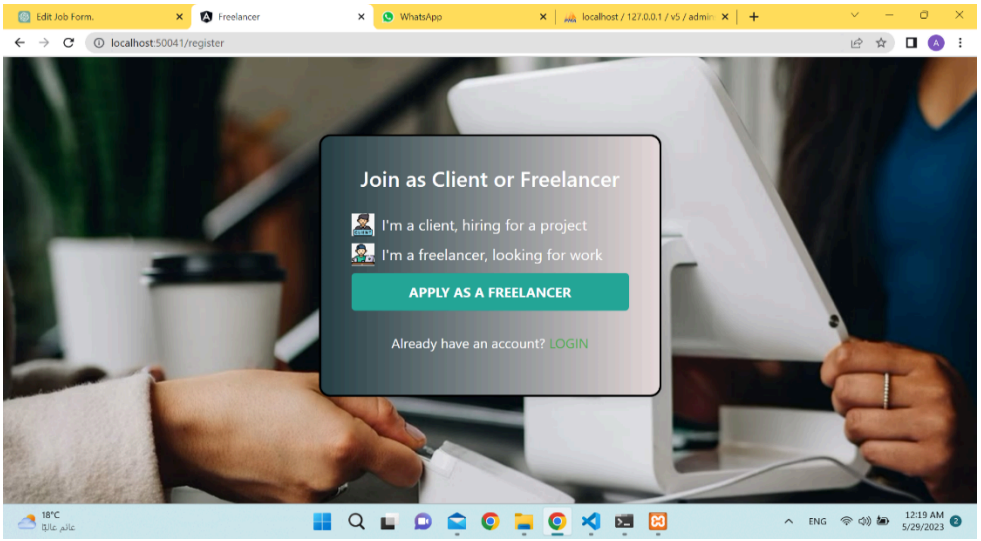


IMAGES FROM PROJECT:

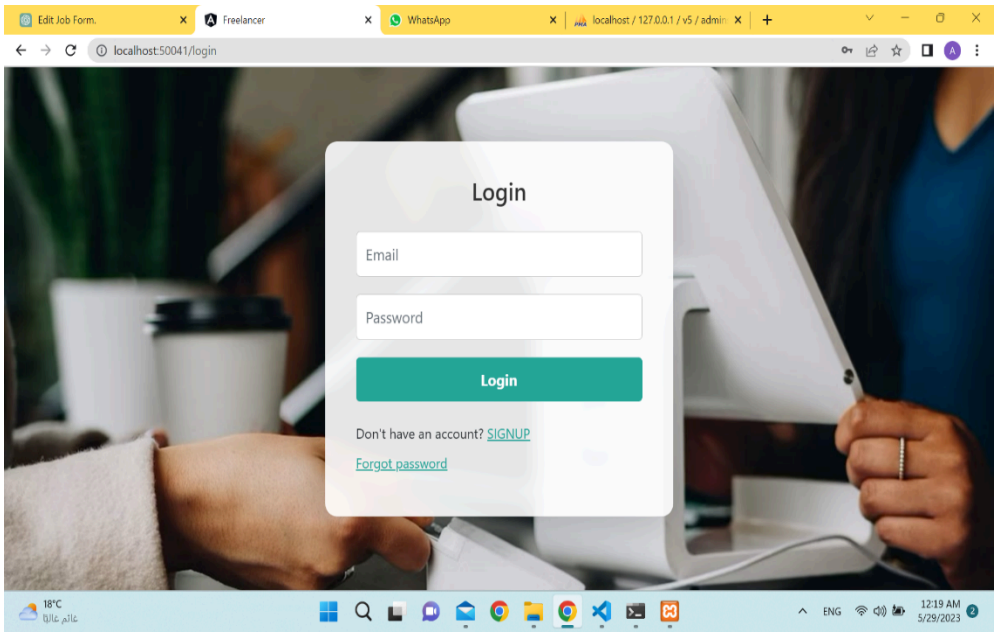
Create account as client:



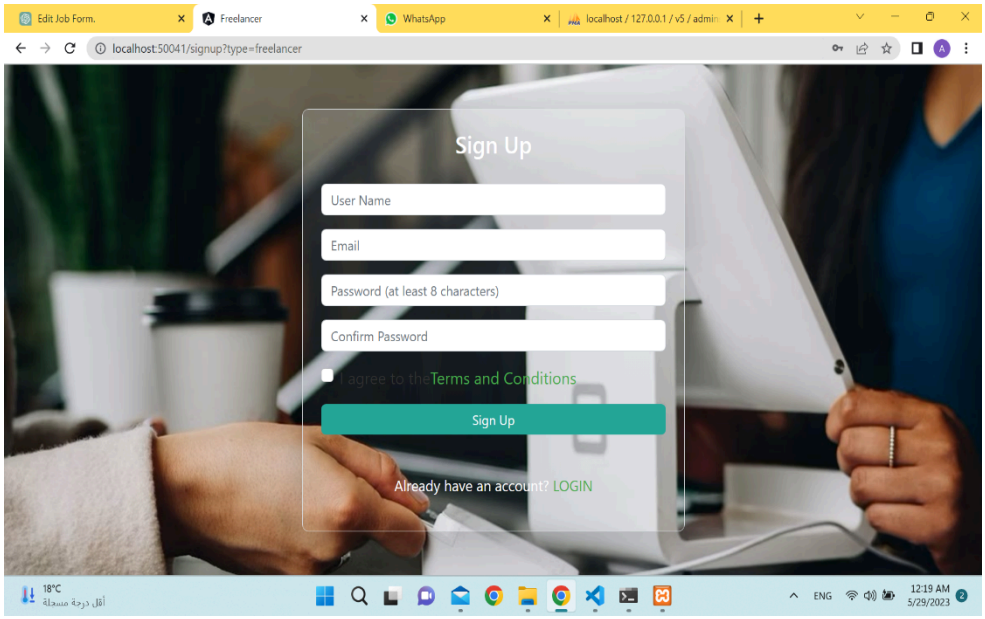
Create account as freelancer page:



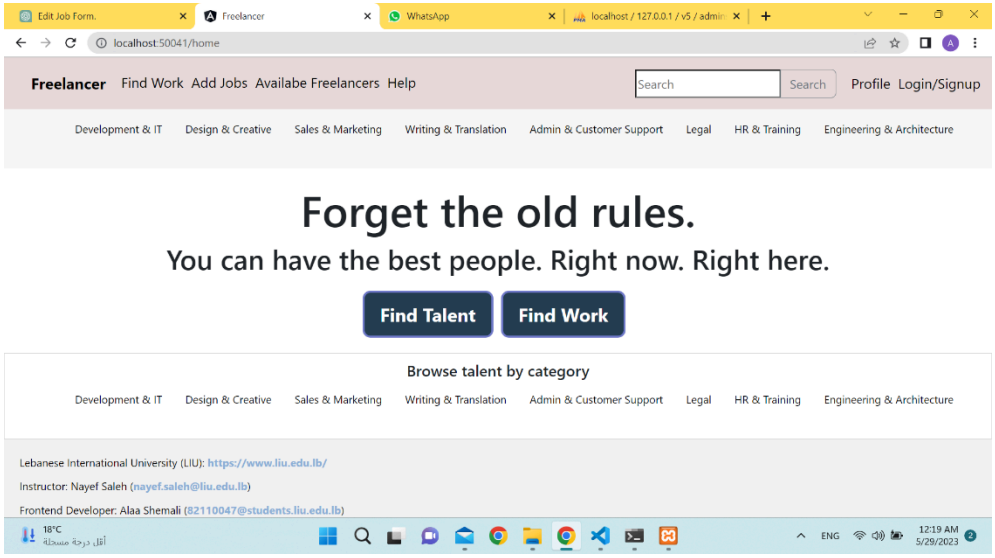
Login page:



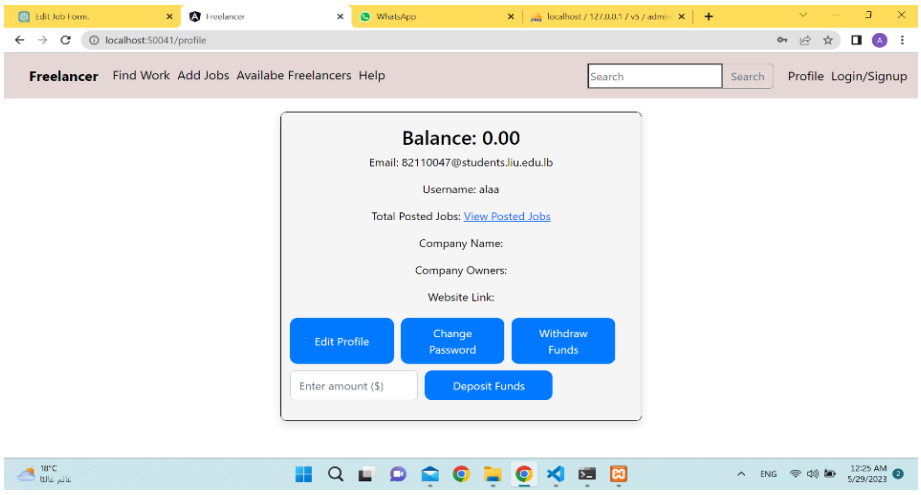
Signup page:



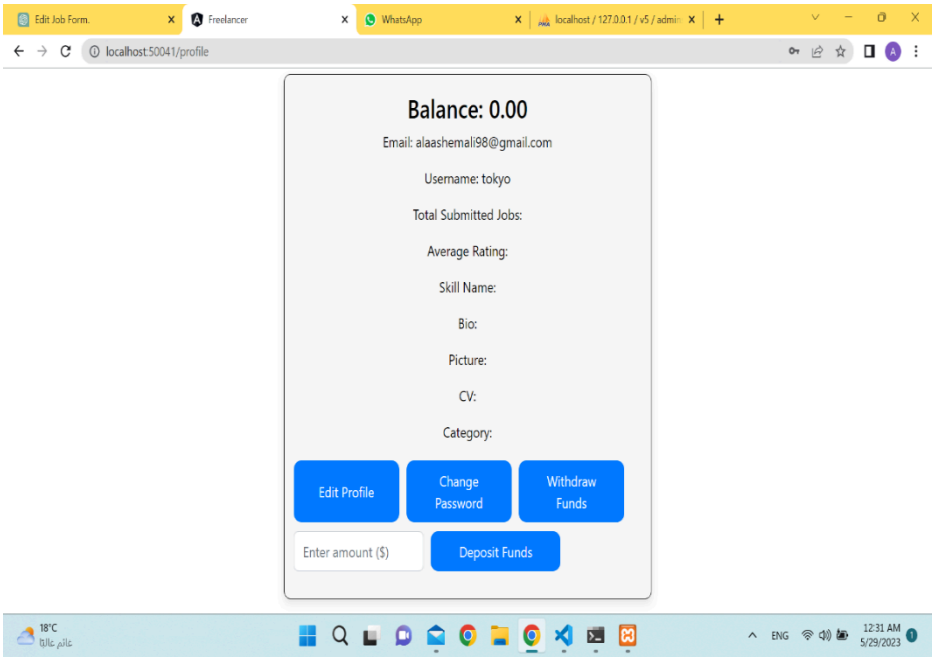
Home page: This is the page where the user can access everything in the project. In the navbar we have insert a navigation to the main pages.



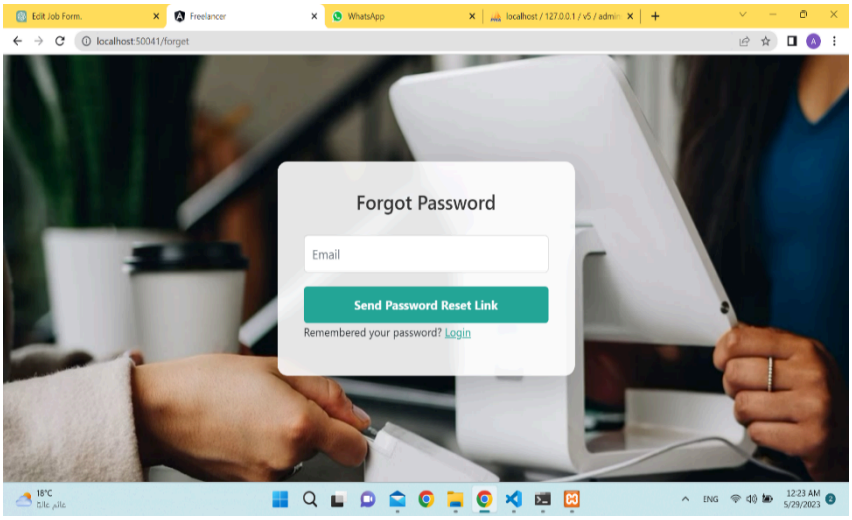
Client profile page:



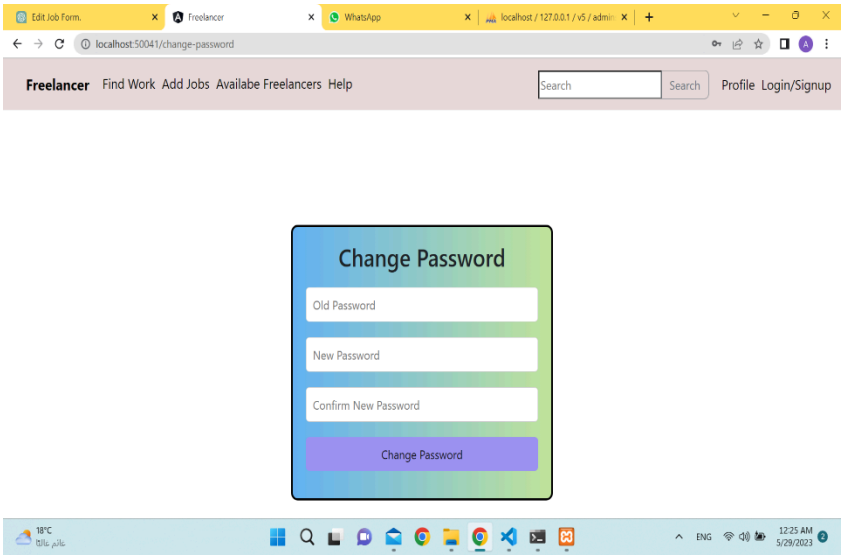
Freelancer profile page:



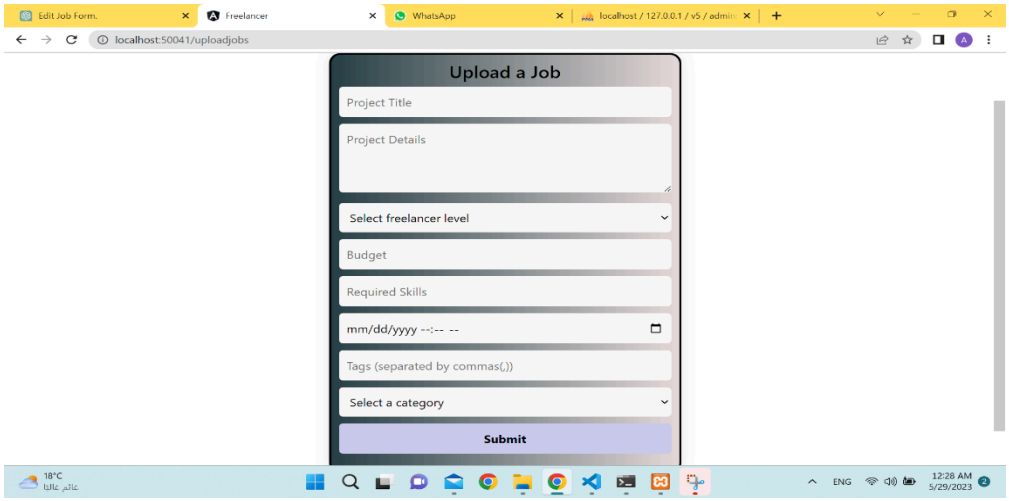
Forget password page: This is the page where the user can the password in case of forget it. Just he must fill the email. If the email valid he receives an email to reset the password.



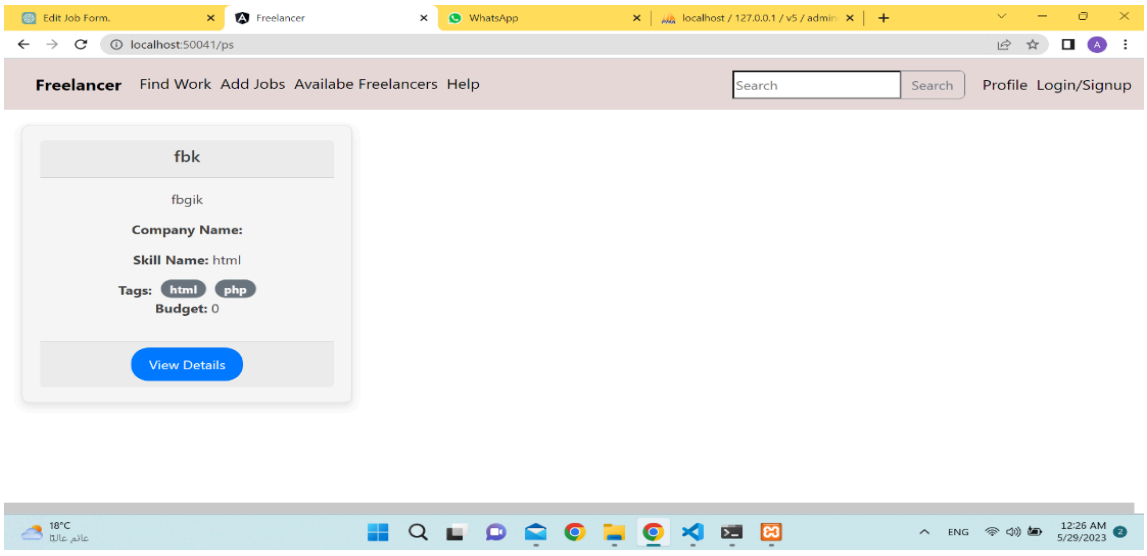
Change password page: this is the page where the user can access it from the profile page.



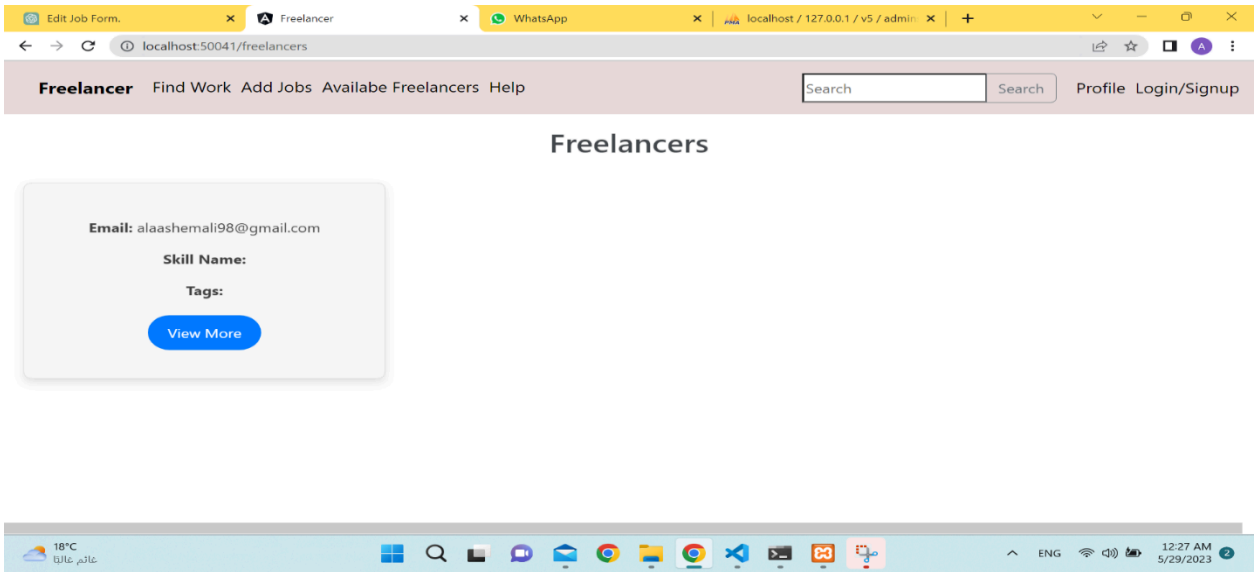
Post new project. This is the page where the client can post new project.



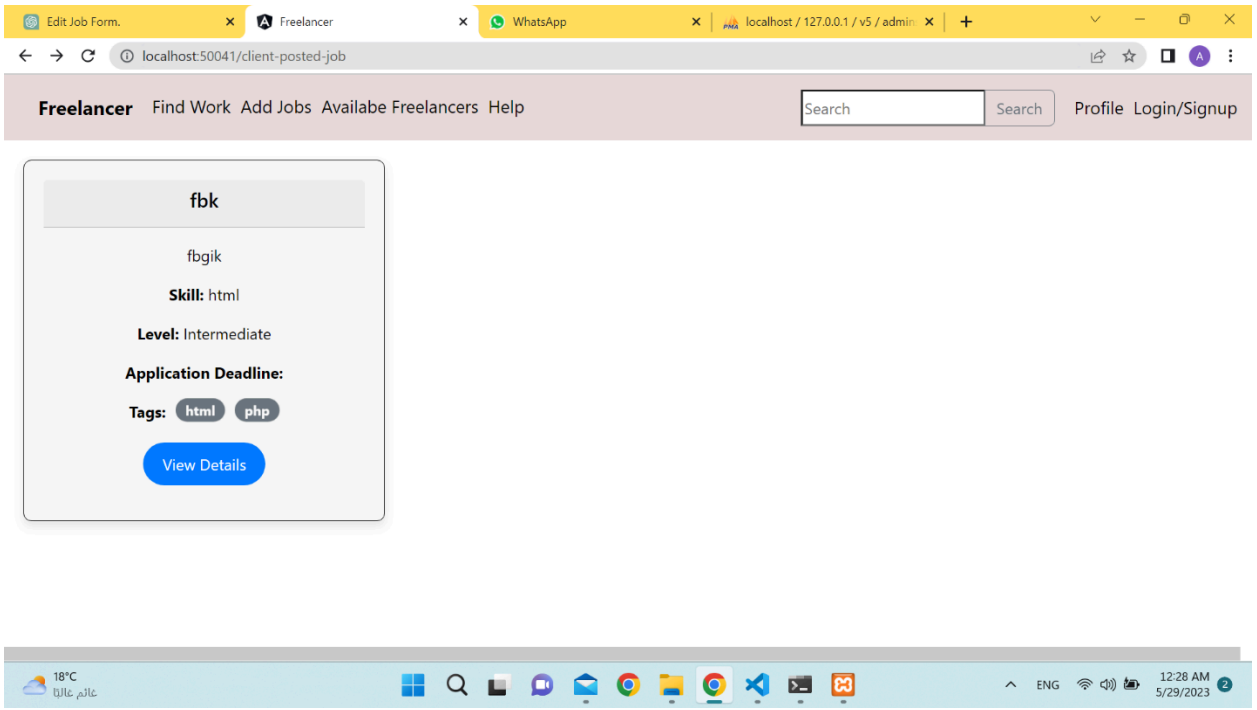
PROJECT LIST. This is the page where the project apper for the freelancer.



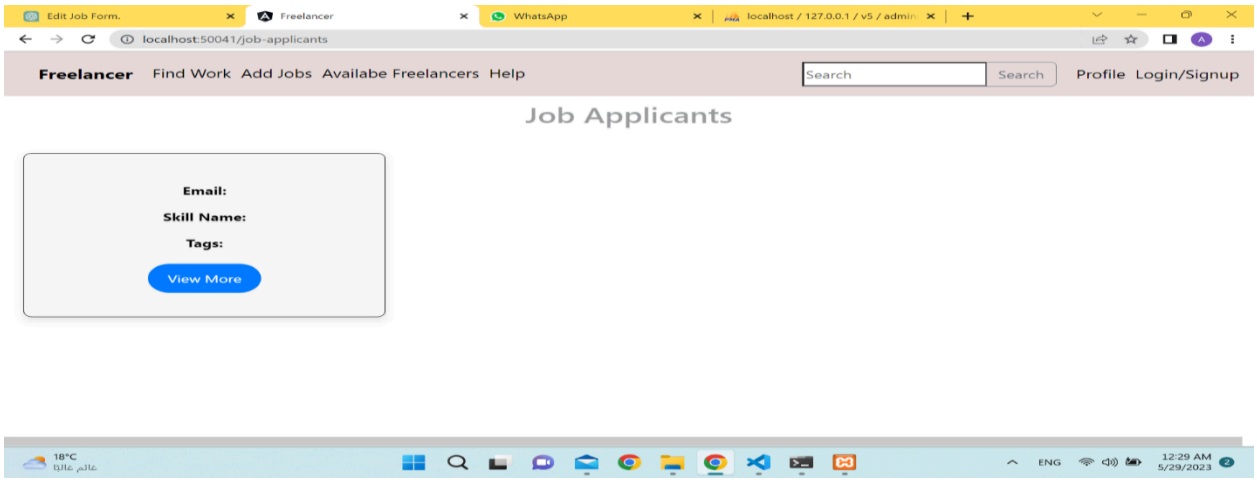
Available freelancers. This is the page that contain all the available freelancers.



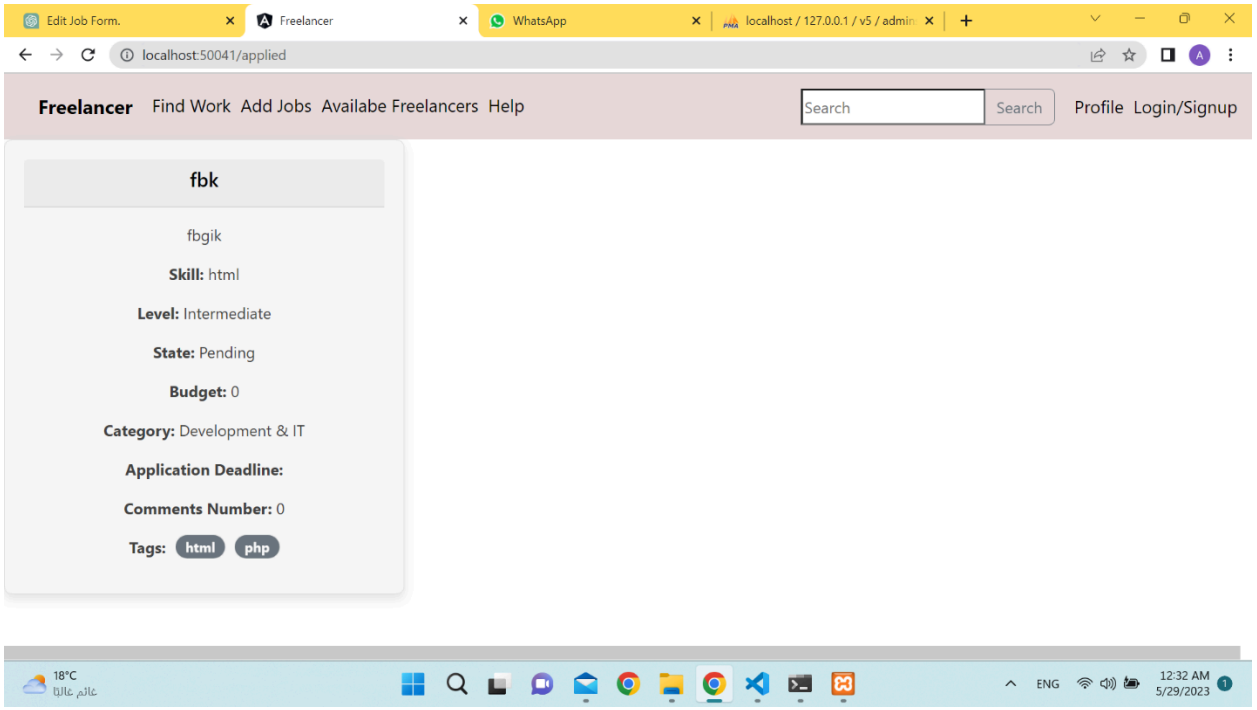
CLIENT POSTED JOB. This is the page where the client can see the projects, he has post.



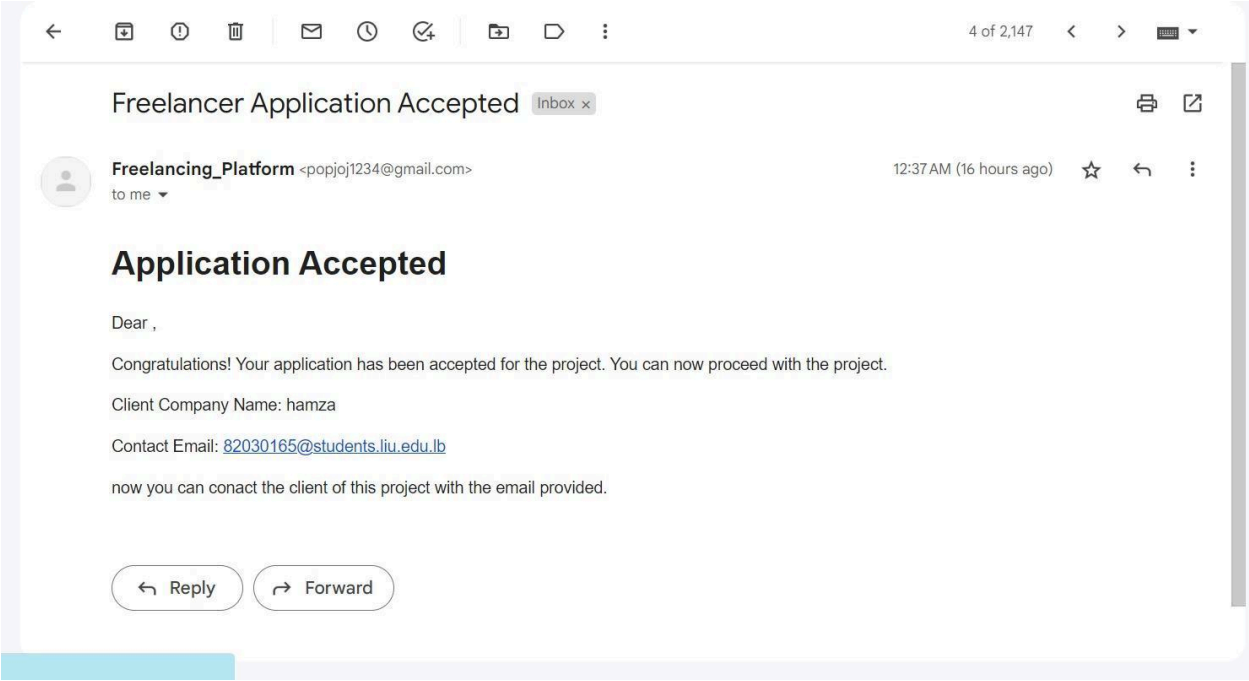
Freelancer applicants. This is the page where the client can see the freelancer applicants who apply for the project, he has post.



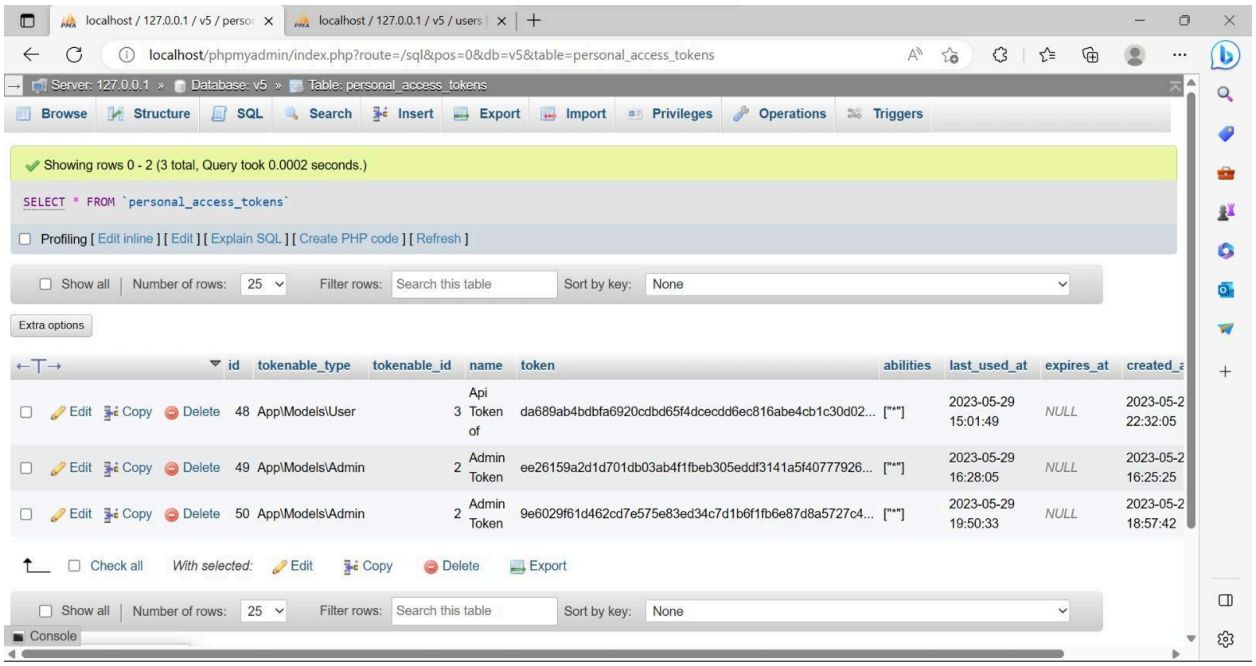
FREELANCER APPLIED PROJECTS. This is the page where the freelancer can see the projects he has applied for.



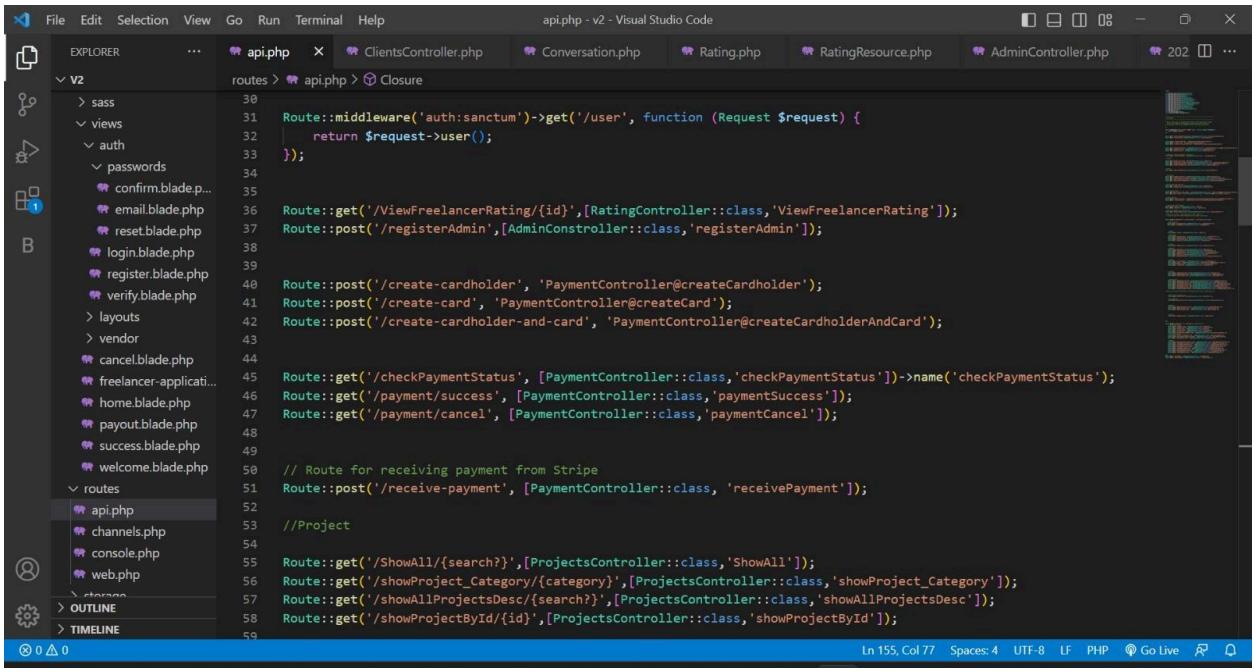
APPLICATION ACCEPTED: when the client accept freelancer's application



Personal_access_tokens database table: where all the tokens will be stored of each of the two types "admins and users" when a user or an admin login the server generates a token to a specific user or admin and hash it in this table knowing that I can modify the expiration duration as I want and modify every scope of permissions for every user or admin according to their tokens.



Users middleware and some of its routes I can't include all of the routes because they are 72 API.

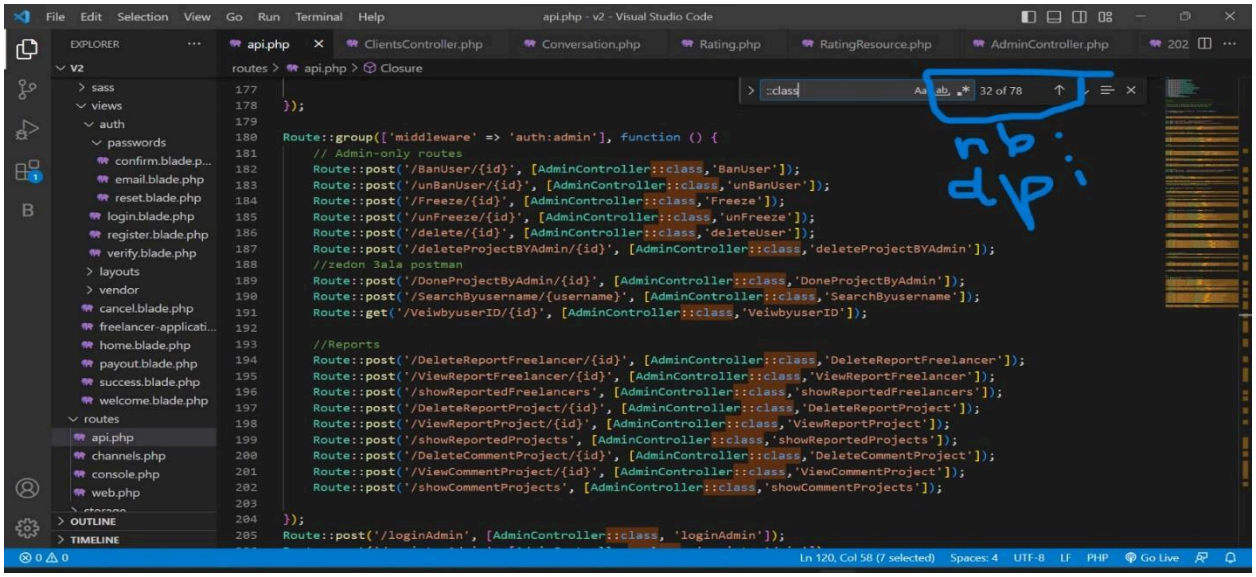


Admin route: these are the routes of the admin knowing that the middleware is like firewall that prevents the access of any route inside it unless you have its specific sanctum token signature.

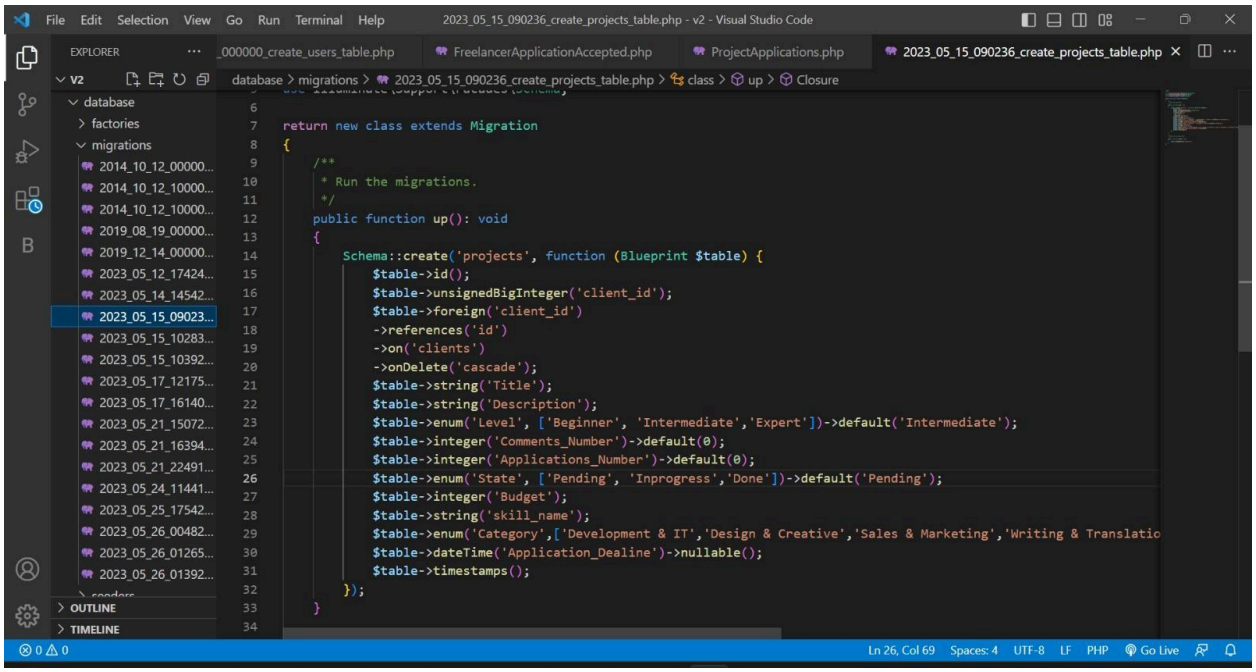
sanctum token: it a token of barrier type that used for authentication knowing that it's a built in Laravel.

and these routes are belonging only for the admins.

Knowing that I have done two separated authentication systems in this project one for the admins and other for the users.



database migrations



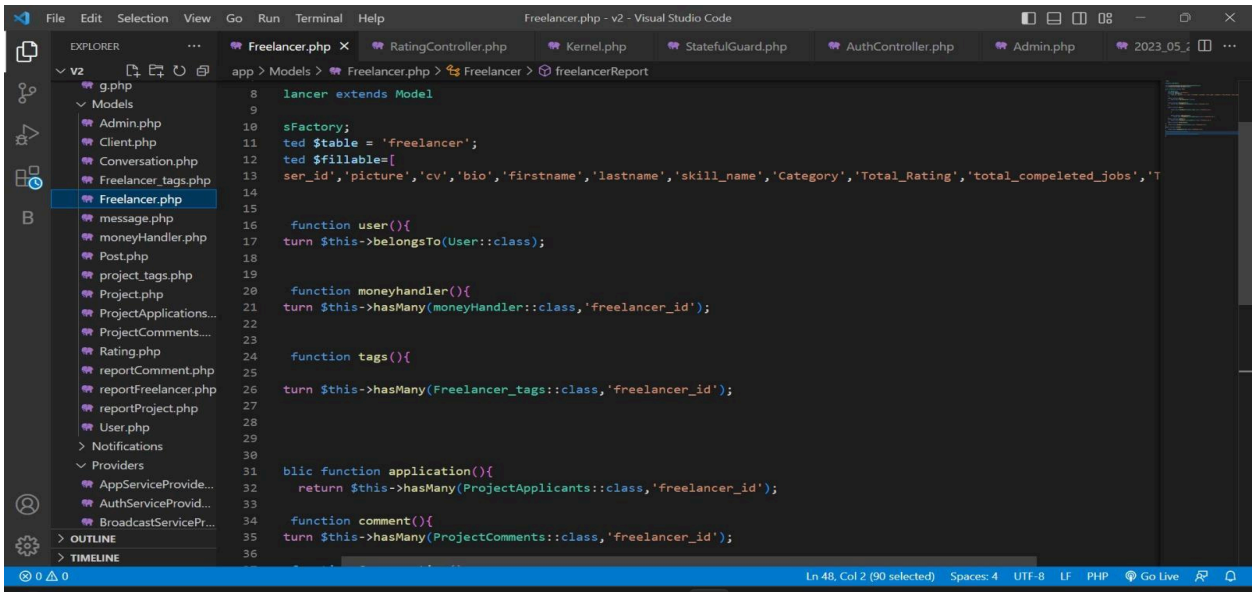
models:

freelancer:

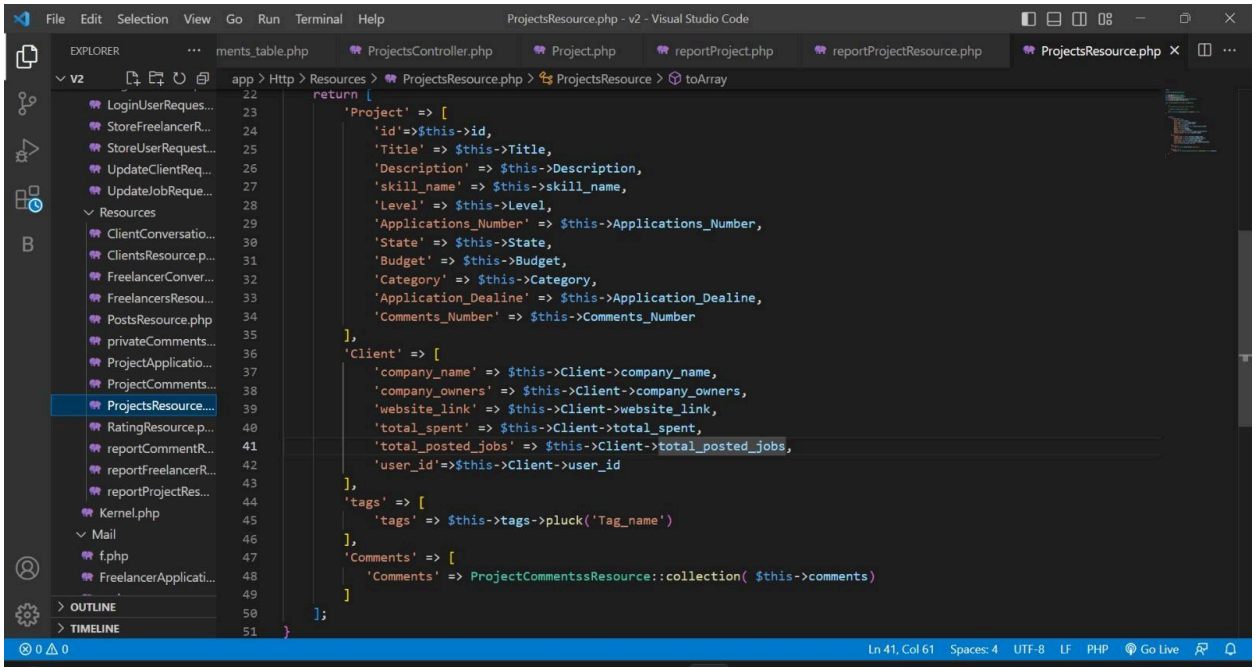
indicates what only filled are allowed to edit them inside the database and make the relations between the database table in it.

I have used eloquent for this.

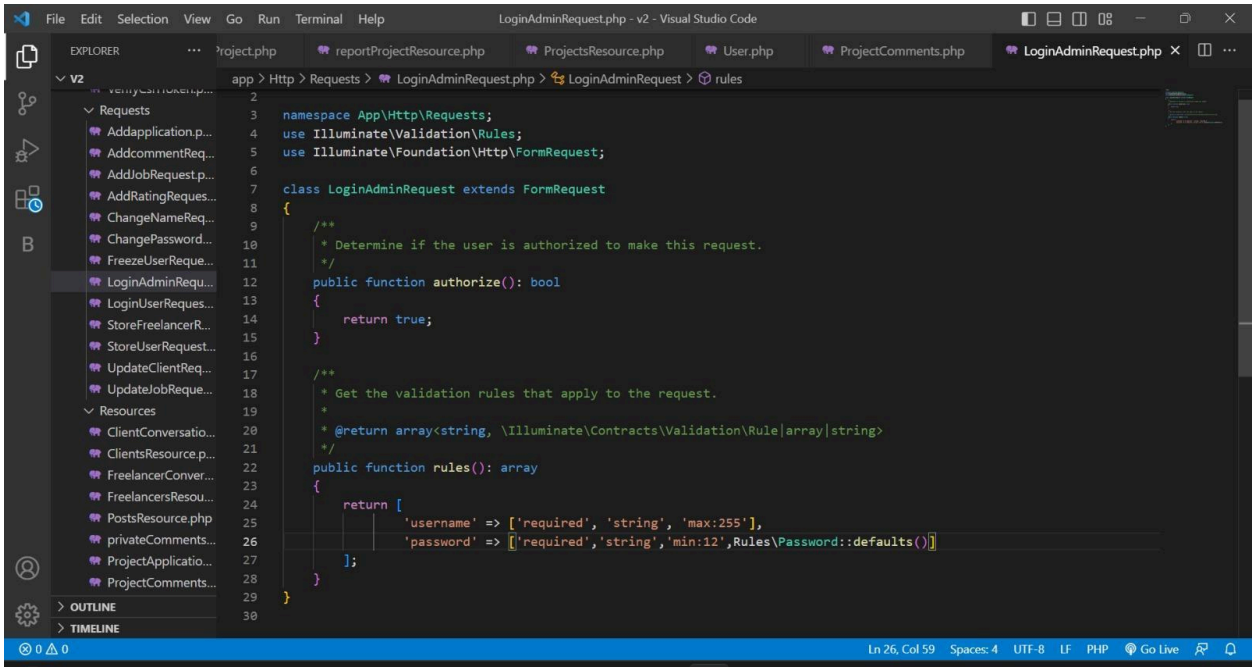
eloquent: an object-relational mapper (ORM) that makes it enjoyable to interact with your database.



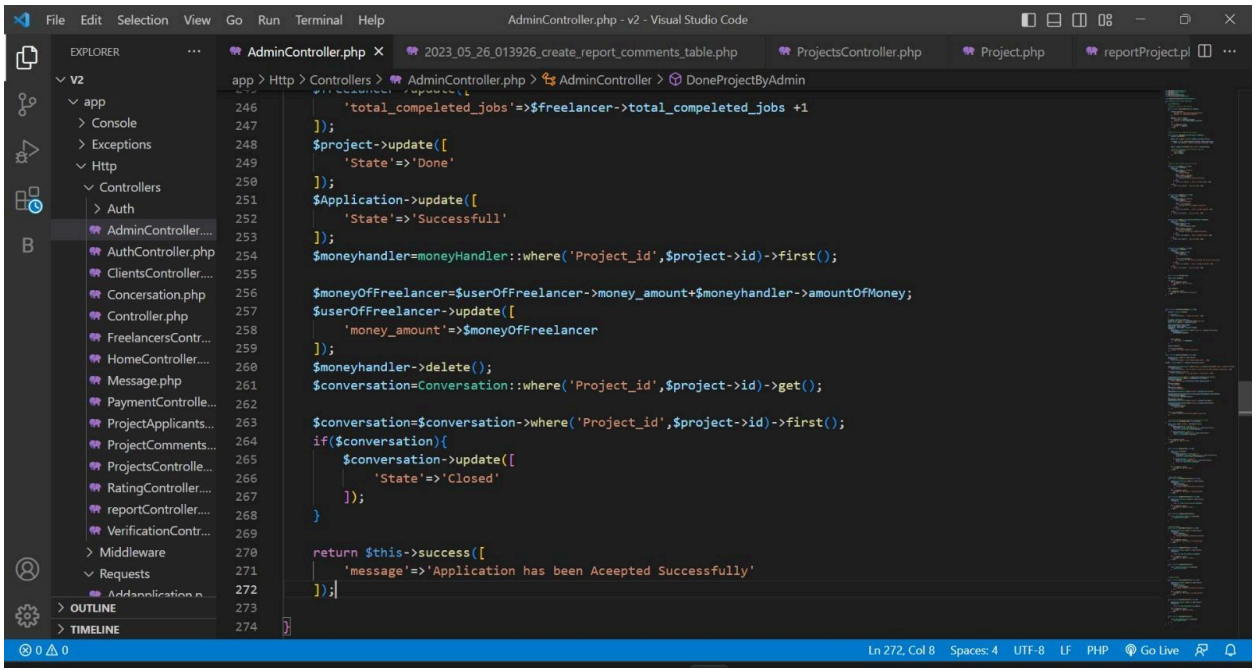
Project resources to indicates the content of the json array that will be sent in each API.



Login admin request: requests is in which I indicate what attributes should receive and their types these are made for security to prevents injection to the API as a requests.

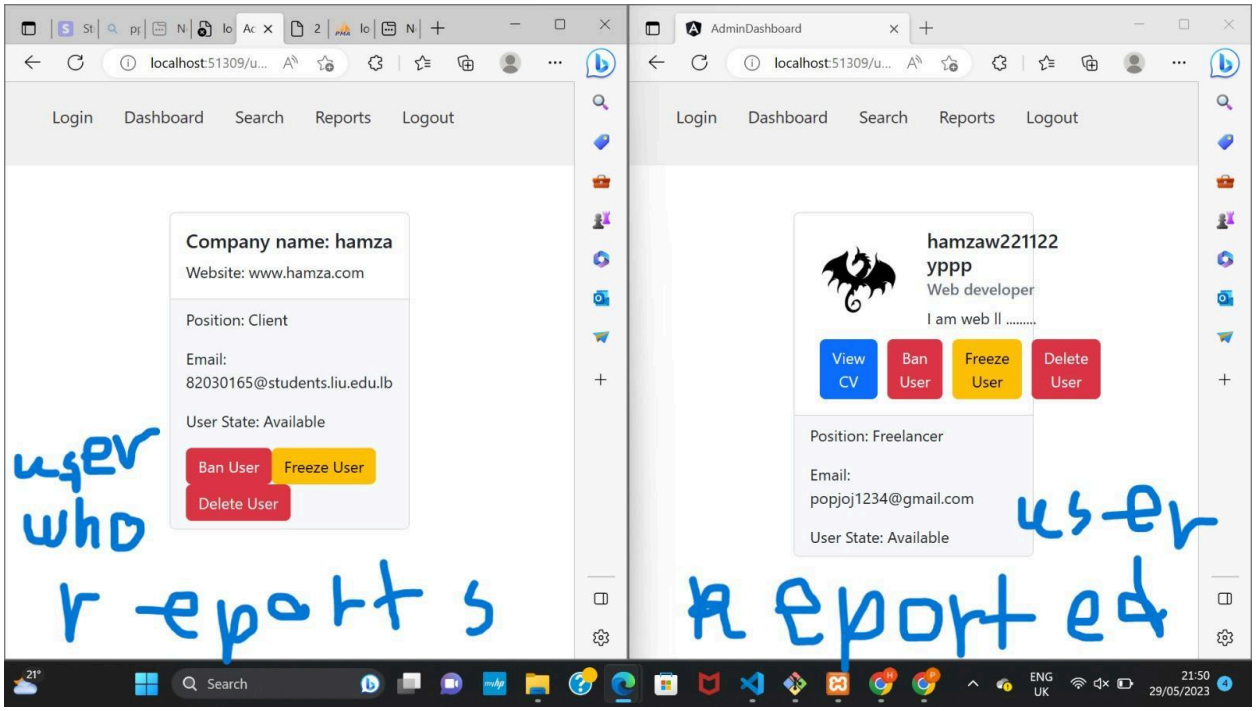


Admin Controllers:



Report and reported user: these are the buttons of the user who reports and user reported.

in the left is the user who reports the freelancer and on the right is the reported freelancer.



view project button:

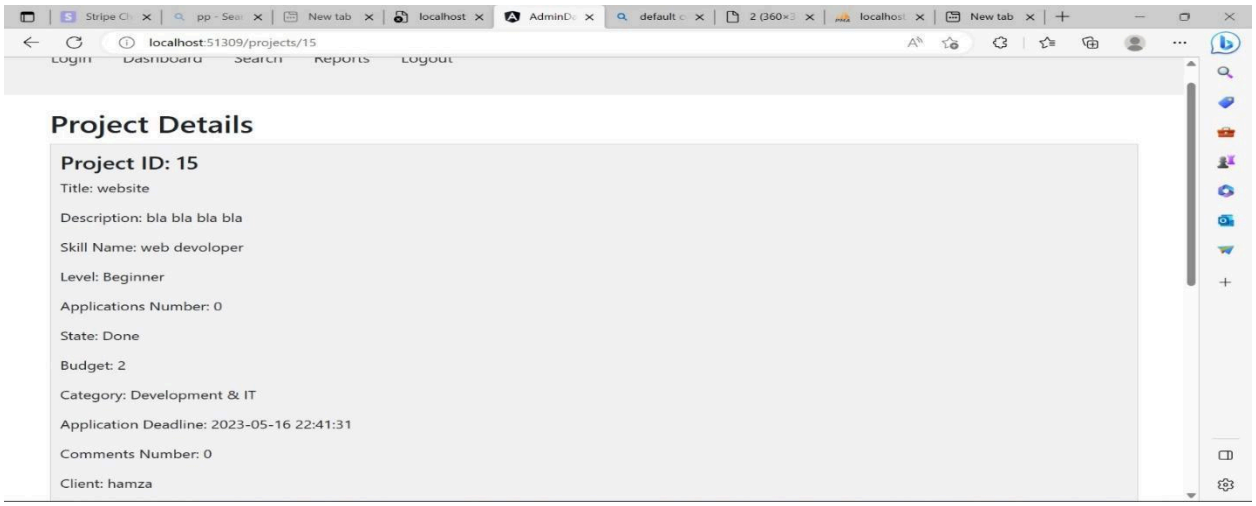
1-showing all project and client of the project details.

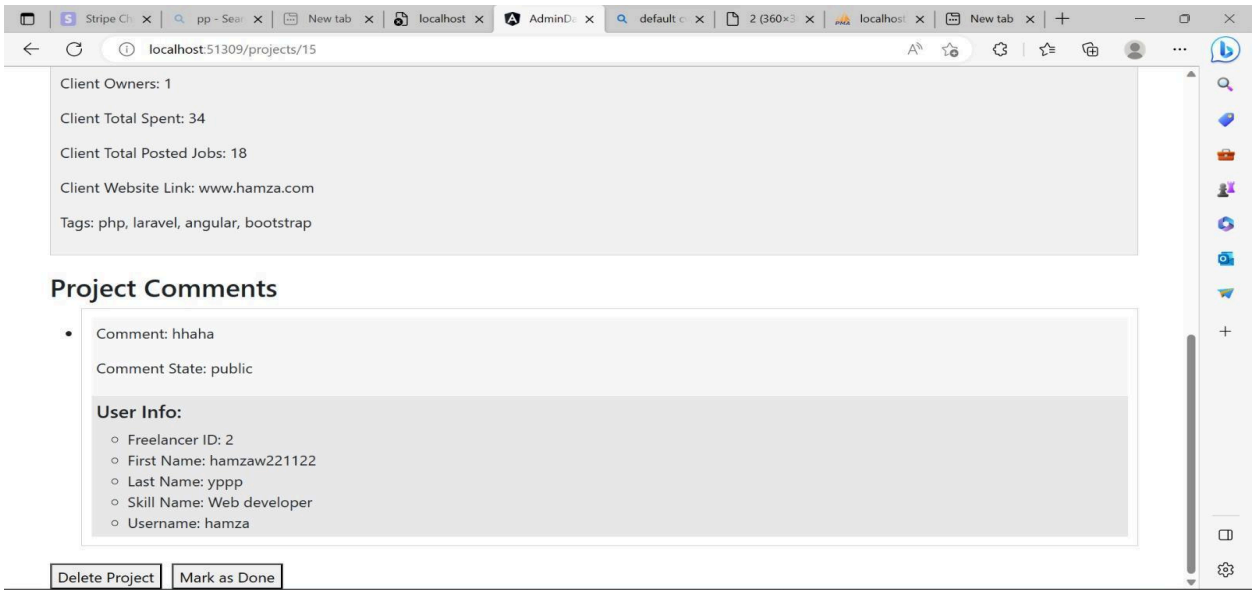
2-showing all comments of this project with some of the freelancer info of this comments.

3-with delete project in case if the client is reports that his freelancer is not doing his job well in this case the project will be deleted and the money will be automatically returning from the money handler table to the client account.

4-in the opposite if the freelancer reports that he make all the work and the client is not making the project as done the admin will do it as done by force.

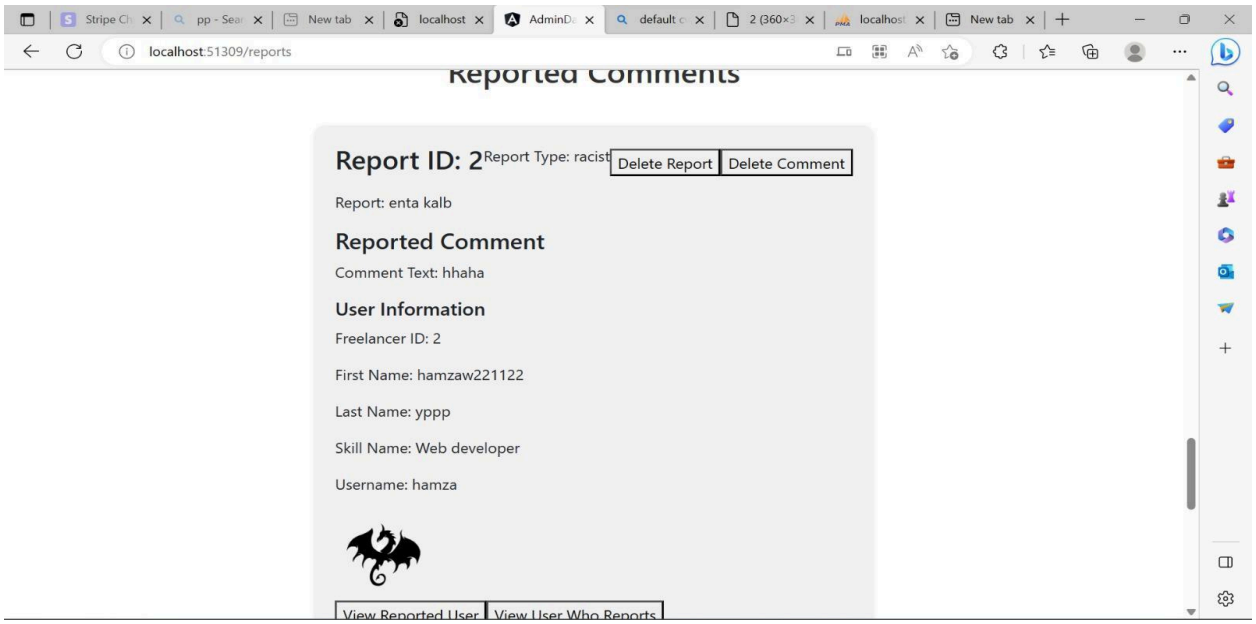
Simply the admin chooses what choice he will do according to the evidence he has.



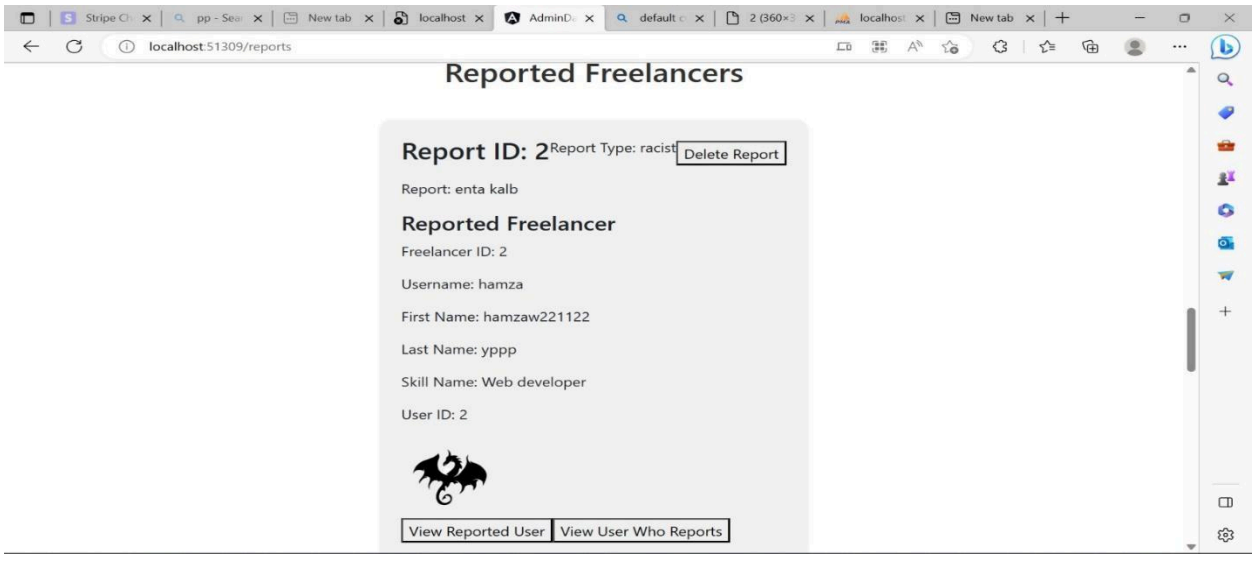


report section :

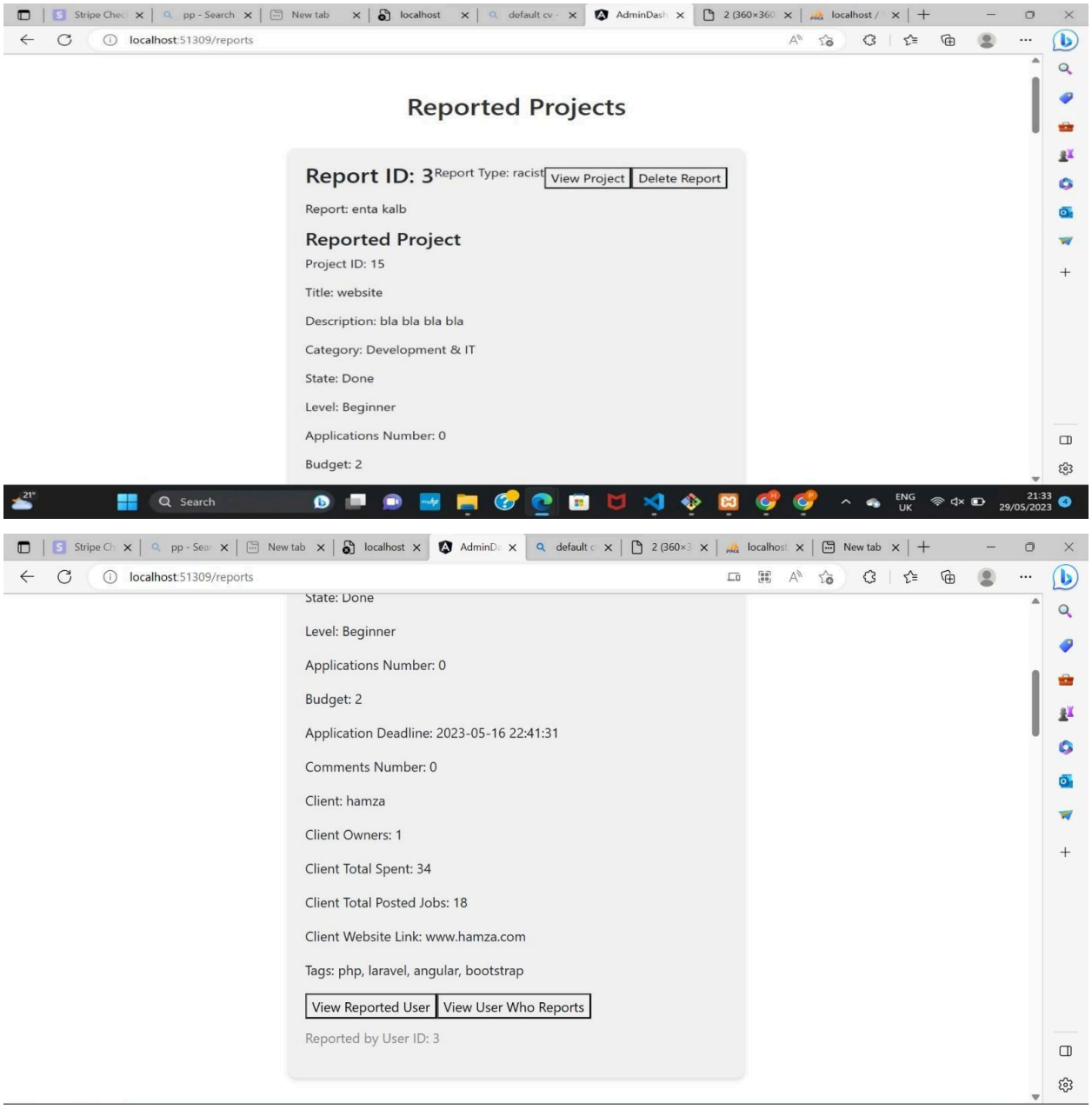
comments reports



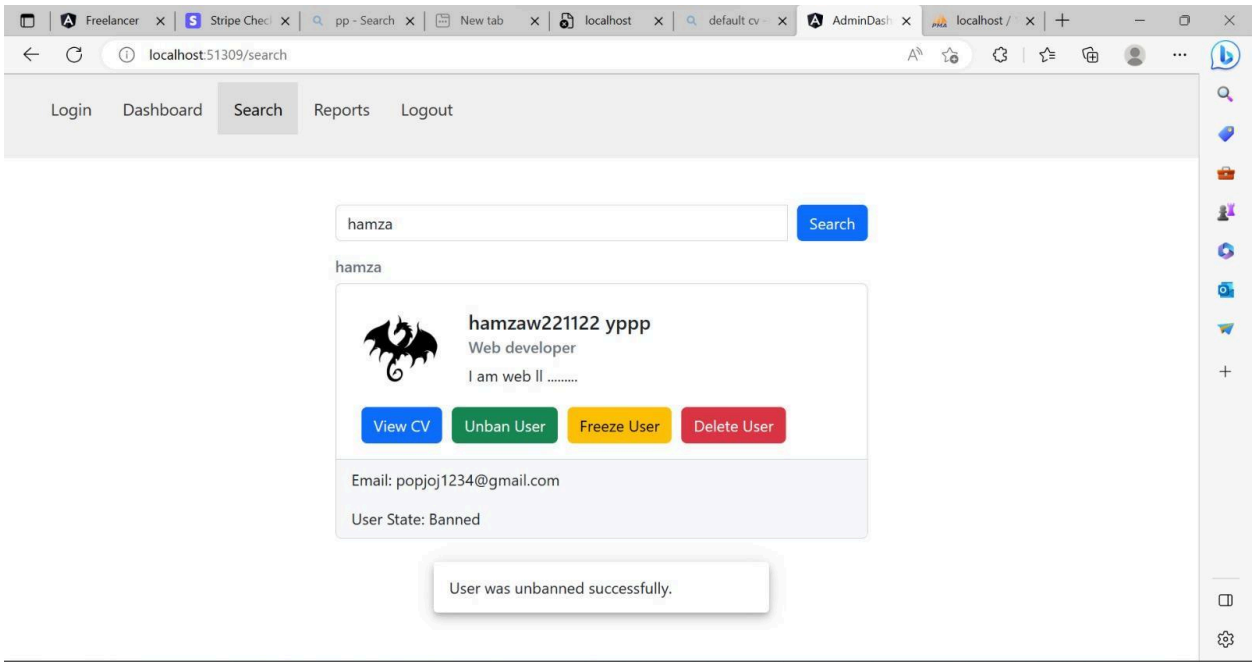
Freelancer reports



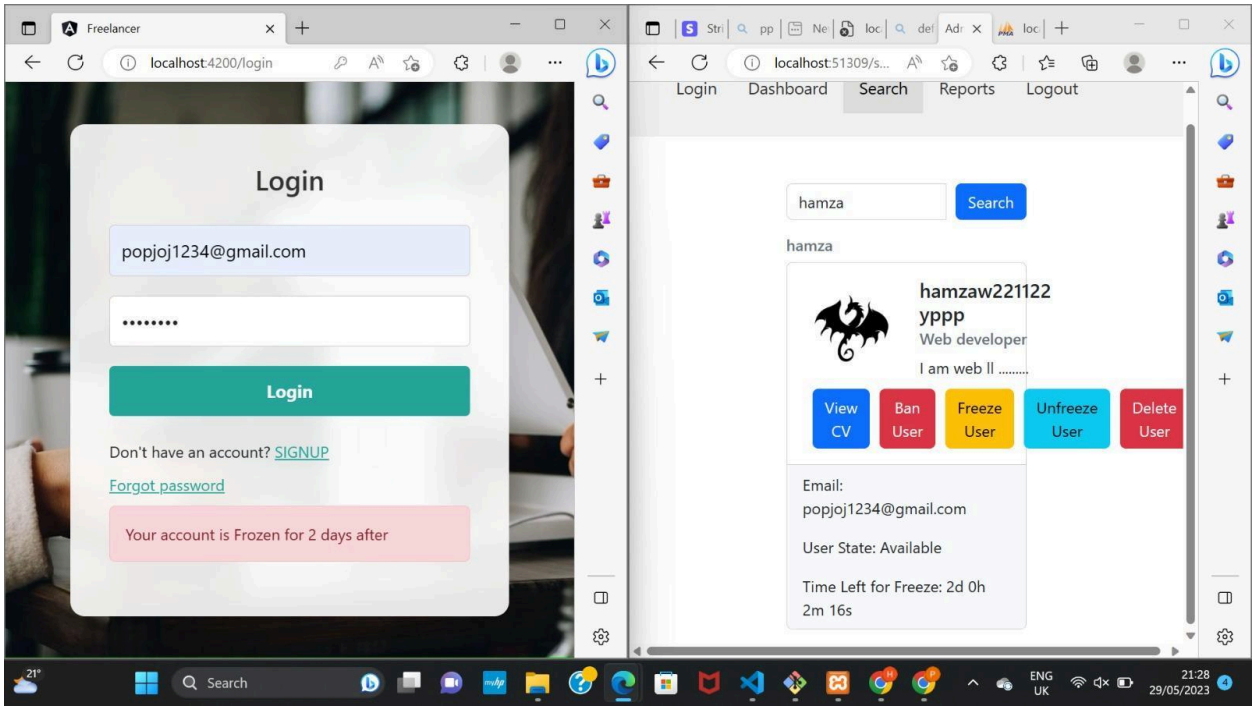
project reports



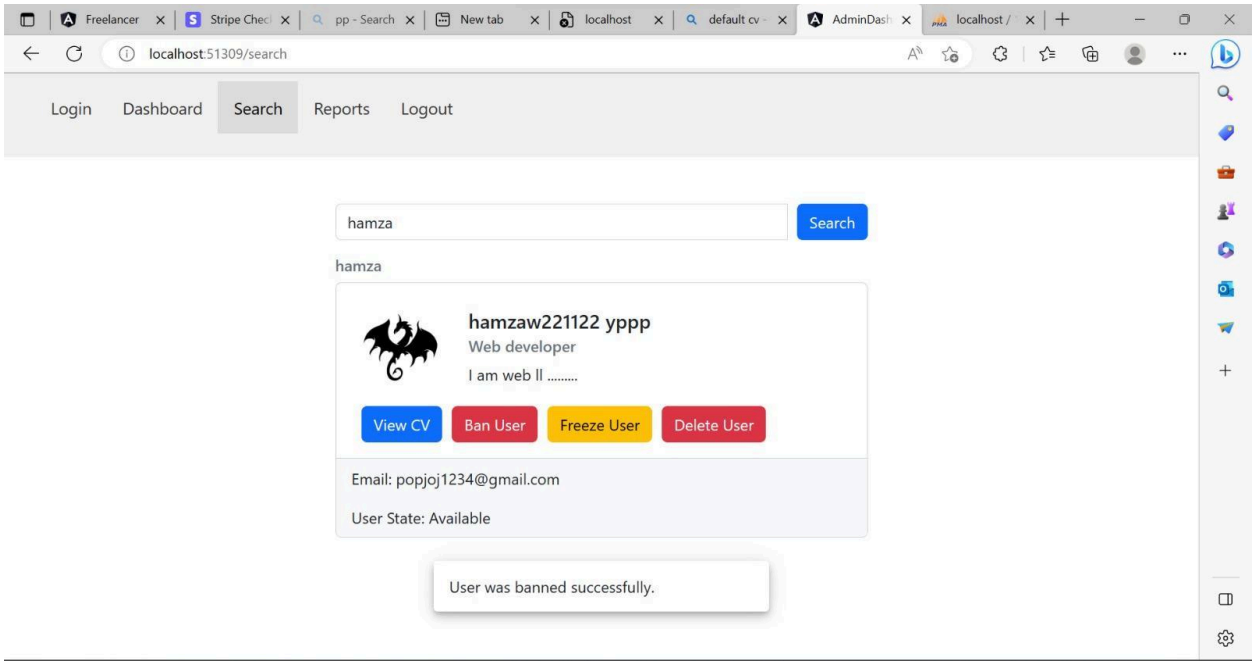
MANAGE USER:

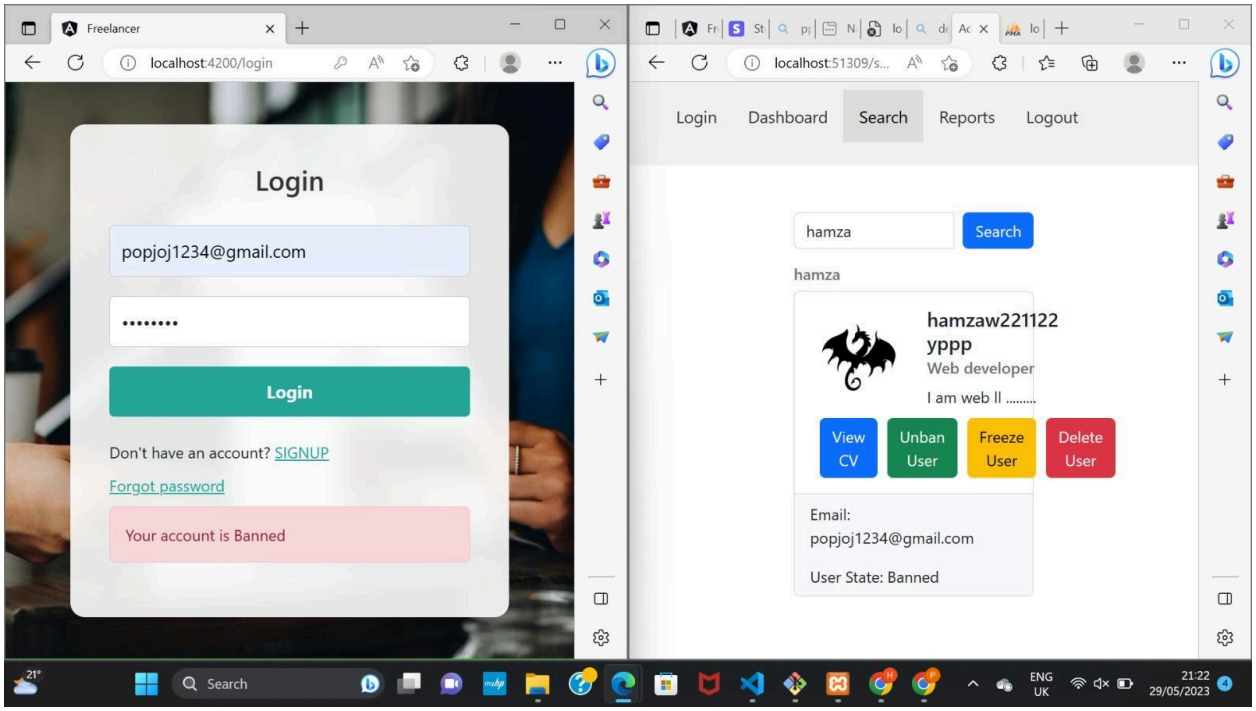

Freelancer
Stripe Chec
pp - Search
New tab
localhost
default cv
AdminDash
localhost /
localhost:51309/search
Login
Dashboard
Search
Reports
Logout
hamza
Search
hamza
hamzaw221122 yppp
Web developer
I am web II .....
View CV
Unban User
Freeze User
Delete User
Email: popjoj1234@gmail.com
User State: Banned
User was unbanned successfully.

FROZEN USER: the user frozen time left appears in both user login and admin panel knowing that the admin can increasethe time pf user's freeze if he want.

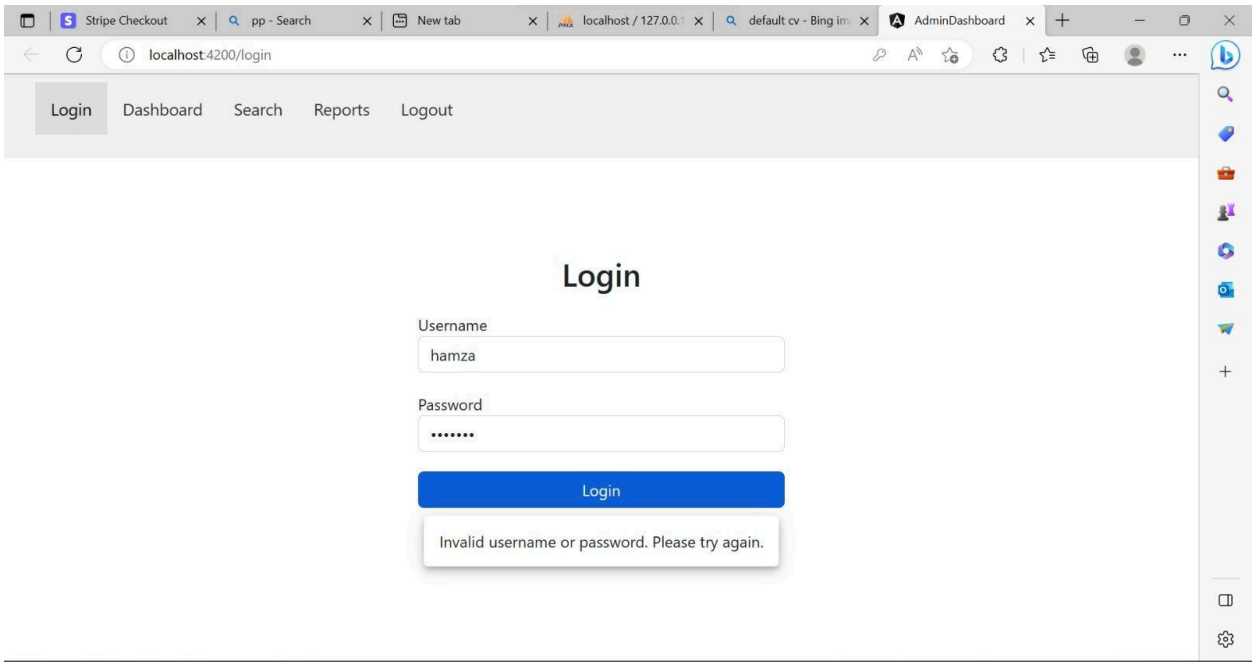


ban a user: knowing that user can be client or freelancer

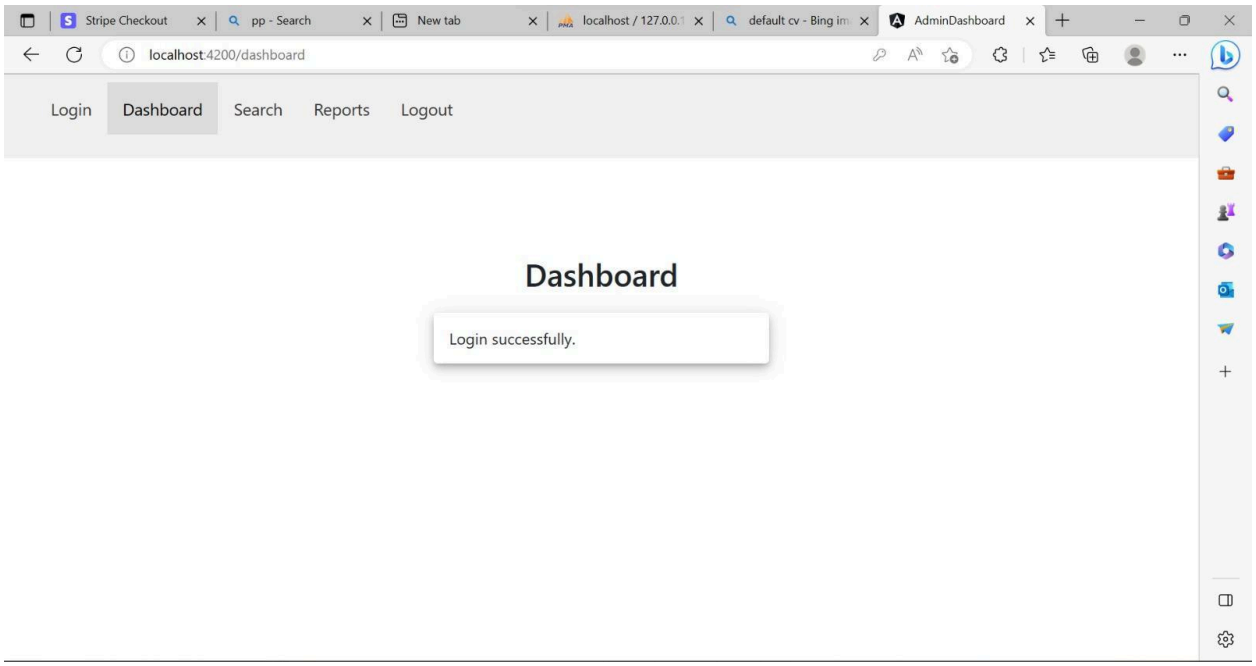




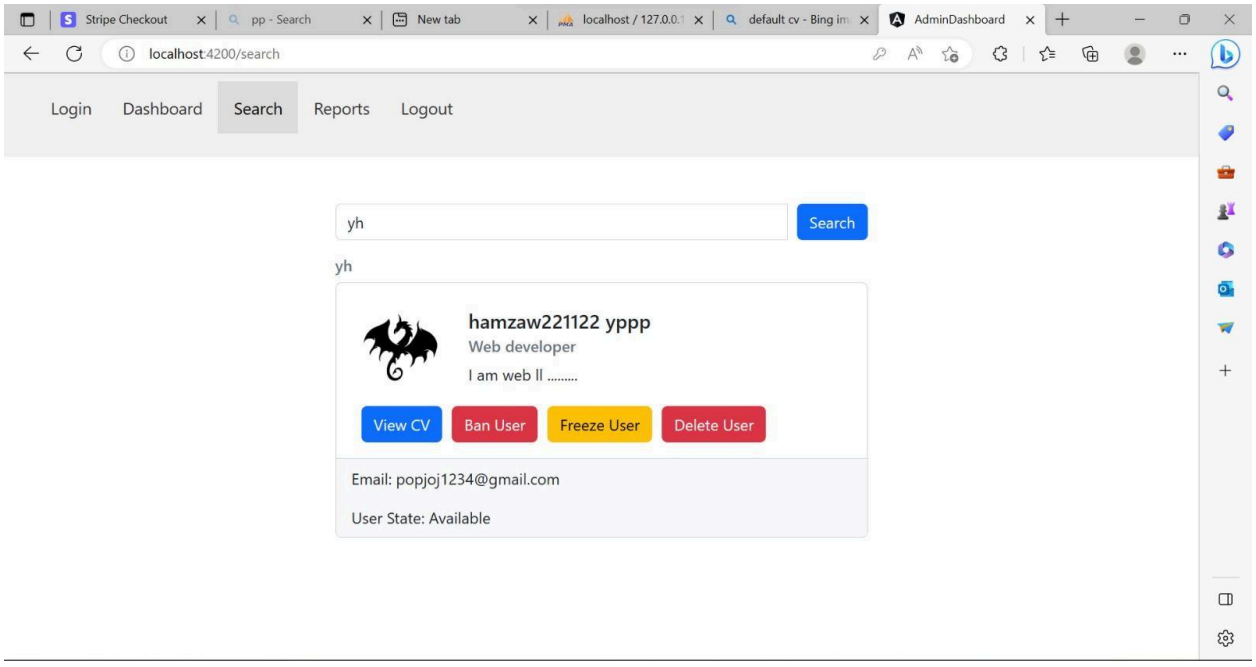
ADMIN LOGIN


Stripe Checkout
pp - Search
New tab
localhost / 127.0.0.1
default cv - Bing im
AdminDashboard
localhost:4200/login
Login
Dashboard
Search
Reports
Logout
Login
Username
hamza
Password
Login
Invalid username or password. Please try again.

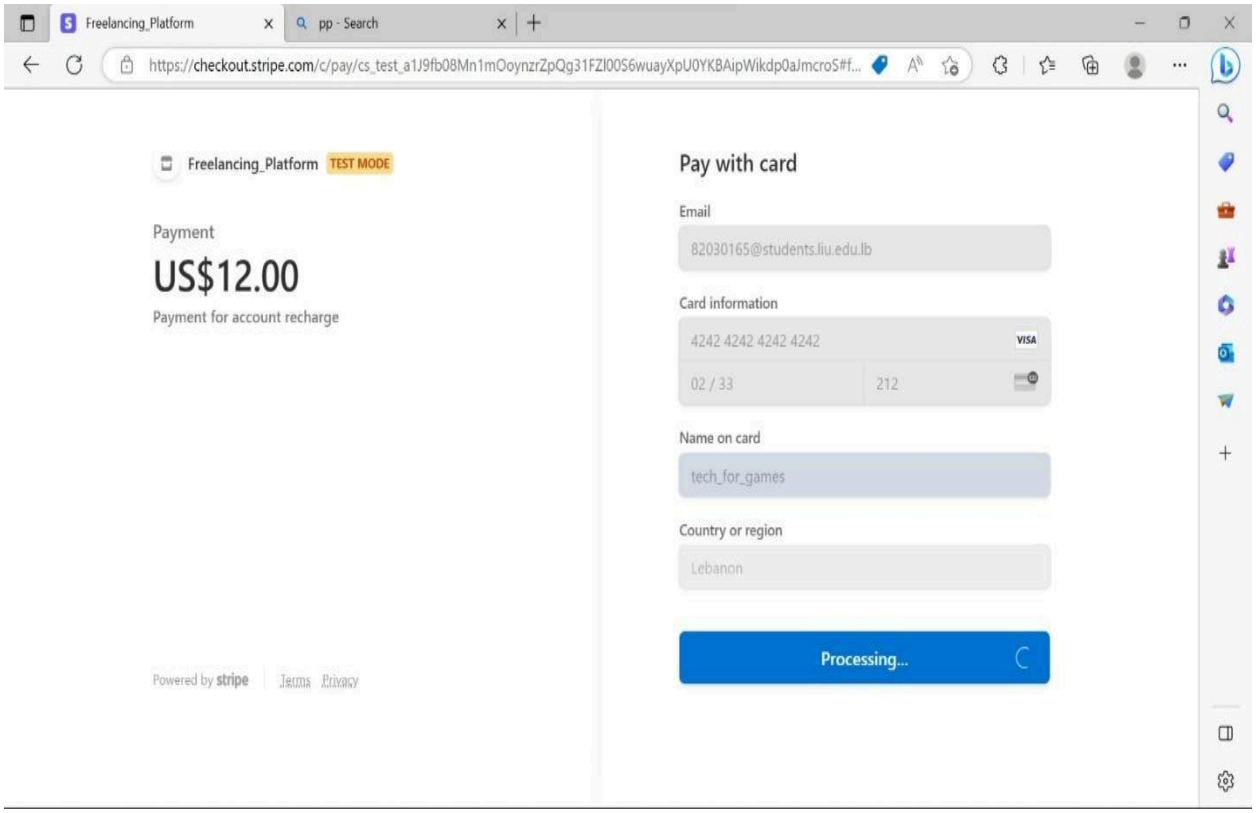
admin login dashboard



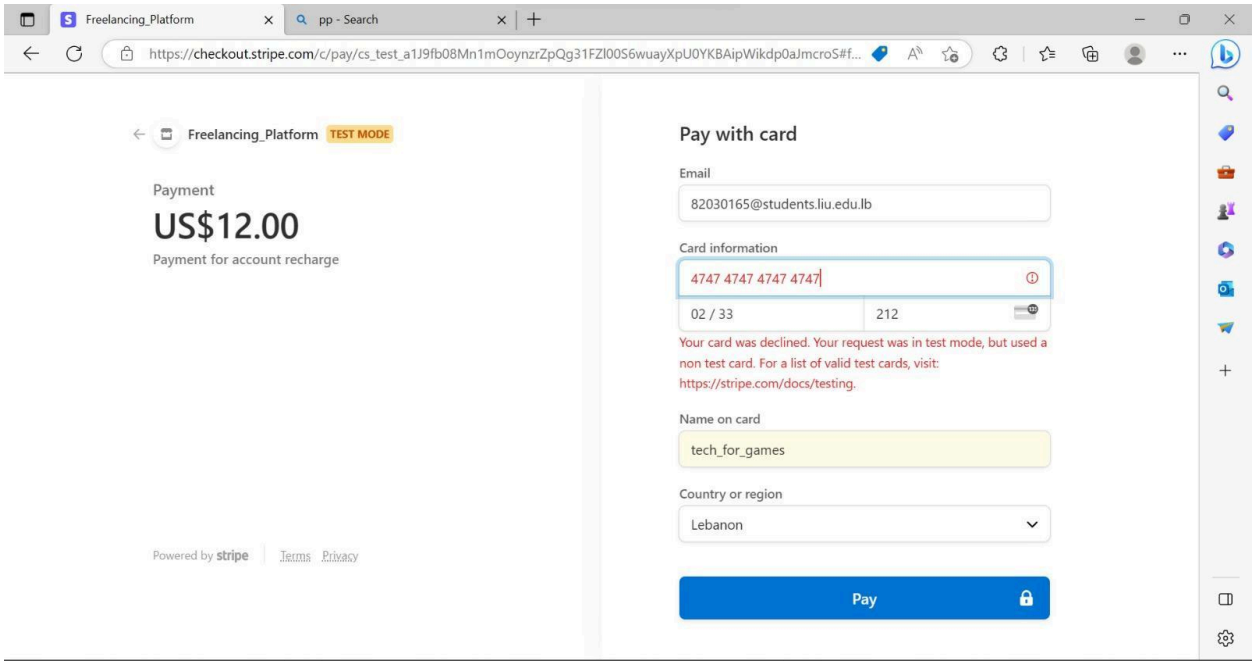
search by username

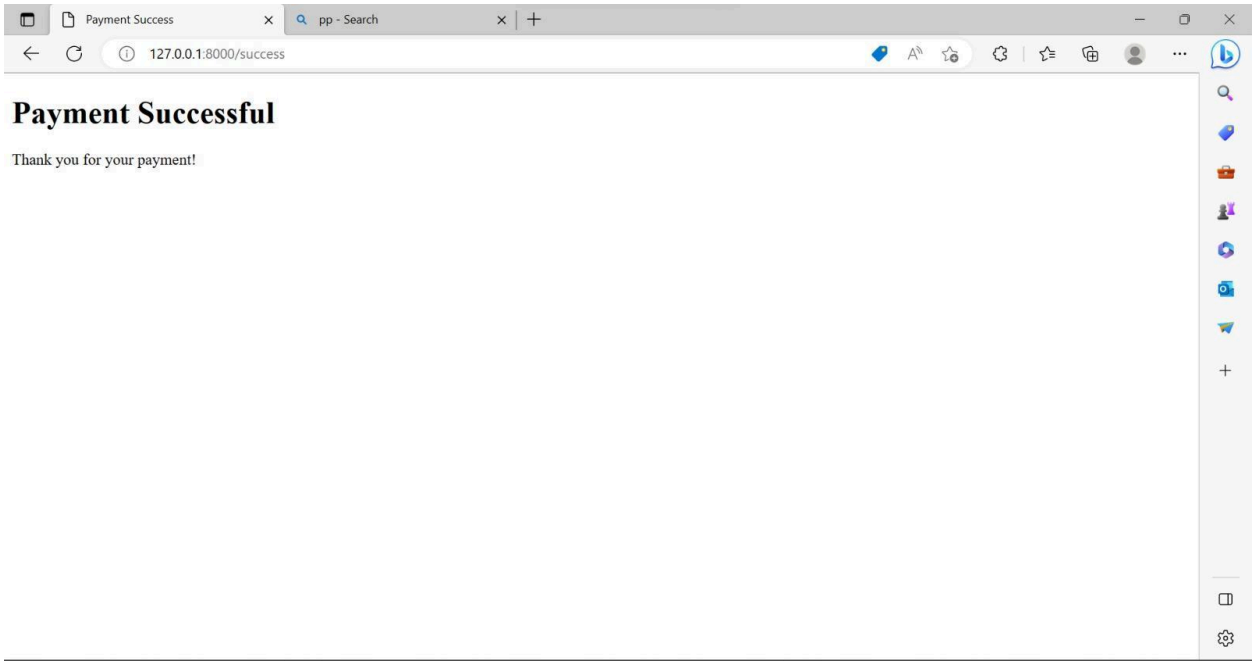


STRIP: this is the credit card that stripe gives me to test on non-real money

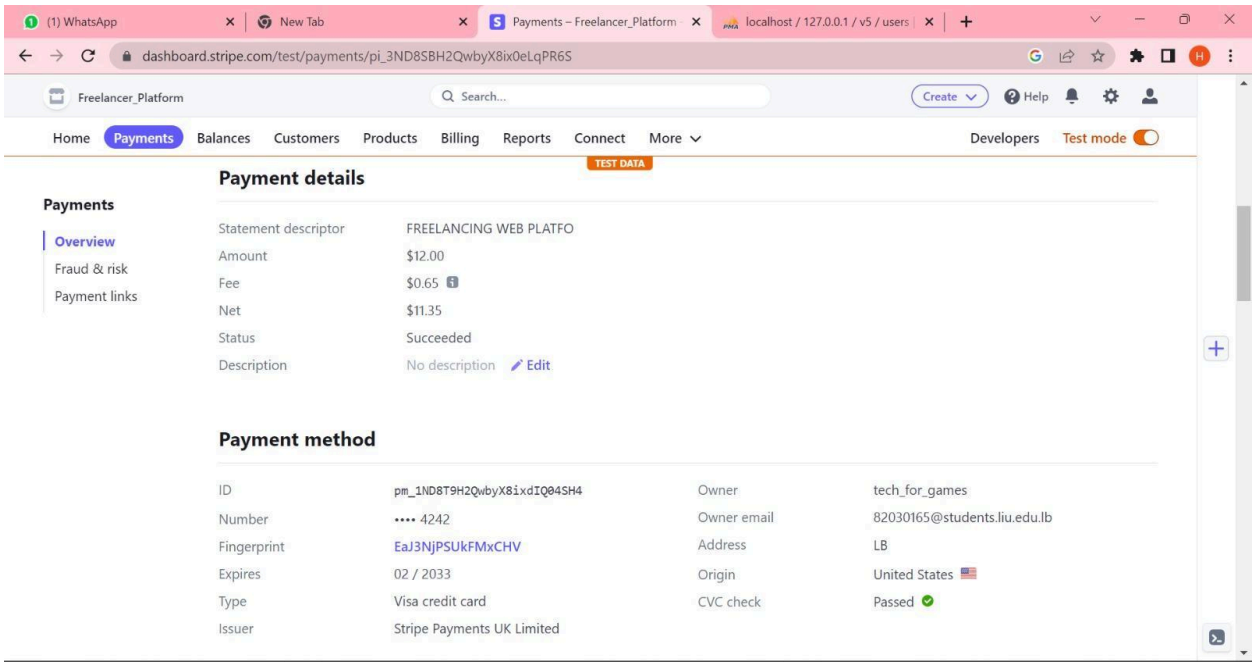


HOW STRIPE WORK:

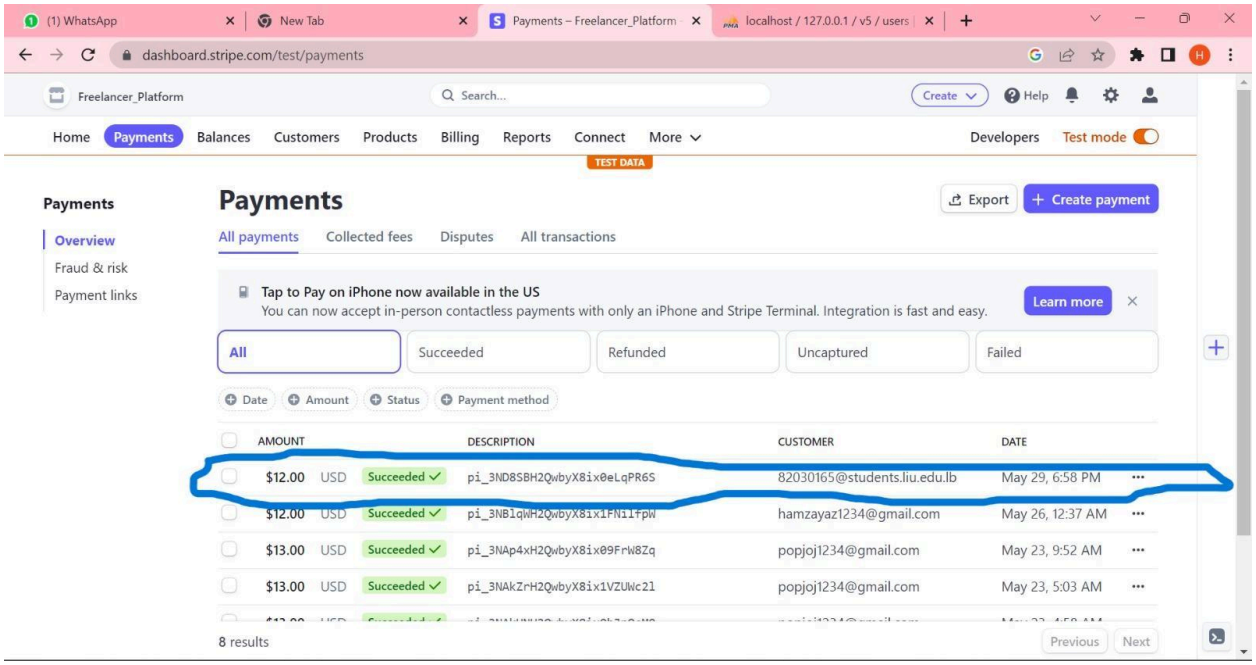




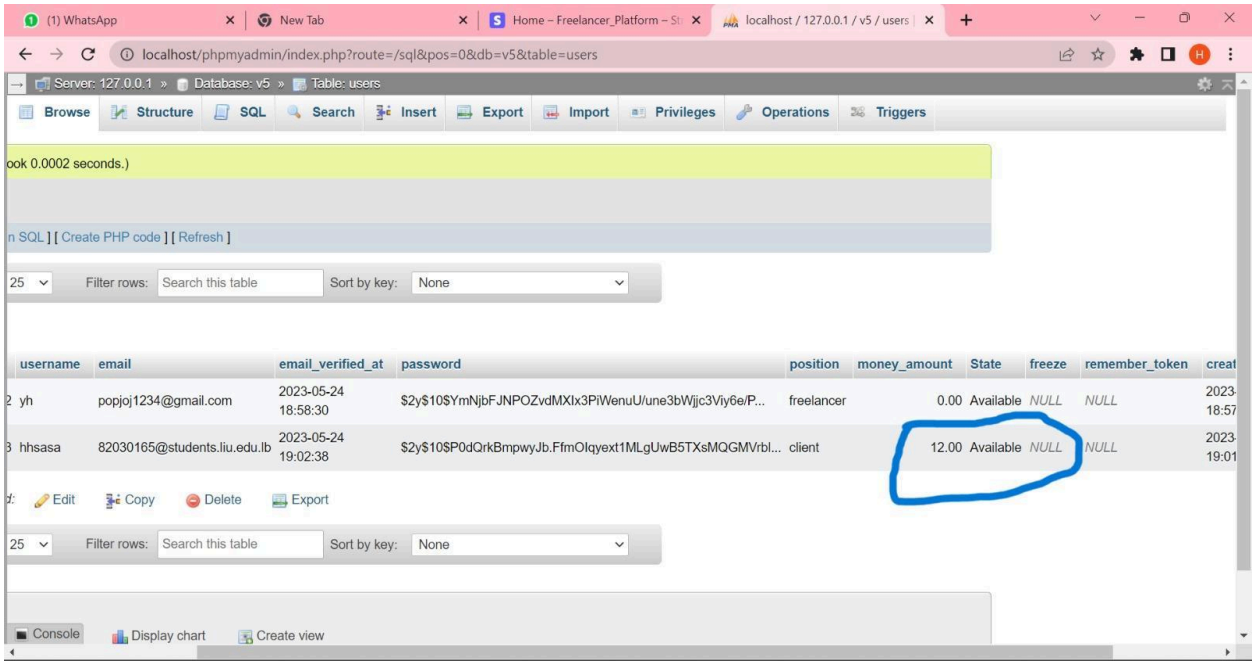
costumer info in stripe: customer payment summary knowing that I am not using any of these info to authenticate the users payment am using sanctum token and user id for these using local webhook between my local host and stripe to retrieve the payment data and ensure it.



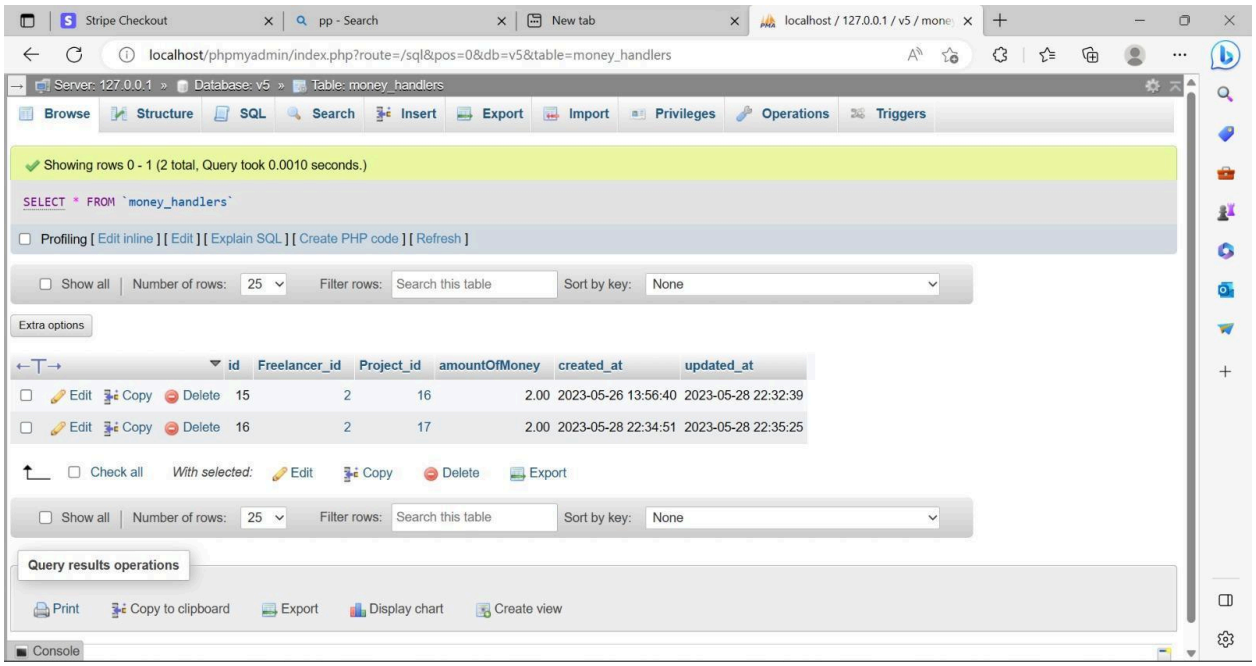
money recieved in my stripe account



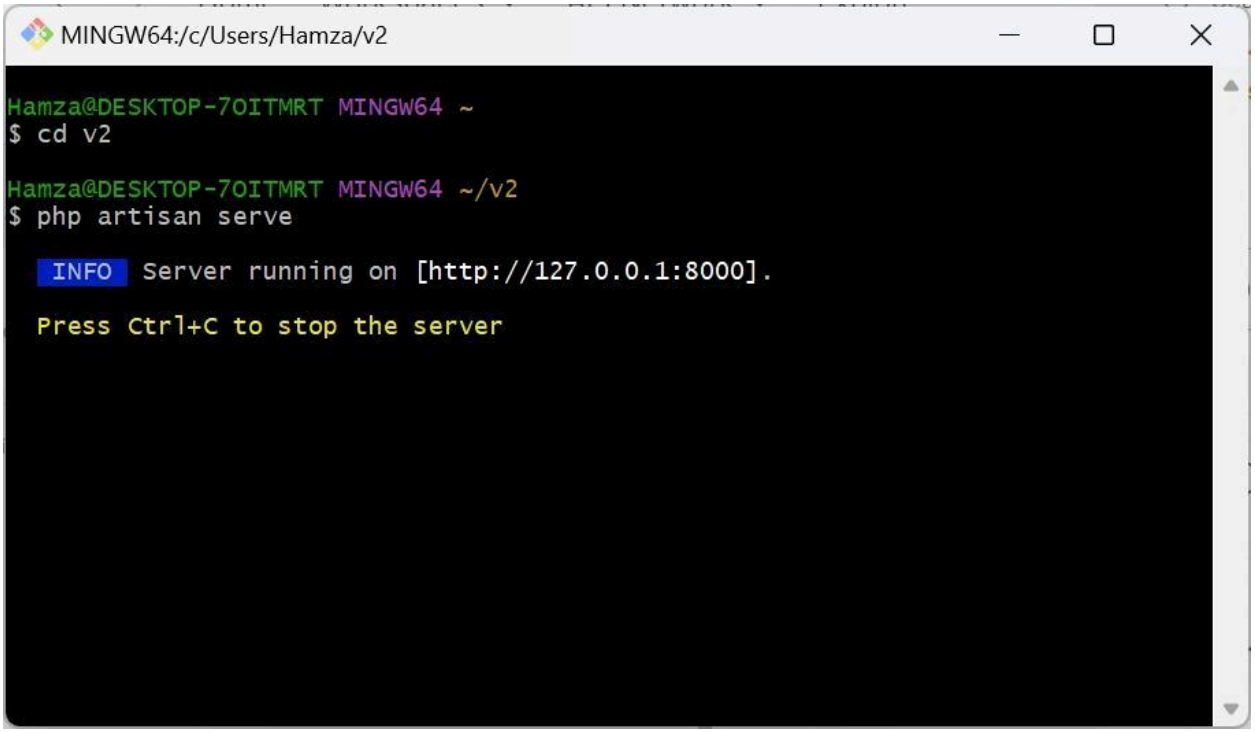
money added to the account successfully



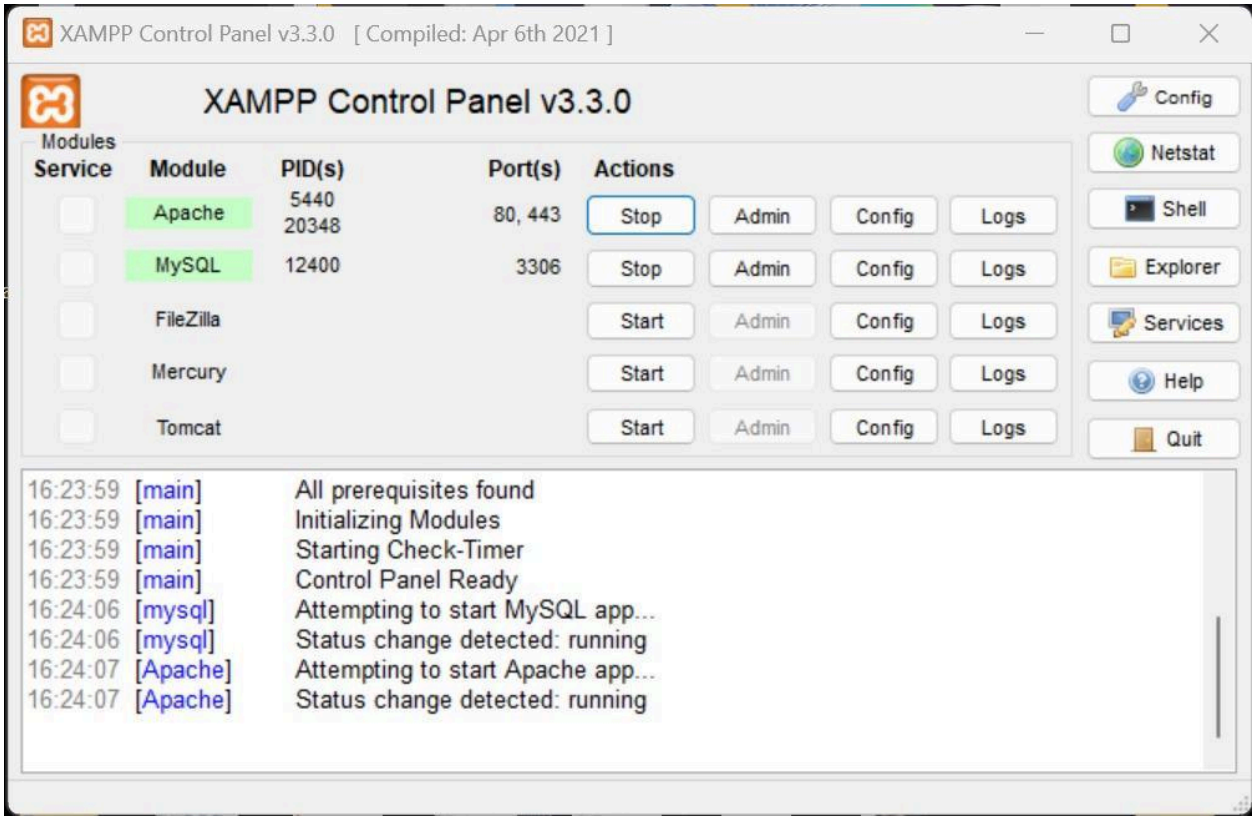
FUNDS TRANSFER: when accepting an application the money will be taken from client account and handle it in money handler then when the client turn the project to be done the money will be transferred from the money handler to the freelancer account.



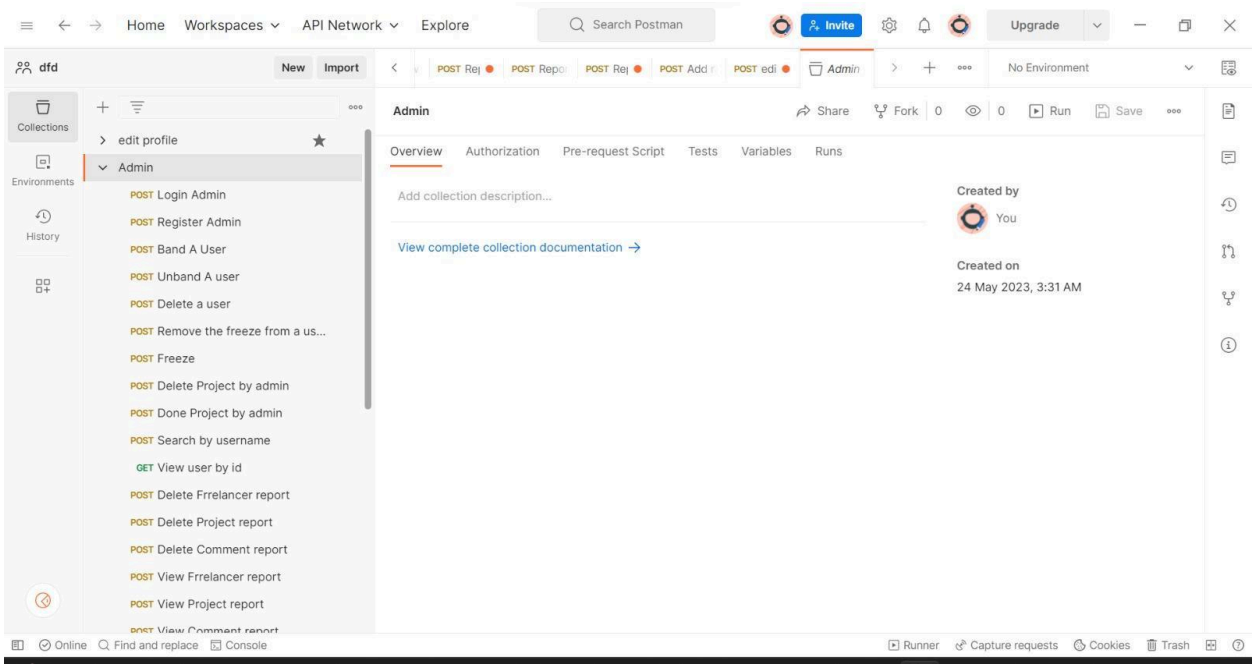
LOCAL HOST SERVER: making local host server for Laravel



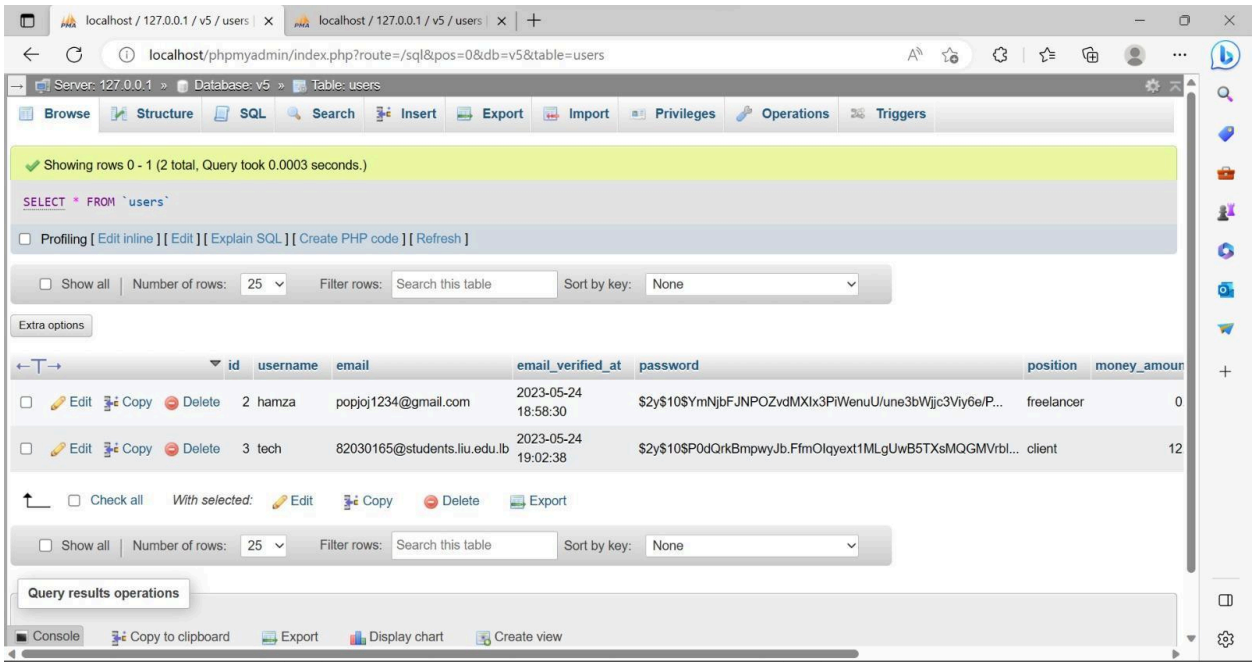
XAMPP: using XAMPP as local host database for MySQL.



POSTMAN: using postman to test the API'S


Home
Workspaces
API Network
Explore
Search Postman
Invite
Upgrade
dfd
New
Import
No Environment
Collections
Environments
History
edit profile
Admin
POST Login Admin
POST Register Admin
POST Band A User
POST Unband A user
POST Delete a user
POST Remove the freeze from a us...
POST Freeze
POST Delete Project by admin
POST Done Project by admin
POST Search by username
GET View user by id
POST Delete Frrelancer report
POST Delete Project report
POST Delete Comment report
POST View Frrelancer report
POST View Project report
Admin
Share
Fork 0
0
Run
Save
Overview
Authorization
Pre-request Script
Tests
Variables
Runs
Add collection description...
View complete collection documentation →
Created by
You
Created on
24 May 2023, 3:31 AM
Online
Find and replace
Console
Runner
Capture requests
Cookies
Trash

USER TABLE: I have hashed all the passwords of all the database tables



GANT CHART

TIME / TASK	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14	W15
Idea overview and tools															
planning															
Analyst and USECASE (system analysis)															
ERDIAGRAM (system design)															
Coding and documentation															
Final presentation															