Final Report

for

Tutorania (FYP)

Version 1.0

Prepared by

Fatima Mujahid - 289558

Fatima Tuz Zehra - 284892

Muhammad Maaz - 283826

Advisor: Dr. Rafia Mumtaz

Co-Advisor: Sir Taufique-ur-Rehman

May 19, 2023

Table of Contents

Tutorania

A Data Driven Platform for Connecting Tutors and Students

Background

Problem Statement

Finding the ideal teacher to help them advance their knowledge or competence in a particular subject or skill can be difficult for many students. This is as a result of their lack of direction and faith in others. This also makes it challenging for gifted people with the knowledge and capacity to assist others to get employment as tutors. By carefully compiling and keeping track of the credentials of tutors, Beacon Tutors has created a solution. When a student contacts the business for assistance, the company can look through its databaxcse of tutors and pair the student with the best fit.

However, because personal interaction and contract initiation are required, this approach may be ineffective and time-consuming. However, Beacon Tutors offer a much-needed service that aids in matching students with knowledgeable instructors and supports the career growth for gifted people who want to impart their knowledge and abilities to others.

Literature Review

In the past few years, online tutoring has become more and more popular. It has several advantages over teaching in person. Online tutoring gives students one-on-one help that can help them with their schoolwork, review what they've learned in class, get feedback, and feel more confident. Online teaching is also easy and flexible, so students can learn from anywhere and at any time. But some students and parents may prefer in-person tutoring because it gives them the chance to talk to the teacher face-to-face. There can also be problems with online tutoring, like technical problems and contact problems. During the COVID-19 pandemic, however, online coaching has been used successfully to give poor students free one-on-one tutoring. Beacon

Point and University Tutor are two online tutoring platforms that offer one-on-one tutoring, as well as tools and training for tutors. In conclusion, online tutoring can be a useful and easy way for tutors and students to meet, but it may not be right for everyone.

Using artificial intelligence (AI) and machine learning algorithms is one way to make it easier for teachers and students to find each other and make a deal. Gong and his colleagues (2020) used a model called a "decision tree" to find the best match between students and teachers based on the students' and tutors' traits and ratings. The matching was very accurate. Another way to make it easier for teachers and students to sign contracts is to use blockchain technology. Li and his colleagues (2020) suggested a blockchain-based system for online tutoring that would let tutors and students make and sign contracts, verify identities, and keep track of payments. But automating the matching and contracting process isn't easy. For example, it's hard to make sure that teachers are good and trustworthy and that the technology is easy for both tutors and students to use.

There may also be legal and regulatory hurdles to using certain technologies for contracts and payments, especially in different countries and jurisdictions. Still, AI systems help online learning and teaching in many ways, such as making learning more personalised for students and automating teachers' routine jobs. Online tutoring can offer personalised help that can be changed to fit the wants and learning style of each student. In-person tutoring, on the other hand, lets teachers see how the student interacts with course material in real time, so they can track the student's progress and make changes as needed. Overall, there are challenges and limits to using certain methods and technologies, but there is also a lot of promise for making online tutoring and personalised learning more efficient and effective.

Business Model

Tutorania is an online site for tutoring that puts students and tutors in touch with each other. Students can post jobs on the platform, which can be seen by teachers who have made profiles on the site. Tutors use a special currency that they keep in a digital wallet to apply for jobs. Students can buy this currency with real money. The platform gets an extra source of income by taking a cut of the money paid by the student to the teacher during the contract.

The primary way of making money for Tutorania is by selling custom currency. This currency is used to apply for jobs and look at tutor profiles on the site. By selling this cash, Tutorania gets a commission on each sale, which brings in money. Tutorania can use advertising on social media, search engine optimization, and reference programmes, among other ways to get people to use its platform. To set itself apart from rivals, the platform can offer extra features like a messaging system, virtual whiteboards, and tools for keeping track of progress. But Tutorania has to make sure it offers a safe and effective service if it wants to keep customers' trust and loyalty.

Stakeholders

Stakeholders in a project are people, organizations, or groups who have an interest in it. They are either involved in the project directly or have interests that could be affected by how it turns out. Tutorania also has multiple stakeholder because of its diverse nature and the categories are mentioned below:

- **Students** are Tutorania's main users because they use the platform to get academic help and support so they can learn more, improve their skills, and reach their academic goals.
- **Tutors** are in charge of giving students high-quality educational material, helping them with their homework and tests, and giving them feedback to help them learn better. Administrators supervise the day-to-day operations of Tutorania, such as managing the platform's infrastructure, supervising tutors' work, managing student accounts, and ensuring that the platform meets the needs of all stakeholders.
- **Parents** can use the platform to check on their child's progress, talk to tutors or administrators, and give feedback on the platform's educational material and services. In

order to offer more help to their students, incorporate Tutorania's resources into their curricula, or reach a wider audience with their online courses and programmes, educational institutions can collaborate with Tutorania.

- ❖ **Payment gateway companies** are crucial to Tutorania's ability to accept payments from its customers.
- ❖ The services of Tutorania are promoted by **marketing and advertising agencies** to potential consumers. Investors give Tutorania money in exchange for ownership of the company or a share of the income.
- ❖ **Regulators** make sure that Tutorania works in accordance with local and national laws. This protects the rights of students and keeps the level of education high.

Technology Stack

Purpose	Framework/Library
Programming Language	➢ Typescript
Microservice Development	➢ Node JS
Payment Integration	➢ Stripe/Payoneer
API Development	➢ Apollo GraphQL Client ➢ Apollo GraphQL Server
Frontend Development	➢ Next JS
DevOps	➢ Docker for Containerization ➢ Vercel for deployment
Recommendation Engine	➢ nltk
Analytics Dashboard	➢ Chart.js ➢ D3.js
Database	➢ Postgresql
UI Design	➢ Figma

	➢ Adobe XD
Graphics & Illustrations	➢ Adobe Illustrator ➢ Adobe Photoshop ➢ Adobe After Effects ➢ Adobe Premiere Pro ➢ Blender
Version Control	➢ Github
Project Management	➢ JIRA

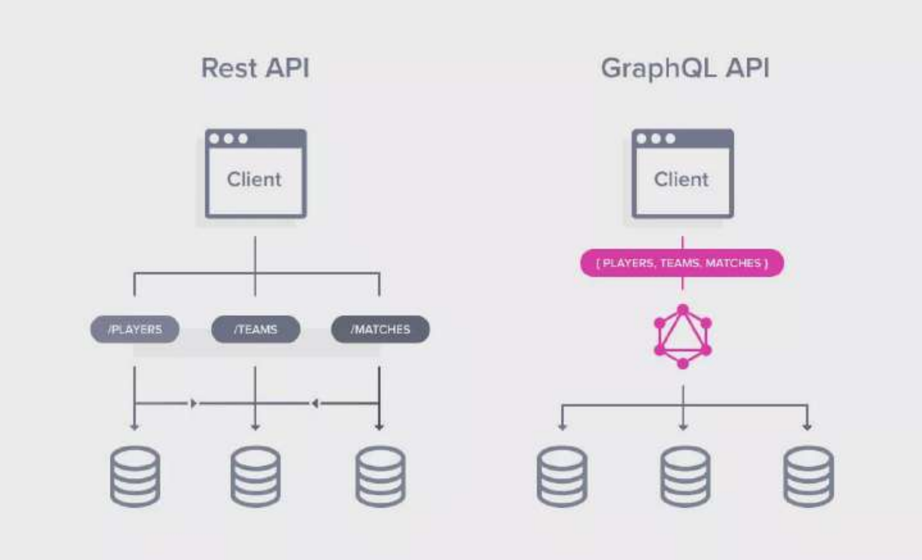
Proposed Solution and Implementation

Tutorania is a complete online tutoring platform that connects students who need help with tutors. The platform uses advanced data-driven methods, secure messaging, and forums to connect both the parties. It also provides a plethora of features for users to manage their contracts. To make transactions easier, Tutorania lets users keep coins in a Virtual Wallet. Coins can be bought using standard currencies like US Dollars or Euros, which requires the integration of a safe payment service. After the completion of a contract, users can leave feedback or reviews on the platform. This helps build a work and hiring history that can be used to indicate knowledgeable tutors. Tutorania also has a strong recommendation engine that is based on successful tactics used by good online marketplaces. This makes it easy for students to find the right tutors.

The platform also has an admin panel with services like data analytics, managing financial transactions, contract details, and managing forum posts. This gives administrators important information about how the platform works and how well it works. Effective tutoring is more than just giving answers or doing homework, and tutors can adapt their teaching methods to the way each student learns and give them one-on-one care. After that, we'll talk about the important parts of Tutorania:

1. API Design

Tutorania's design is based on the Microservices pattern, where each key module is built as a separate microservice. The microservices comprise the recommender, forums, user, contract, payment, and chat services. The GraphQL API gateway serves as a singular point of entry for client requests and utilizes REST APIs to direct these requests to their corresponding microservices. One of the benefits of using GraphQL in this design is the ability to do declarative data fetching. GraphQL enables clients to precisely specify the necessary data for a given request, thereby minimizing the occurrence of over-fetching or under-fetching of data. This makes it easier to get info and puts less pressure on the network bandwidth.



Major microservices are mentioned as follows:

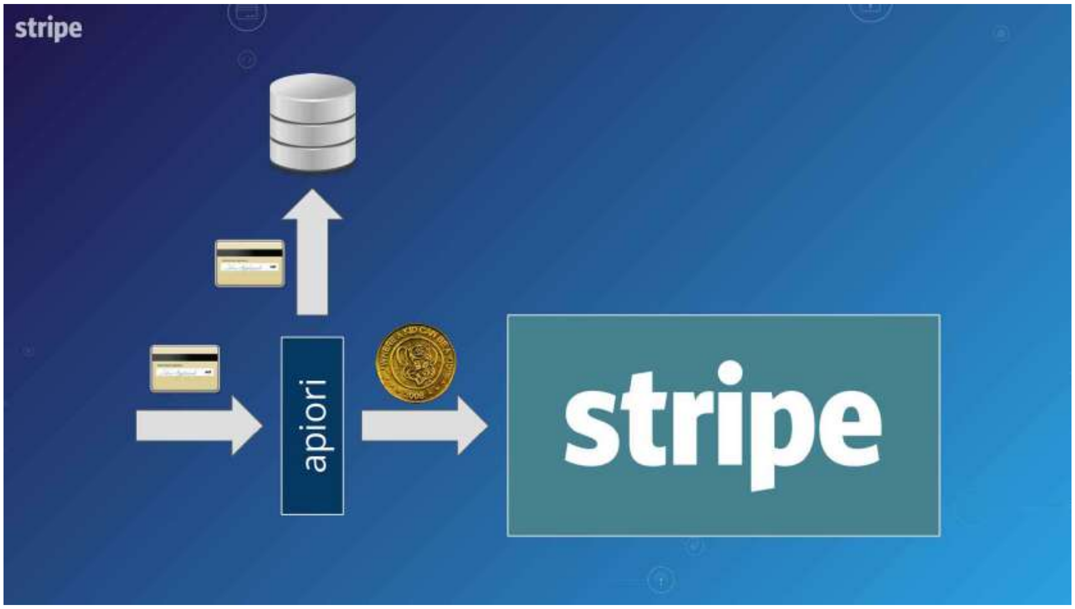
- **Recommendation service** matches students with the right teachers for them based on a number of factors.
- Participants in the forum can talk about academic problems, ask each other questions, and share information by using the **forum's microservice**.

- The **user microservice** is in charge of things like user authentication, user permission, and user maintenance. It is in charge of handling user accounts, profiles, and any changes to the data.
- **Contract microservice** can help teachers and students make, track, and keep track of deals. It can also change and end contracts that are already in place.
- **Payment microservice** are in charge of making sure payment methods are safe and taking care of money transfers between students and teachers. This function protects both buying virtual currencies and buying things inside of an app.
- The **chat microservice** makes it possible for students and teachers to talk to each other in real time. Alarms, messages, and encrypted contact are all taken care of by the system.
- The **analytics microservice** is responsible for all the analytics and charts and stats displayed on the admin panel.

With microservice design, Tutorania's system is more scalable, flexible, and stable, and each team has more freedom. The design of the site makes it easy for teachers and students to work together.

2. Payment Gateway Integration

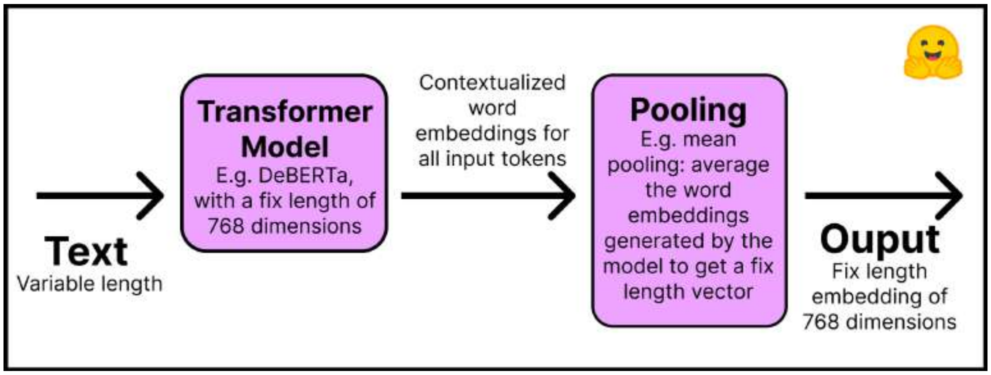
Stripe is the payment gateway that Tutorania has added so that students and teachers can send and receive money. Stripe is a trusted payment tool that is used by a lot of online businesses and has a lot of useful features and benefits. Stripe offers a safe way to process payments by using security methods that are common in the industry to protect sensitive financial information. This makes sure that all transfers on Tutorania are safe and can't be hacked or used for fraud. Stripe allows many ways to pay, such as credit cards, debit cards, and digital wallets like Apple Pay and Google Pay. This means that students can choose their preferred way to pay, which is more convenient and improves the user experience. Stripe makes it easy to make international payments because it works with multiple currencies and lets you do business in different places.



If Tutorania offers subscription-based services or recurring payments, Stripe has built-in tools to handle and automate these payment models. This makes billing and collecting payments on a regular basis much easier. The well-documented and developer-friendly APIs provided by Stripe facilitate the integration of payment functionality into Tutorania's platform. Stripe offers ways to settle disputes and makes it easier to handle chargebacks. If there are any problems with a payment, Stripe gives Tutorania the tools and rules it needs to handle the situation well and keep its operations running smoothly. Stripe is known for always coming up with new ideas and adding new features. This means that Tutorania can keep up with the latest payment technologies and trends by taking advantage of Stripe's constant innovations and new features. In general, using Stripe as the payment gateway gives Tutorania a reliable, safe, and feature-rich way to handle financial transactions within the platform.

3. Recommendation Engine

The SentenceTransformers, which are considered to be a benchmark in natural language processing and semantic similarity, are utilized by the recommendation engine. By utilizing SentenceTransformers, the system is able to build high-quality, dense embeddings for sentences or text samples. These embeddings retain the semantic meaning of the text and make it possible to do comparisons and similarity assessments in an effective manner. A SentenceTransformer is used by the recommendation system in order to build an embedding for a job description whenever a student publishes one. In a similar manner, embeddings are crafted for the profile descriptions of each tutor as well as their prior work experiences.


Text
Variable length
Transformer Model
E.g. DeBERTa, with a fix length of 768 dimensions
Contextualized word embeddings for all input tokens
Pooling
E.g. mean pooling: average the word embeddings generated by the model to get a fix length vector
Ouput
Fix length embedding of 768 dimensions

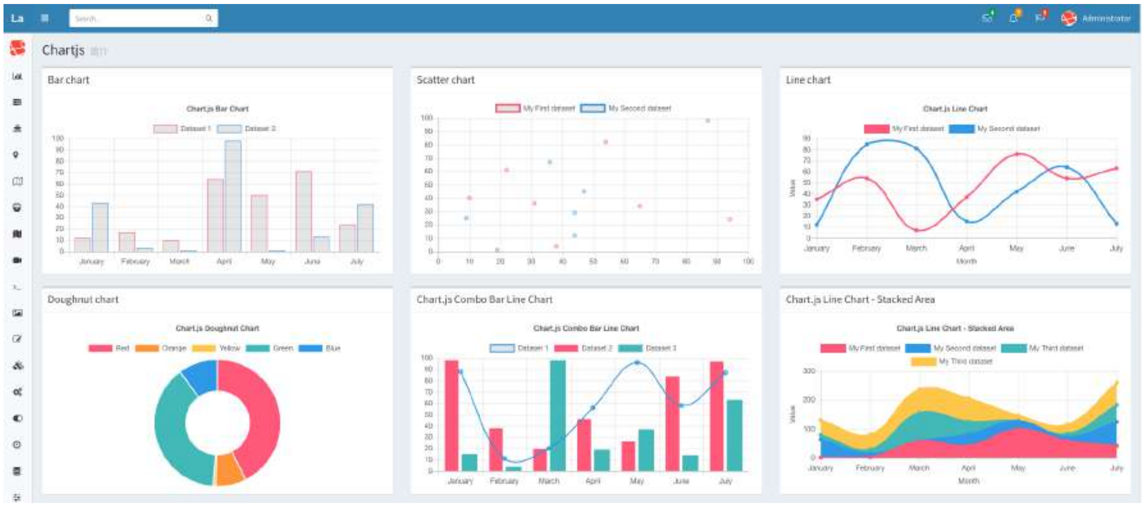
A cosine similarity analysis is performed on the embeddings in order to determine how well the job description and the tutor profiles match up with one another. In addition to job experiences and profile descriptions, the recommendation system takes into account weighted averages of the comments received for past jobs. Using the weighted average of feedback scores, the system calculates the cosine similarity between the job description embedding and each tutor's profile description and past job embeddings. The algorithm identifies the tutors whose descriptions and experiences are most similar to the job criteria based on these computations, taking into consideration both semantic similarity and feedback ratings. Because of this method, Tutorania is able to build a robust recommendation engine that takes advantage of the most recent developments in natural language processing techniques. It helps connect students with the best tutors based on how well their profile descriptions, job experiences, and feedback scores match up.

4. Contract Management

Tutorania has a thorough contract management system that makes it easier for students and tutors to talk to each other and work together throughout the tutoring process. The contract management system lets you send messages, keep track of milestones, get approval for contracts, and end contracts. Students and tutors can exchange messages within the platform, allowing for plain and direct communication regarding the tutoring arrangement, scheduling, specific requirements, and other pertinent information. The contract management system lets milestones, which are specific goals or deliverables for the tutoring job, be added to contracts. Students and tutors can both set goals to track progress and make sure the tutoring process stays on track. Before coaching starts, both the student and the teacher look over the contract and agree to it. This makes sure that both sides agree on the contract's scope, price, length, and other details. After the teaching contract is over, the student and teacher can talk about what they learned. The contract management option from Tutorania makes it easier to find and work with a tutor by putting all communication, tracking of milestones, and contract management in one place. It makes students and teachers more responsible and improves the user experience. Tutorania combines these tools so that teaching contracts can be managed and kept track of. This makes teaching both fun and useful.

5. Admin Panel with Analytics

Tutorania's managers can keep an eye on what's going on on the platform by using the admin interface. Using the admin panel, administrators can take care of user accounts, jobs, and contracts. It does this by giving outlines of important data in the form of charts and graphs. Administrators can take care of site-listed jobs. They can keep an eye on everything, see how things are going, and step in if needed. In the administration area, you can take care of contracts between students and teachers. It can also help you understand contracts, keep track of how they are going, and solve any problems that may come up. The charts and graphs in the administrative panel show a summary of the statistics. This shows how well the platform is doing, how much its users are using it, how the job market is changing, and other important things.



All of Tutorania's parts can be broken down into specific reports that can be made from the administrative control panel. These reports give management information that can be used to evaluate the platform, find places to make improvements, and make strategic choices. Administrators can manage user accounts, enforce platform regulations, handle user-generated material, monitor forums, and perform other administrative tasks to keep the platform secure and reliable from the administration panel. In Tutorania's Admin Panel, you can control user profiles, jobs, and contracts. Administrators can better analyze the data and make choices that will improve the user experience and the platform's performance.

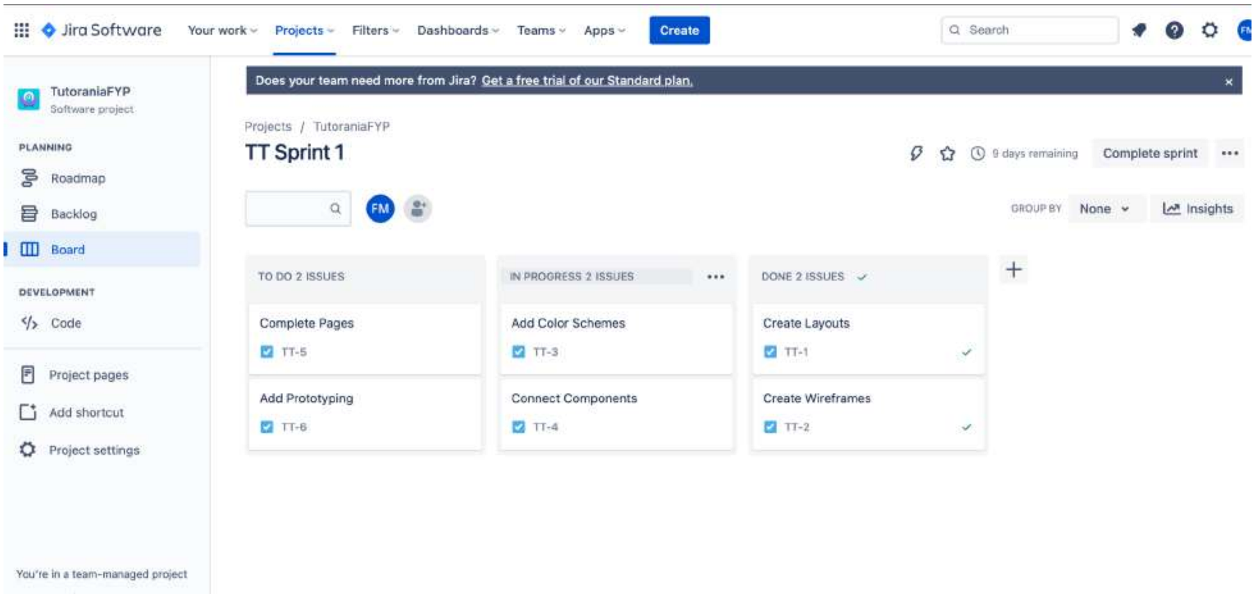
6. User Interface and Experience

Tutorania puts the user interface (UI) and user experience (UX) at the top of its list of priorities to make a platform that looks good and is easy to use. The UI of Tutorania has been carefully designed to look good, be easy to use, and be interesting to look at. All of the design elements, like the font, spacing, and style, were chosen with care to make the site look polished and professional. Tutorania keeps the same colour scheme throughout the whole platform. This makes sure that everything looks good and fits together. The platform makes use of components from the Mantine library, which provides a set of aesthetically pleasing and modifiable user interface components. These parts can be changed so that they fit with Tutorania's branding and design guidelines. This gives the

UI a personal touch and makes sure that it stays true to Tutorania's character. Tutorania follows important UX laws and principles to make a straightforward and consumer-friendly experience. It takes into account things like simplicity of navigating, organized structure, rational processes, and effective exchanges to improve utility and customer satisfaction.

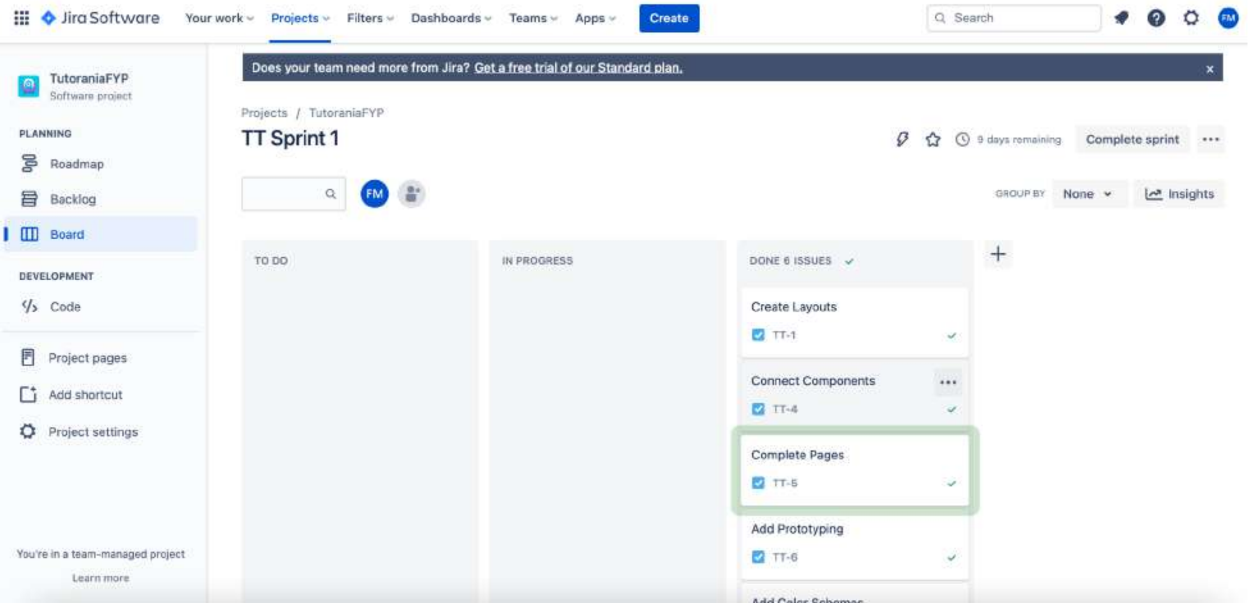
Project Management, Collaboration and Version Control

For our project, Tutorania, we used JIRA for project management and followed the SCRUM methodology. We worked in sprints, which were short timeframes where we focused on specific tasks. This helped us stay organized, review progress regularly, and make adjustments as needed.



TT Sprint 1

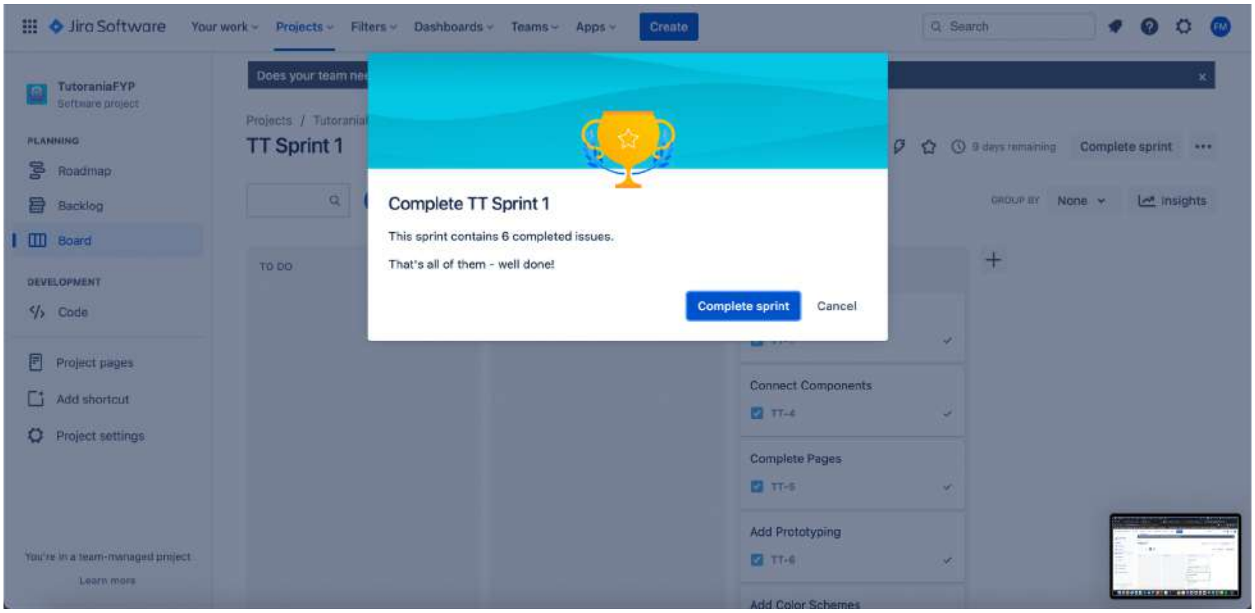
TO DO 2 ISSUES	IN PROGRESS 2 ISSUES	DONE 2 ISSUES ✓
Complete Pages ☑ TT-5	Add Color Schemes ☑ TT-3	Create Layouts ☑ TT-1 ✓
Add Prototyping ☑ TT-6	Connect Components ☑ TT-4	Create Wireframes ☑ TT-2 ✓



Does your team need more from Jira? Get a free trial of our Standard plan.

TT Sprint 1

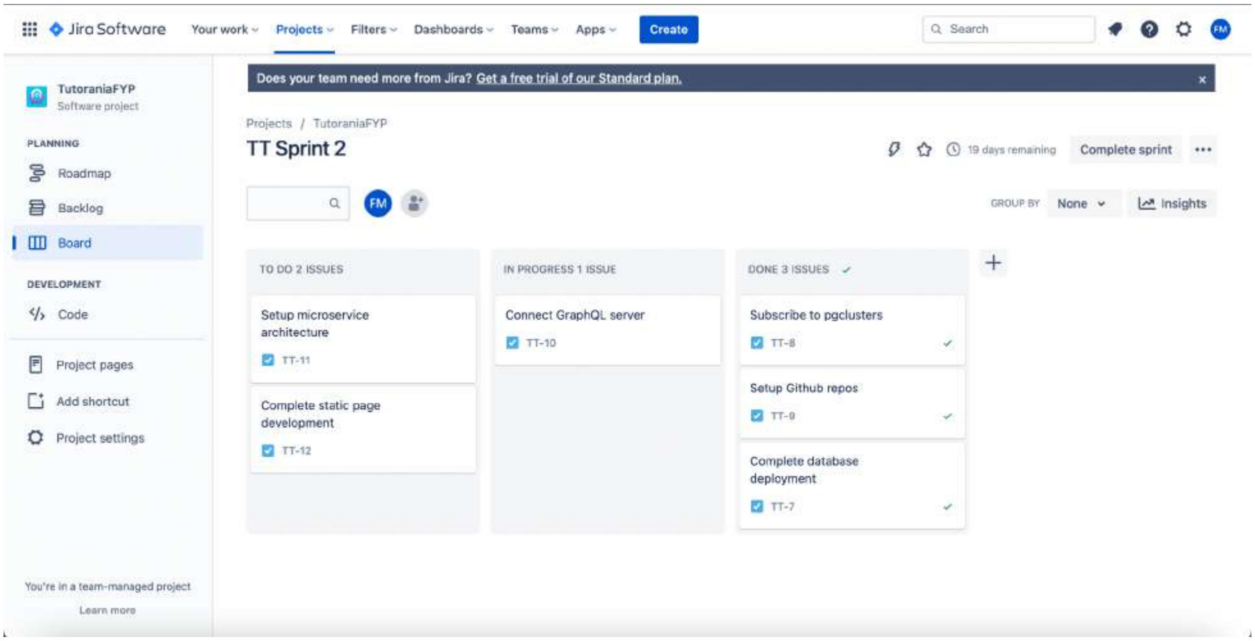
- Create Layouts — TT-1
- Connect Components — TT-4
- Complete Pages — TT-5
- Add Prototyping — TT-6
- Add Color Schemes



Complete TT Sprint 1

This sprint contains 6 completed issues.

That's all of them - well done!



Does your team need more from Jira? Get a free trial of our Standard plan.

Projects / TutoraniaFYP

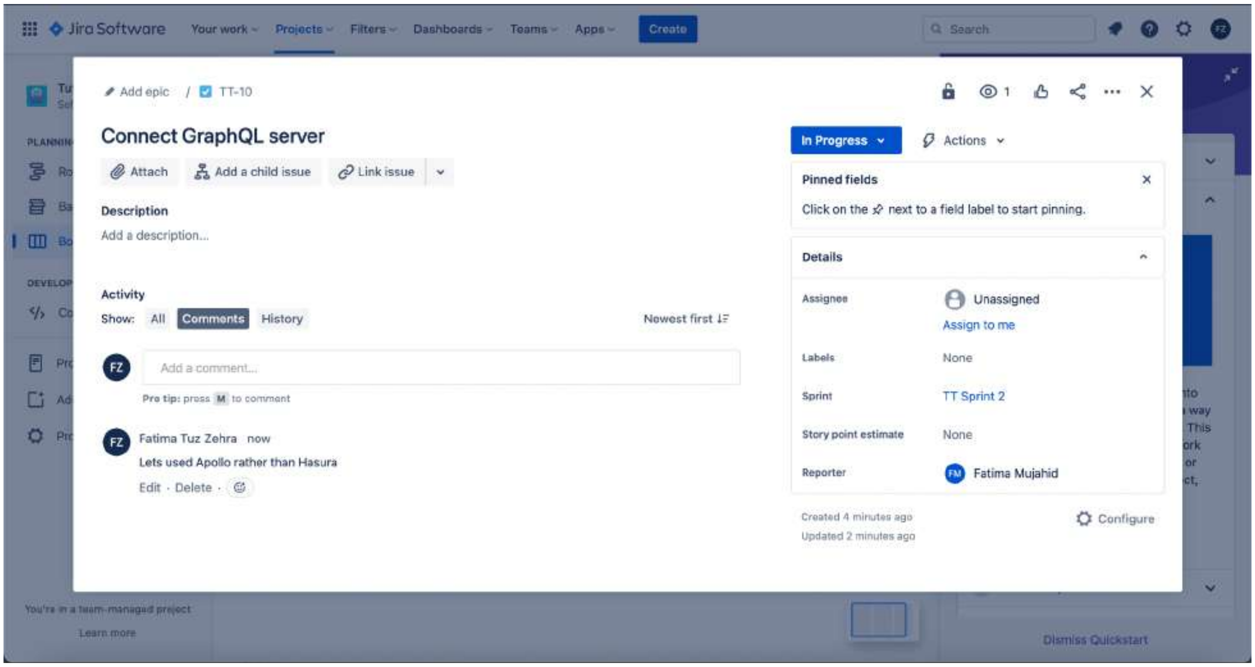
TT Sprint 2

19 days remaining

TO DO 2 ISSUES
- Setup microservice architecture — TT-11
- Complete static page development — TT-12

IN PROGRESS 1 ISSUE
- Connect GraphQL server — TT-10

DONE 3 ISSUES
- Subscribe to pgclusters — TT-8
- Setup Github repos — TT-9
- Complete database deployment — TT-7



Add epic / TT-10

Connect GraphQL server

Description

Add a description...

Activity

Show: All Comments History Newest first

Add a comment...

Pro tip: press M to comment

Fatima Tuz Zehra now

Lets used Apollo rather than Hasura

Edit · Delete ·

Pinned fields

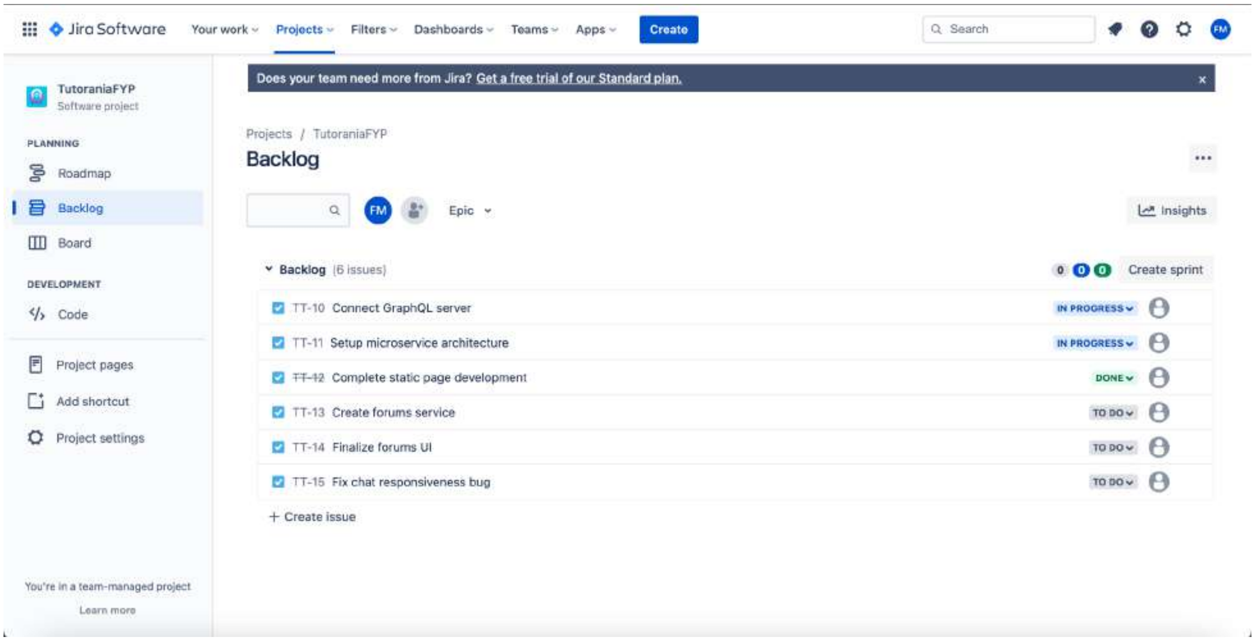
Click on the next to a field label to start pinning.

Details

Assignee	Unassigned Assign to me
Labels	None
Sprint	TT Sprint 2
Story point estimate	None
Reporter	Fatima Mujahid

Created 4 minutes ago

Updated 2 minutes ago



Does your team need more from Jira? Get a free trial of our Standard plan.

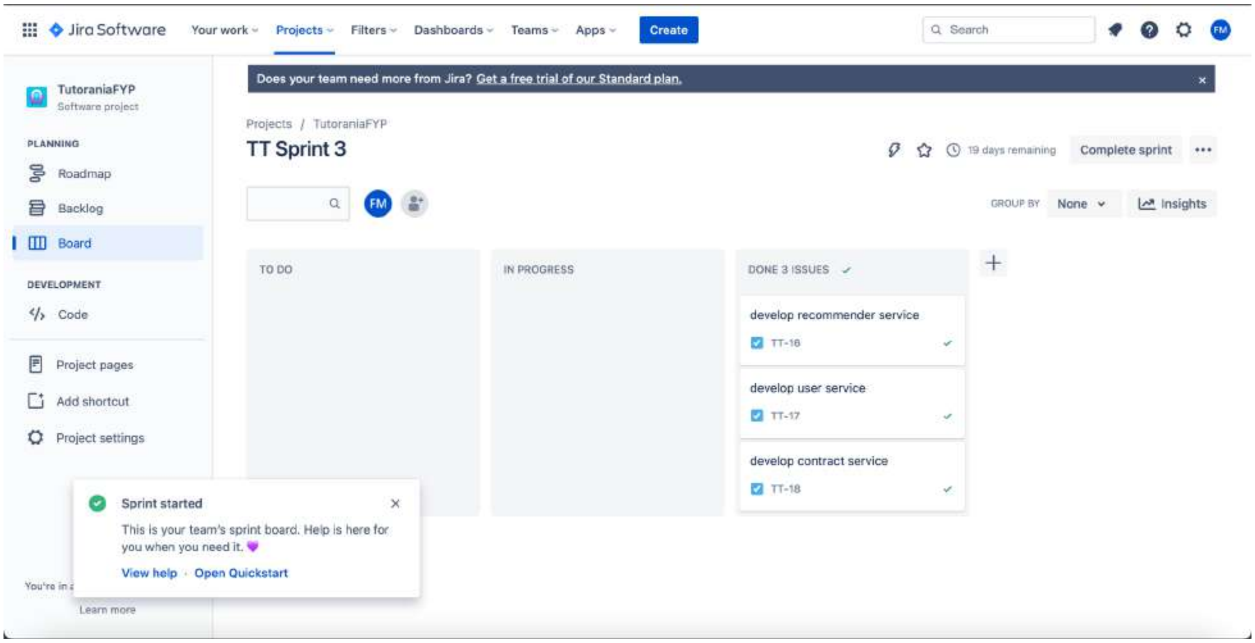
Projects / TutoraniaFYP

Backlog

Backlog (6 issues) 0 0 0

TT-10	Connect GraphQL server	IN PROGRESS
TT-11	Setup microservice architecture	IN PROGRESS
~~TT-12~~	Complete static page development	DONE
TT-13	Create forums service	TO DO
TT-14	Finalize forums UI	TO DO
TT-15	Fix chat responsiveness bug	TO DO

You're in a team-managed project



Does your team need more from Jira? Get a free trial of our Standard plan.

Projects / TutoraniaFYP

TT Sprint 3

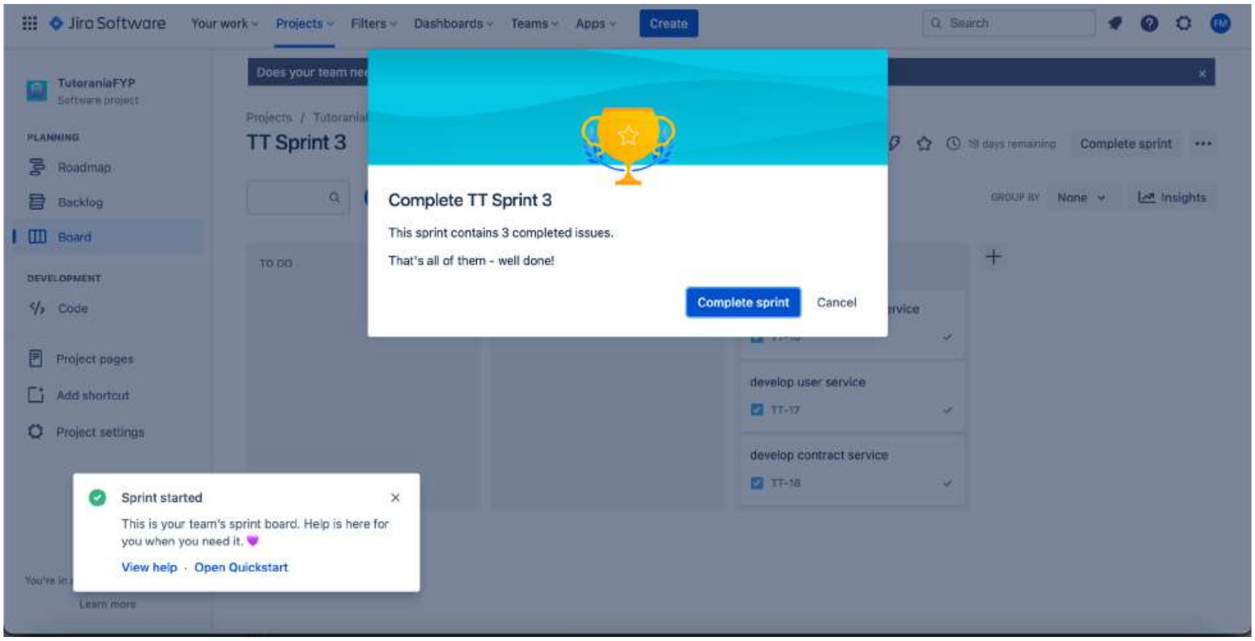
19 days remaining

TO DO	IN PROGRESS	DONE 3 ISSUES ✓
		develop recommender service TT-16 ✓
		develop user service TT-17 ✓
		develop contract service TT-18 ✓

Sprint started

This is your team's sprint board. Help is here for you when you need it. 💜

You're in a



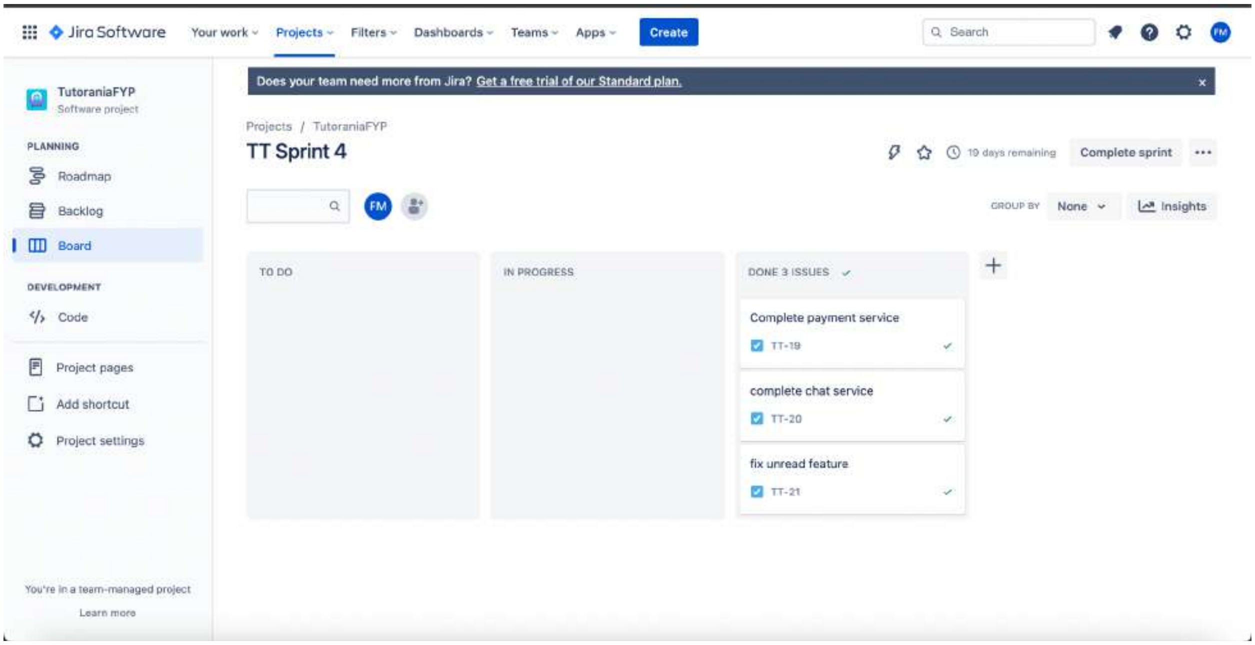
Complete TT Sprint 3

This sprint contains 3 completed issues.

That's all of them - well done!

Sprint started

This is your team's sprint board. Help is here for you when you need it. 💜



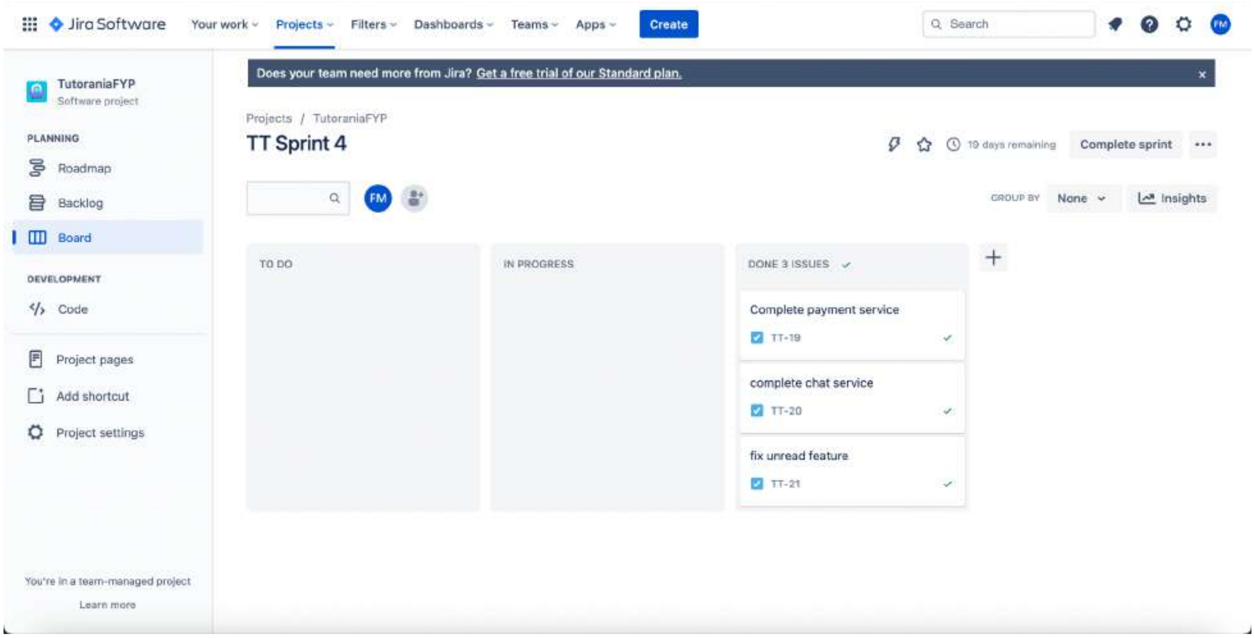
Does your team need more from Jira? Get a free trial of our Standard plan.

TT Sprint 4

Complete payment service

complete chat service

fix unread feature



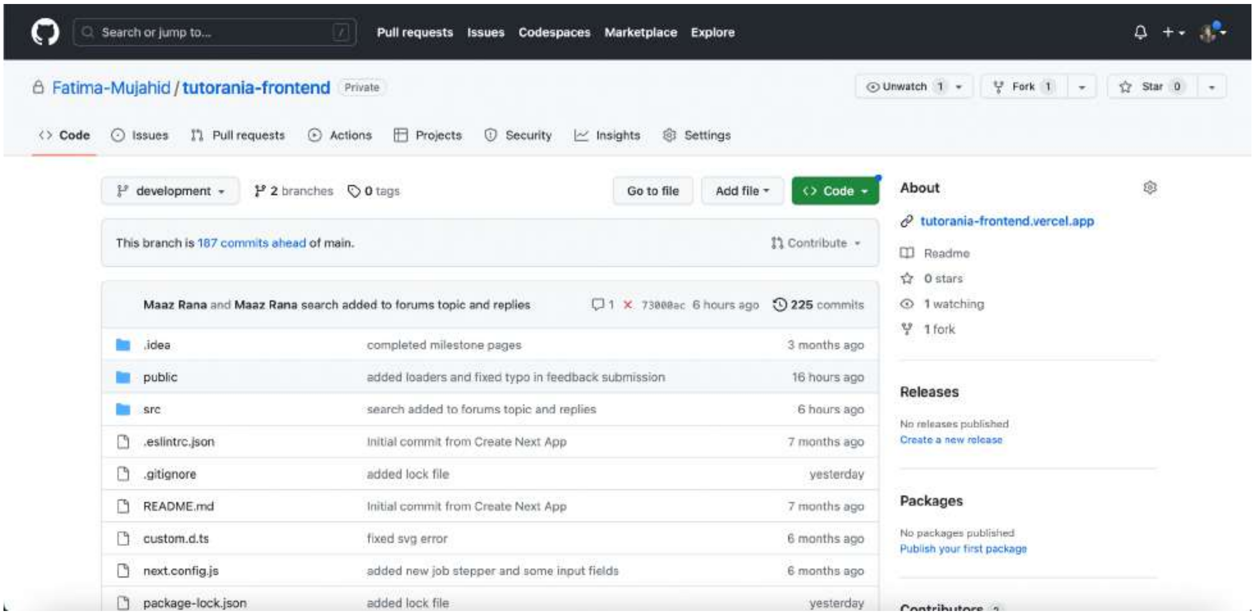
Does your team need more from Jira? Get a free trial of our Standard plan.

TT Sprint 4

19 days remaining

TO DO	IN PROGRESS	DONE 3 ISSUES ✓
		Complete payment service ☑ TT-19 ✓
		complete chat service ☑ TT-20 ✓
		fix unread feature ☑ TT-21 ✓

To collaborate and keep track of our Tutorania project, we used GitHub. It helped us work together, track changes, and go back to older versions if necessary.



Fatima-Mujahid / tutorania-frontend Private

This branch is 187 commits ahead of main.

Maaz Rana and Maaz Rana search added to forums topic and replies 1 ✕ 73000ac 6 hours ago 225 commits

.idea	completed milestone pages	3 months ago
public	added loaders and fixed typo in feedback submission	16 hours ago
src	search added to forums topic and replies	6 hours ago
.eslintrc.json	Initial commit from Create Next App	7 months ago
.gitignore	added lock file	yesterday
README.md	Initial commit from Create Next App	7 months ago
custom.d.ts	fixed svg error	6 months ago
next.config.js	added new job stepper and some input fields	6 months ago
package-lock.json	added lock file	yesterday

tutorania-frontend.vercel.app

Readme

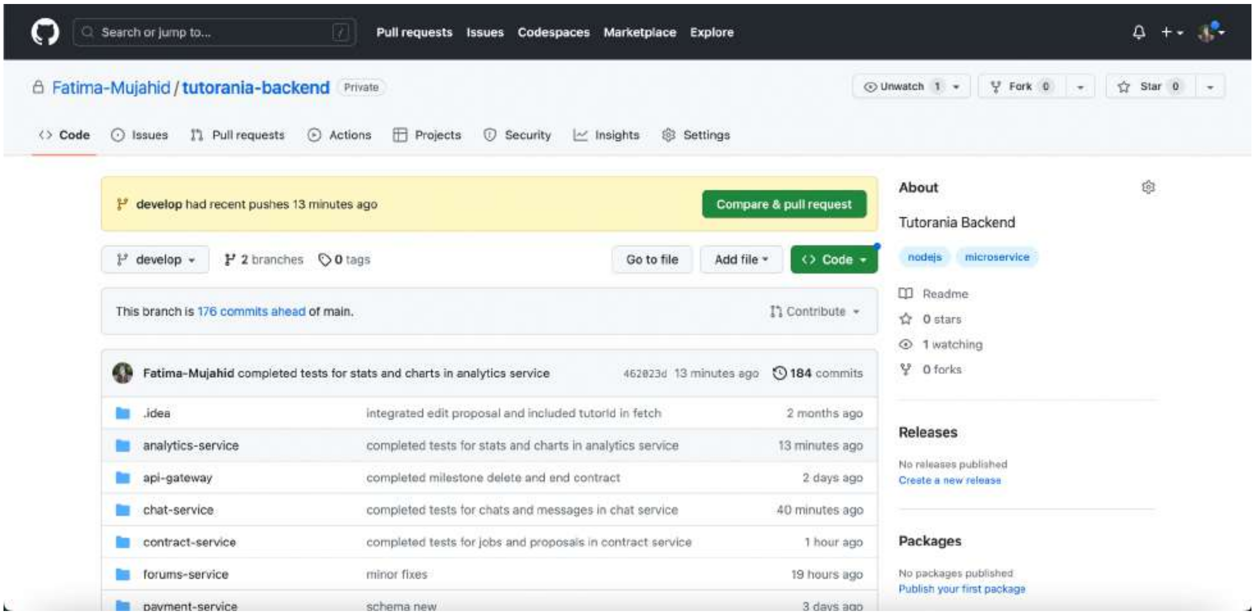
0 stars

1 watching

1 fork

No releases published
Create a new release

No packages published
Publish your first package



develop had recent pushes 13 minutes ago

2 branches 0 tags

This branch is 176 commits ahead of main.

Fatima-Mujahid completed tests for stats and charts in analytics service 462023d 13 minutes ago 184 commits

.idea	integrated edit proposal and included tutorId in fetch	2 months ago
analytics-service	completed tests for stats and charts in analytics service	13 minutes ago
api-gateway	completed milestone delete and end contract	2 days ago
chat-service	completed tests for chats and messages in chat service	40 minutes ago
contract-service	completed tests for jobs and proposals in contract service	1 hour ago
forums-service	minor fixes	19 hours ago
payment-service	schema new	3 days ago

Tutorania Backend

0 stars

1 watching

0 forks

No releases published

No packages published

Software Requirement Specification

Requirements Elicitation Process

One of the most important steps in making any software project is called "requirement elicitation." During this time, different stakeholders' needs and expectations are gathered, understood, and written down. In the case of Tutorania, a method called "requirement elicitation" was a key part of how the platform was changed to meet the needs of the tutoring industry. There were a number of important steps that had to be taken in the right order to figure out what Tutorania needed.

- Interviews with the **owner of Beacon Tutors** since he was the main person who called for the project.
- **Prototypes made in Figma** that can be used for getting input, working together, and analyzing a system. The process took into account what customers said, what the owner knew, and what the business standards were.

- To learn more about requirement elicitation, there are many resources available online, such as tutorials and courses offered by **Simplilearn, GeeksforGeeks, Testbytes, and SoftwareTestingHelp.**

It is important for product teams to use the requirement elicitation method to find problems with requirements before the development process starts. This helps bring down the total cost of the job. Business analysts need to be able to know what questions to ask and how to ask them in the best way. On top of that, they must be able to speak well and work together with key stakeholders throughout the whole process.

Functional Requirements

1.1 User Profile Creation

1.1.1 Description and Priority

The system shall enable users suited for the platform to create accounts or profiles.

Priority: High

1.1.2 Stimulus/Response Sequences

The user visits the landing page and navigates to the sign in or sign up section of the website.

1.1.3 Functional Requirements

REQ-1: The system shall maintain three types of accounts as shown below:

1. Student
2. Tutor
3. Admin

REQ-2: The system shall provide interface for the creation of only two types of accounts as shown below:

1. Student

2. Tutor

REQ-3: The system shall allow users of all account types to log in to their accounts.

REQ-4: The system shall obtain given details while logging in:

1. Email
2. Password

REQ-5: The system shall ask for the given mandatory input while account creation of both student and tutor:

1. First Name
2. Last Name
3. Email
4. Date of Birth
5. Location
6. Gender

REQ-6: The system shall enable both student and teacher to optionally add a profile picture while signing up.

REQ-7: The system shall send a 4 digit code to the email address specified by the user in order to complete the registration process.

REQ-8: The system shall allow the user to resend the email verification code after every 60 seconds.

1.2 User Profile Editing and Management

1.2.1 Description and Priority

The system shall enable users registered on the profile to edit their profile details and add additional information based on their account type

Priority: High

1.2.2 Stimulus/Response Sequences

The user logs in to his account and navigates to the my profile section.

1.2.3 Functional Requirements

REQ-1: The system shall allow the user to change the current password after a successful login.

REQ-2: The system shall allow the user to change first and last name after a successful login.

REQ-3: The system shall allow the user to change his location after a successful login.

REQ-4: The system shall allow the user to change his profile picture after a successful login.

REQ-5: The system shall allow a tutor to add a profile description after a successful login.

REQ-6: The system shall allow a tutor to add a profile headline after a successful login.

REQ-7: The profile headline entered by the tutor should be no greater than 100 characters and no lesser than 8 characters

REQ-8: The profile description entered by the tutor should be no greater than 700 characters and no lesser than 50 characters

REQ-9: The system shall allow a tutor to add past employment items to his profile which will reflect past jobs completed outside Tutorania.

REQ-10: The system shall allow a tutor to add job history items to his profile which
will reflect previous jobs completed in Tutorania.

REQ-11: The job history items will have the inputs specified in .

REQ-12: The system shall require following inputs to associate a past employment item with a tutor profile:

1. Job Title
2. Company or Employer Title
3. Job Description
4. Start Date
5. End Date

REQ-13: The start and end date for profile history items will have {monthName year} format.

REQ-14: The system shall allow a tutor to choose from a list of pre-existing skill items.

REQ-15: The system shall allow a tutor to edit the add a profile headline after a successful login.

REQ-16: The system shall allow a tutor to edit the profile description after a successful login.

REQ-17: The system shall allow a tutor to edit a job history item after a successful login.

REQ-18: The system shall allow a tutor to edit a past employment item after a successful login.

REQ-19: The system shall allow a tutor to edit a skill item after a successful login.

1.3 Job Posting

1.3.1 Description and Priority

The system shall provide the feature of posting and editing jobs.

Priority: High

1.3.2 Stimulus/Response Sequences

The student navigates to post a job option or edit job details from a list of current jobs. Both these options will appear on the student dashboard.

1.3.3 Functional Requirements

REQ-1: The system shall enable a student to create a job post.

REQ-2: The system shall require the following mandatory fields during job creation:

1. Job Title

2. Job Description
3. Broad Skills
4. Specific Skills
5. Number of Tutors Wanted
6. Preferred Gender of Tutor
7. Service Required
8. Languages
9. Location
10. Budget
11. Online (Boolean)

REQ-3: The job title will have a length no more than 100 characters and no less than 10 characters

REQ-4: The job description will have a length no more than 800 characters and no less than 10 characters

REQ-5: The system shall allow the student to enter 5 skill broad and specific skills each for a job post.

REQ-6: The number of tutors wanted for a job shall be no more than 1000.

REQ-7: The system shall allow a student to edit all the mandatory fields for a job post.

REQ-8: The system shall recommend tutors using data driven techniques to the student for a specific job post.

REQ-9: The system shall allow the student to delete a job post.

1.4 Viewing and Applying to Jobs

1.4.1 Description and Priority

The system shall allow users to submit job applications.

Priority: High

1.4.2 Stimulus/Response Sequences

The tutor will view a list of jobs on the dashboard and click on the apply option for a job being displayed.

1.4.3 Functional Requirements

REQ-1: The system shall allow a tutor to view jobs posts.

REQ-2: The system shall allow the tutor to filter job posts by using keywords.

REQ-3: The system will sort the jobs visible to a tutor according to the posting date.

REQ-4: The system shall enable a tutor to apply for a job.

REQ-5: The system shall require the following mandatory inputs while submitting a job application:

1. Cover Letter

REQ-7: The length of a cover letter shall be no more than 13000 characters and no less than 500 characters.

1.5 Contract Management

1.5.1 Description and Priority

The system shall allow students to initiate a formal contract with a tutor.

Priority: High

1.5.2 Stimulus/Response Sequences

The student can view a list of tutors through either the search option or applications submitted by the tutors and then initiate a contract with the tutor they desire.

1.5.3 Functional Requirements

REQ-1: The system shall allow a student to view the following details of a tutor profile:

1. Full Name
2. Profile Headline
3. Profile Description
4. Skills
5. Job History Items
6. Past Employment Items

REQ-2: The system shall allow a student to search for tutors using keywords.

REQ-3: The system shall allow the student to initiate a contract with one or more tutors for a particular job post.

REQ-4: The system shall allow a tutor to accept contract request from a student.

REQ-5: The system shall make provision for payment integration to enable a student to pay for a job during the contract period.

REQ-6: The system shall initially provide support for given card vendors:

1. VISA
2. MasterCard

REQ-7: The system shall deduct 5% of the charges from the money deposited by the student.

REQ-8: The system shall deduct 10% of the charges from the money received by the tutor.

REQ-9: The system shall allow both tutor or student to end a contract.

REQ-10: The system shall allow both tutor or student to leave feedback after ending a contract.

REQ-11: The feedback left after the job will require the following inputs:

1. Star Rating Out of 5
2. Comments

REQ-12: The feedback comments will have a length no more than 1000 characters and no less than 10 characters.

1.6 Virtual User Wallets

1.6.1 Description and Priority

The system shall allow users to maintain a wallet having coins of a custom currency which will be used to perform several operations on the platform.

Priority: High

1.6.2 Stimulus/Response Sequences

The student will be asked to pay through wallet coins as a result of viewing the contact details of the tutor without initiating a contract using wallets and for posting jobs. A tutor will be asked to pay through wallet coins as a result of submitting a job application.

1.6.3 Functional Requirements

REQ-1: The system shall allow both tutors and students to pay for the purchase of coins using given card vendors:

1. VISA
2. MasterCard

REQ-2: The system shall provide different packages for purchase of coins.

REQ-3: The system shall enable a user to purchase a custom amount of coins.

REQ-4: The exchange price of coins shall fluctuate based on market considerations.

REQ-5: A student shall be required to pay 15 coins for viewing contact details of a tutor.

REQ-6: A student shall be required to pay 5 coins for sending a message to a tutor who has not applied for the job using the built-in chat system of the platform.

REQ-7: A student shall be required to pay 2 coins for posting a job.

REQ-8: A tutor shall be required to pay 2 coins for applying to a job.

REQ-9: The system shall initially provide 20 coins to both tutor and student upon completing the registration process on the platform.

1.7 Forum Posts

1.7.1 Description and Priority

The system shall allow users to initiate forum posts to initiate discussions regarding different subjects.

Priority: Low

1.7.2 Stimulus/Response Sequences

A student or tutor shall navigate to the Forums option in the navbar to initiate a forum post, track activity on existing ones that the user is participating in and search
for posts using a search bar.

1.7.3 Functional Requirements

REQ-1: The system shall allow either a teacher or student to initiate a forum post.

REQ-2: The system shall allow all three types of users to view a forum post.

REQ-3: The system shall allow all three types of users to search for a forum post.

REQ-4: A forum post shall only become visible after the approval of an admin.

REQ-5: The system shall allow all three types of users to reply to a forum post.

REQ-6: The system shall ask for the following mandatory inputs while initiating a forum post:

1. Title
2. Description

REQ-7: The length of a forum post title shall be no more than 200 characters and no less than 50 characters.

REQ-8: The length of a forum post title shall be no more than 800 characters and no less than 50 characters.

1.8 Admin Panel

1.8.1 Description and Priority

The system shall provide an interface to the admin user enabling him to perform create, read, update and delete operations on various database resources.

Priority: Medium

1.8.2 Stimulus/Response Sequences

The admin panel is an interface that will be displayed to the admin user after a successful login.

1.8.3 Functional Requirements

REQ-1: The system shall allow an admin user to view all the profile details of both teacher and student expect the following ones:

1. Password
2. Payment Method Details

REQ-2: The system shall enable an admin to view all the jobs posted by a student.

REQ-3: The system shall enable an admin to view the all details of every contract associated with a user or tutor profile.

REQ-4: The system shall enable an admin to view all the forum posts.

REQ-5: The system shall enable an admin to view the chat history of every tutor and student.

REQ-6: The system shall enable an admin to suspend any tutor or student account.

REQ-7: The system shall enable an admin to delete any job post.

REQ-8: The system shall enable an admin to delete any forum post.

REQ-9: The system shall enable an admin to delete a reply to any forum post.

REQ-11: The system shall enable an admin to approve all the requests for tutors submitted by the students.

REQ-12: The system shall display a funnel analysis of the platform usage to the admin.

REQ-13: The system shall display the following quantities to the admin using a line charts or plain numbers:

1. Number of New Tutors

2. Number of New Students
3. Number of Deleted Accounts
4. Amount of Revenue Earned

REQ-14: The system shall enable an admin to apply time or location based filters to the quantities specified in **REQ-10.**

1.9 Messaging

1.9.1 Description and Priority

The system shall enable tutors and students to chat with each other prior to and after initiating a contract.

Priority: High

1.9.2 Stimulus/Response Sequences

The student will click on the message option being displayed along the profile of the tutor to message him using the built-in chatting system of the platform.

1.9.3 Functional Requirements

REQ-1: The system shall allow a student to initiate a chat with a tutor.

REQ-2: The system shall allow a student to initiate a chat with a tutor who has applied for a job free of cost.

REQ-3: The system shall allow a tutor to reply to the chat initiated by a student.

REQ-4: Each chat history item will be associated with a specific contract.

REQ-5: Each chat history item will be sorted according to the time of the last message of each chat history item.

1.10 Notification Generations

1.10.1 Description and Priority

The users will be notified in case of several operations performed concerning them on the platform.

Priority: Medium

1.10.2 Stimulus/Response Sequences

The users can navigate to the notification by clicking on the notification icon on the navbar and view the list of notifications as a result.

1.10.3 Functional Requirements

REQ-1: The system shall generate a notification for both tutor and student if their purchase of coins is successful.

REQ-2: The system shall generate a notification for both tutor and student if a contract associated with their account starts.

REQ-3: The system shall generate a notification for both tutor and student if a contract associated with their account ends.

REQ-4: The system shall generate a notification for both tutor and student if they receive a message.

REQ-5: The system shall generate a notification for all three types of users in case there has been activity on a forum post they have created or replied to.

REQ-6: The system shall sort the notifications in descending order according to the time of notification.

REQ-7: The system shall only maintain a record of the past 50 notifications of a user.

Non-Functional Requirements

1.1 Performance Requirements

The application is web-based, so it should be interactive with an easy-to-use interface. It should have the following performance requirements.

1. **Speed**
   - **A.** The system should take no longer than 5 seconds to display the results of a search when the user searches for jobs or tutors.
   - **B.** If records in the system's database are updated or deleted, it should take no longer than 3 seconds, on average.
   - **C.** In normal usage, the system's click response time should not exceed 1 second for 95% of the clicks.
   - **D.** The system should take less than 2 seconds, on average, to refresh the screen of the application.
2. **Size**
   - **A.** The system should be able to accept image files of up to 12 MB for student and tutor profile pictures and other images.
3. **Efficiency**
   - **A.** The application should not experience any lags and should be battery-efficient.

1.2 Safety Requirements

1. The system should only accept passwords of minimum length of 8 characters which can be alphanumeric containing both upper-case and lower-case alphabets.

1.3 Security Requirements

1. An email address and a password should be used to login to the system.
2. The database should store encrypted passwords.

3. The system should be able to protect user data from any unauthorized accesses.
4. The system should use proper authentication mechanisms to protect users' data and profiles.
5. The system should validate all the input fields for script tags and sanitize them before further usage.

1.4 Software Quality Attributes

1. **Adaptability**
   A. The web application should be adaptable to 95% of the screens ranging from 400 x 400 px to 3000 x 3000 px screen resolution.
2. **Maintainability**

   Maintenance must be easy for the web application and efforts should be made to maximize the life of the system.

   A. The system should be able to meet its new requirements easily and should be maintainable and extendable.

   B. Errors and anomalies should be corrected quickly and easily.
3. **Reliability**

   A. The system should be able to handle multiple user records simultaneously without failure. In case of any problem with updating the user's profile in the database, the user should be notified.

   B. The system should have a failure rate of less than 2 per month.

   C. The system should have a mean time to failure of approximately 3 weeks at a minimum.
4. **Portability**

   A. The system should be portable and run on all operating systems including Windows, Linux, MacOS, Android, IOS, etc. as it will be a web-based application.

   B. An internet connection and a web browser should enable the user to access it from anywhere including personal computers and mobile devices.
5. **Scalability**

A. The system should be able to support 5000 users per hour while keeping the click-response time less than 2 seconds.

B. The system should be able to accommodate for 20% growth in the number of users every 6 months.

6. **Correctness**

A. The system should ensure correctness so that the information and results returned by the server should have an error margin of less than 0.5%.

7. **Robustness**

A. The system should be able to make complete recovery in case of any failure within less than 3 hours.

B. The system should have a data corruption probability of less than 0.1% in case of any failure.

C. The system should be able to detect and inform the user when an API or database fails to transmit information to the application.

8. **Usability**

Tutorania is intended to be a modern online marketplace, highly accessible and easily usable by the general public, even the ones with little technical know-how.

A. First-time users of the application should not have any difficulties using it.

B. In addition to being interactive, the user interface should also be aesthetically pleasing.

C. UI/UX principles and laws should be taken into consideration when designing the application.

D. After completing a half-hour training session, the users should be able to perform all system functions.

9. **Availability**

E. The system should be available to the users if they have a stable internet connection.

F. The system should have 99.99% uptime across the region.

10. **Flexibility**

A. It is essential that the overall system should be flexible enough to cope with uncertainty without any problem.
B. It should also be easy to add new features or functionalities to the system.

1.5 Business Rules

1. The system shall allow the admin to take action and ban users in case they violate any rules, regulations or safety protocols.

Software Design Specification

Design Methodology and Software Process Model

Tutorania uses the Agile development methodology/Software Process Model, which is a way to build software that is done in small steps and iterations. In Agile, the requirements and answers change over time as teams that work in different areas work together.

Agile methodology focuses on iterative and incremental development, flexible planning and delivery, collaboration and communication between the development team and the stakeholders, and customer happiness through the continuous delivery of usable software. Agile's iterative and incremental method lets it be flexible and adapt to changing requirements. This makes it a good choice for projects like Tutorania, where requirements change quickly or aren't clear.

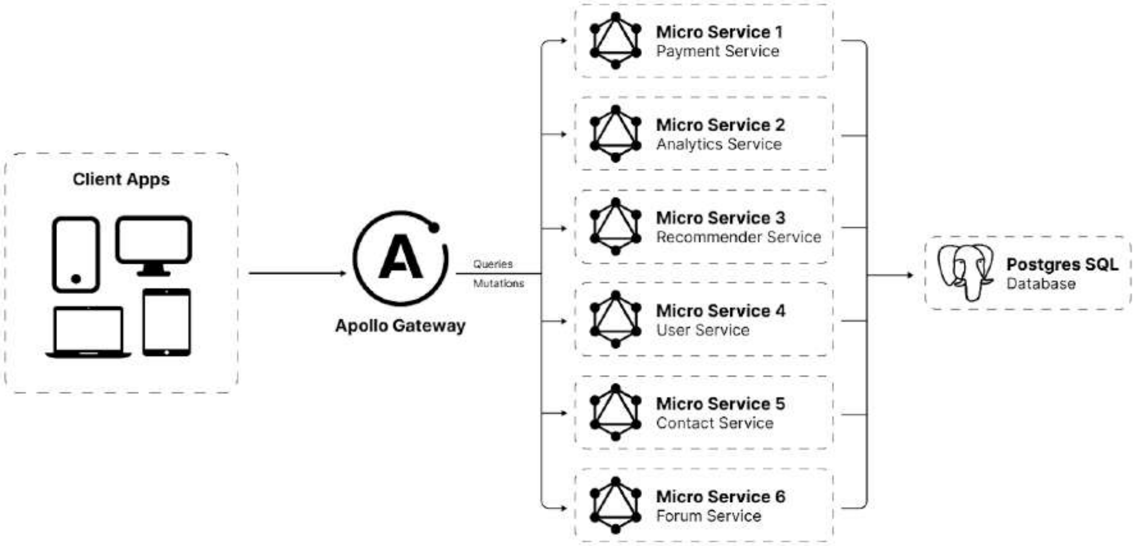
Tutorania uses a mix of Object-Oriented Programming (OOP) and Procedural Programming for its creation process.

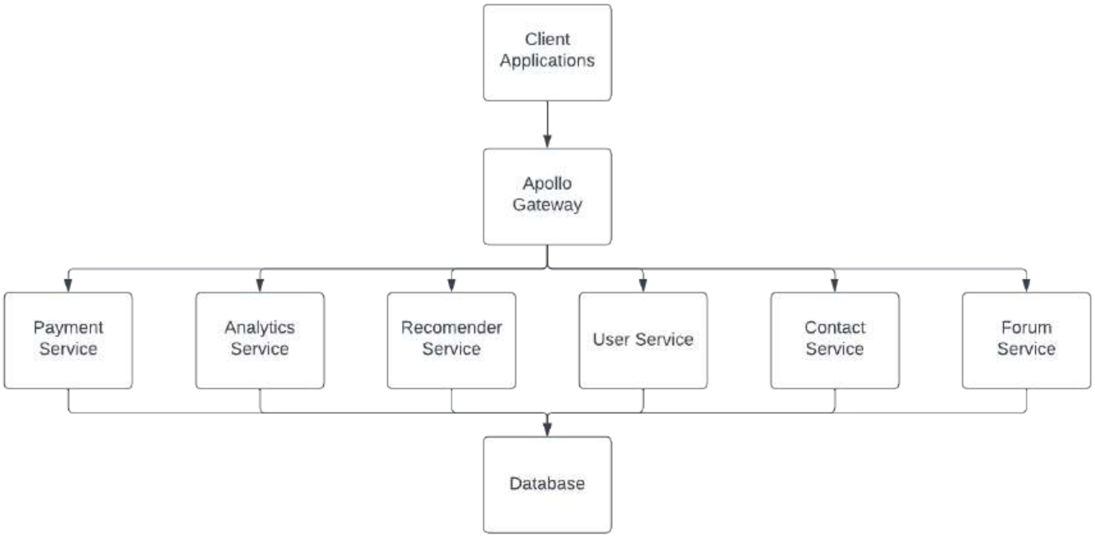
Since the Tutorania project has a lot of functions and features that work together, it makes sense to use OOP because it helps put data and behaviour into objects. This makes the code easier to manage and scale. But the procedural method works better for parts of the job where certain steps need to be done in a certain order.

Architectural Design

Tutorania is a Software as a Service platform that uses Next.js, GraphQL, Postgres, REST API, Apollo, Prisma, and TypeScript, among other things: It uses a design that is scalable and easy to manage. Here are some details:

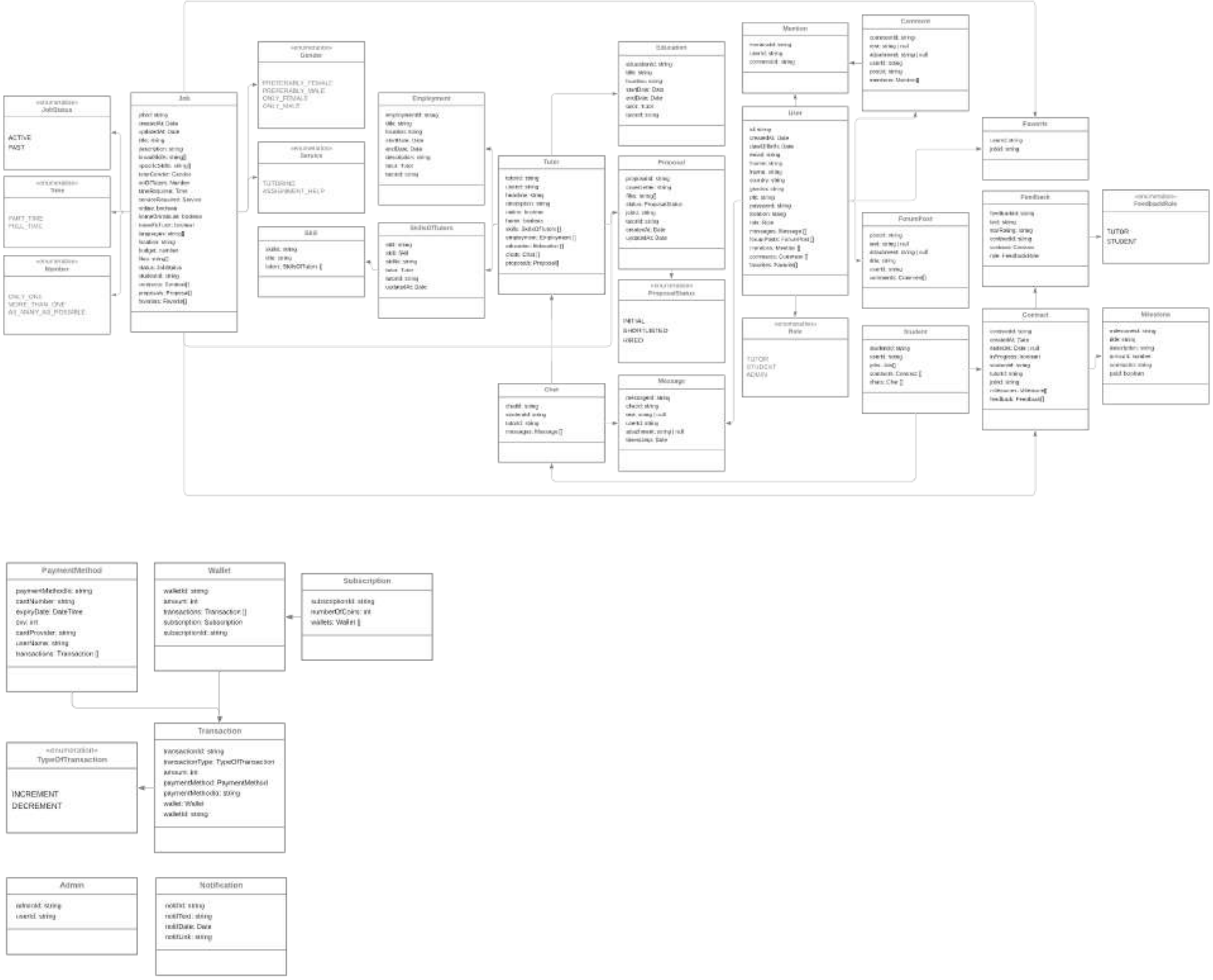
- **Microservices Architecture:** The system is made up of many independent microservices, each of which is in charge of a certain function, such as authentication, profile management, a suggestion engine, contract management, payment integration, and so on. This approach lets each microservice be deployed and scaled in its own way.
- **REST API:** REST APIs let each microservice talk to the other microservices and the database. The REST APIs provide a standard way for microservices to exchange data and keep the services from being too tightly linked.
- **Apollo and GraphQL:** Apollo is used as the API gateway, which gives all the REST APIs a single point of entry. The GraphQL engine is also used by the Apollo server. This lets the client only ask for the data it needs, so it doesn't get too much or too little information.
- **Database:** As the database, Postgres is used, which is a powerful and stable relational database management system. The database is where the system's organised data, like information about users, jobs, contracts, payments, chats, forums and so on, are kept.
- **Frontend:** The frontend is in charge of showing the users the data and handling how the users interact with the site. The Apollo and GraphQL engine allow the frontend to talk to the server to get and change data. For better SEO speed and a higher core web vitals score, the front end uses pre-rendering tools like Static Site Generation and Server Side Rendering.





Class Diagrams

Class diagrams are used to show how the classes, attributes, and functions of a system fit together. They are part of the Unified Modelling Language (UML), which is a standard language for describing, visualising, building, and recording the artefacts of software systems. In software engineering, class diagrams are often used to show how an object-oriented system is put together and how objects and classes relate to each other. They show the links between classes, such as inheritance, aggregation, and association, in a clear way. Then, this knowledge can be used to write code, figure out how a system is put together, and let team members know about design decisions.

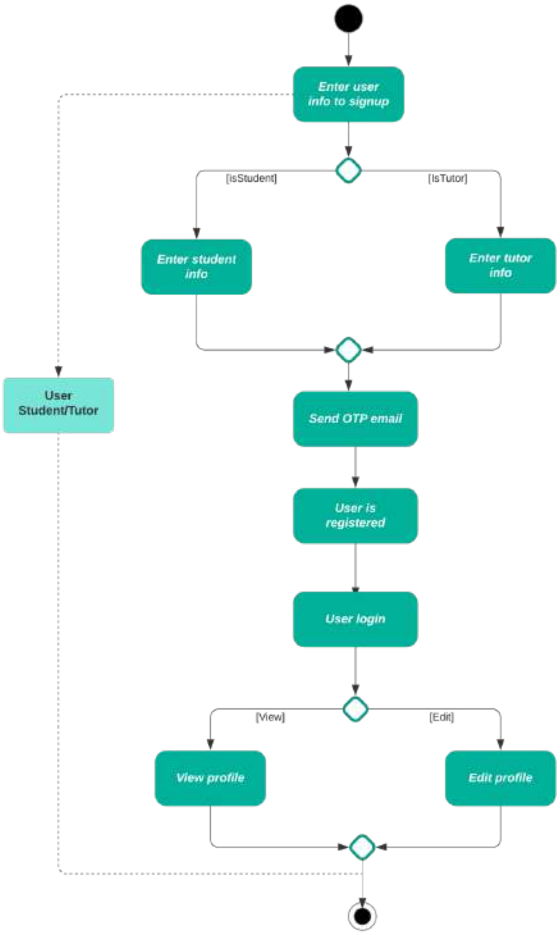


Activity Diagrams

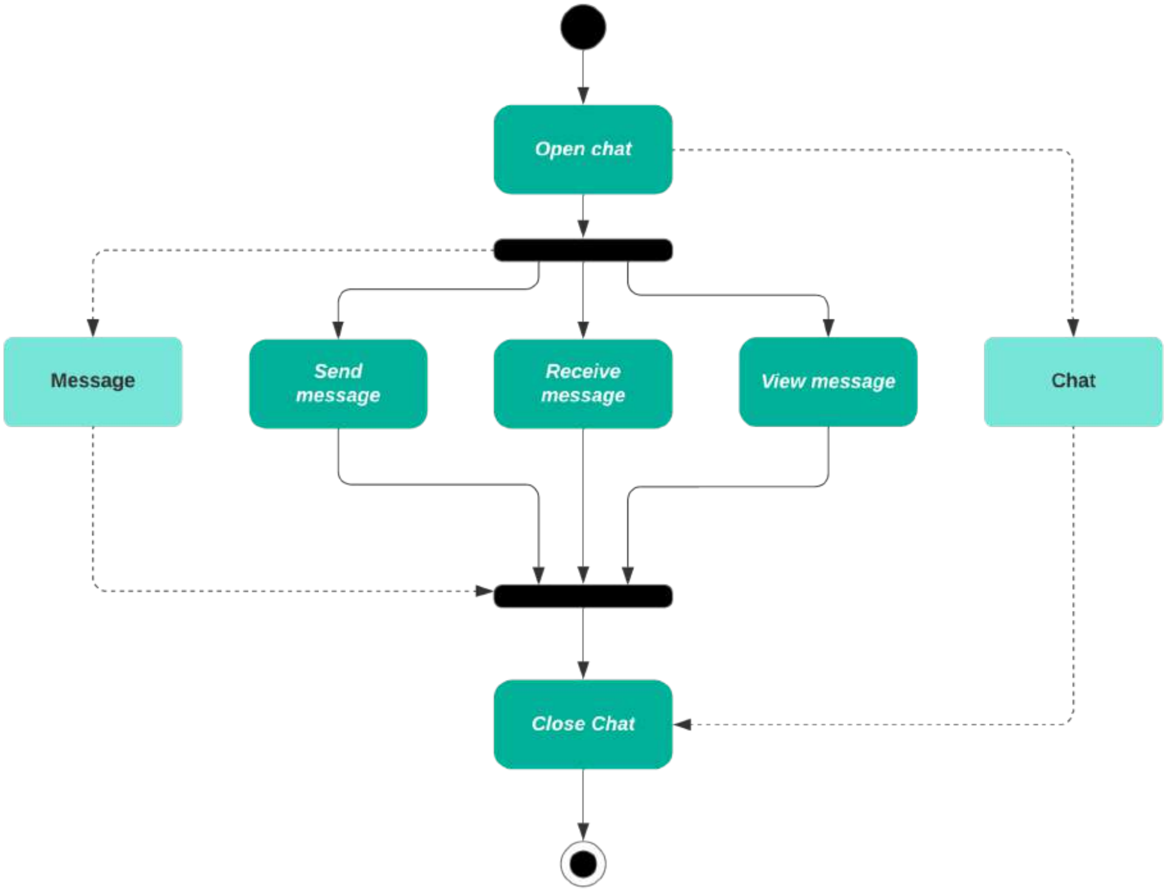
Activity graphs show the flow of jobs or tasks through a system. They are part of UML, a standard language for defining, visualizing, constructing, and recording software system components.

Activity diagrams are often used in software engineering to model business workflows and processes, write down software needs, and make software systems. They are used to show how things are done in a system, and how the different parts of a system affect each other.

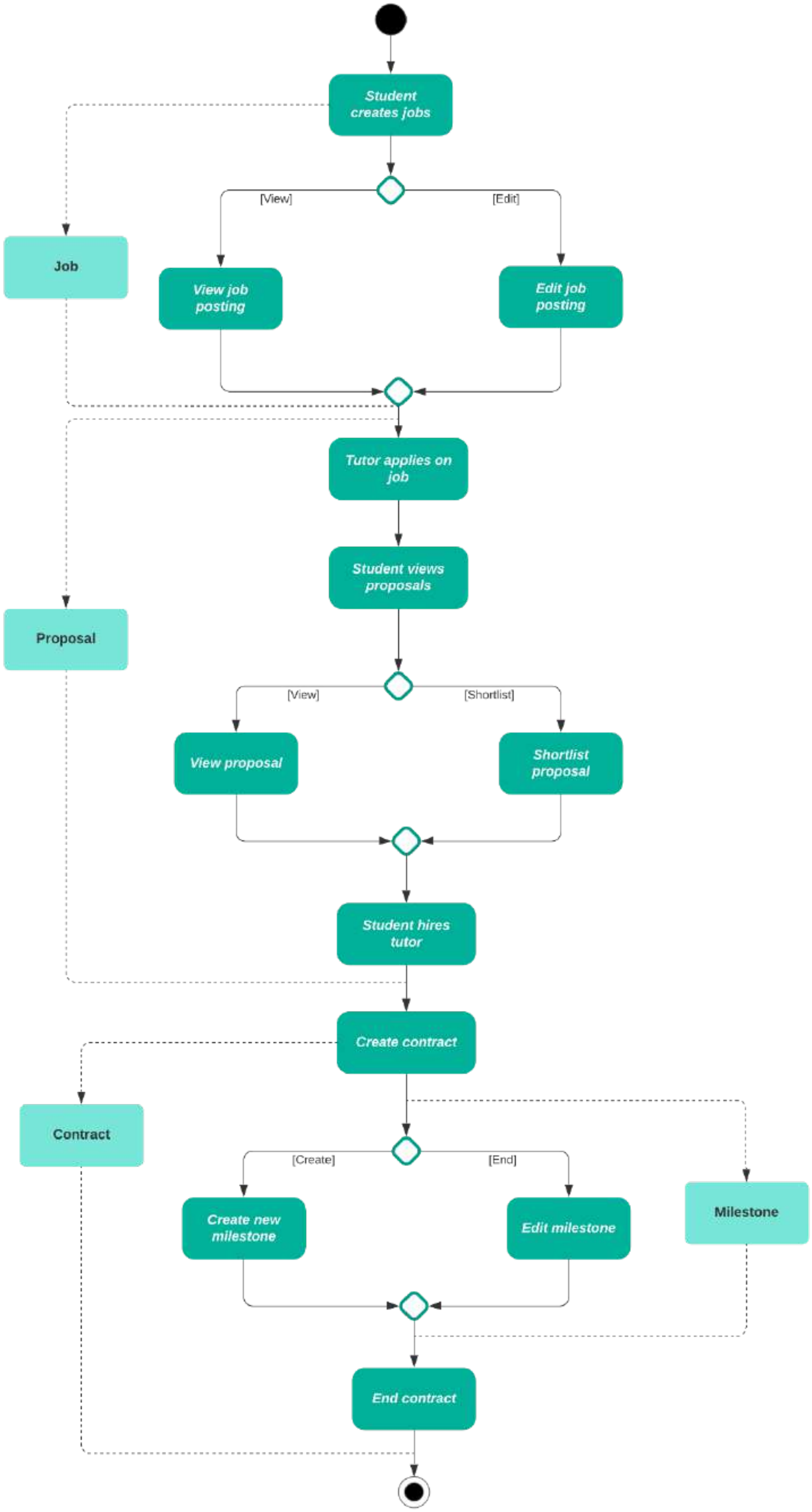
- **User Authentication**



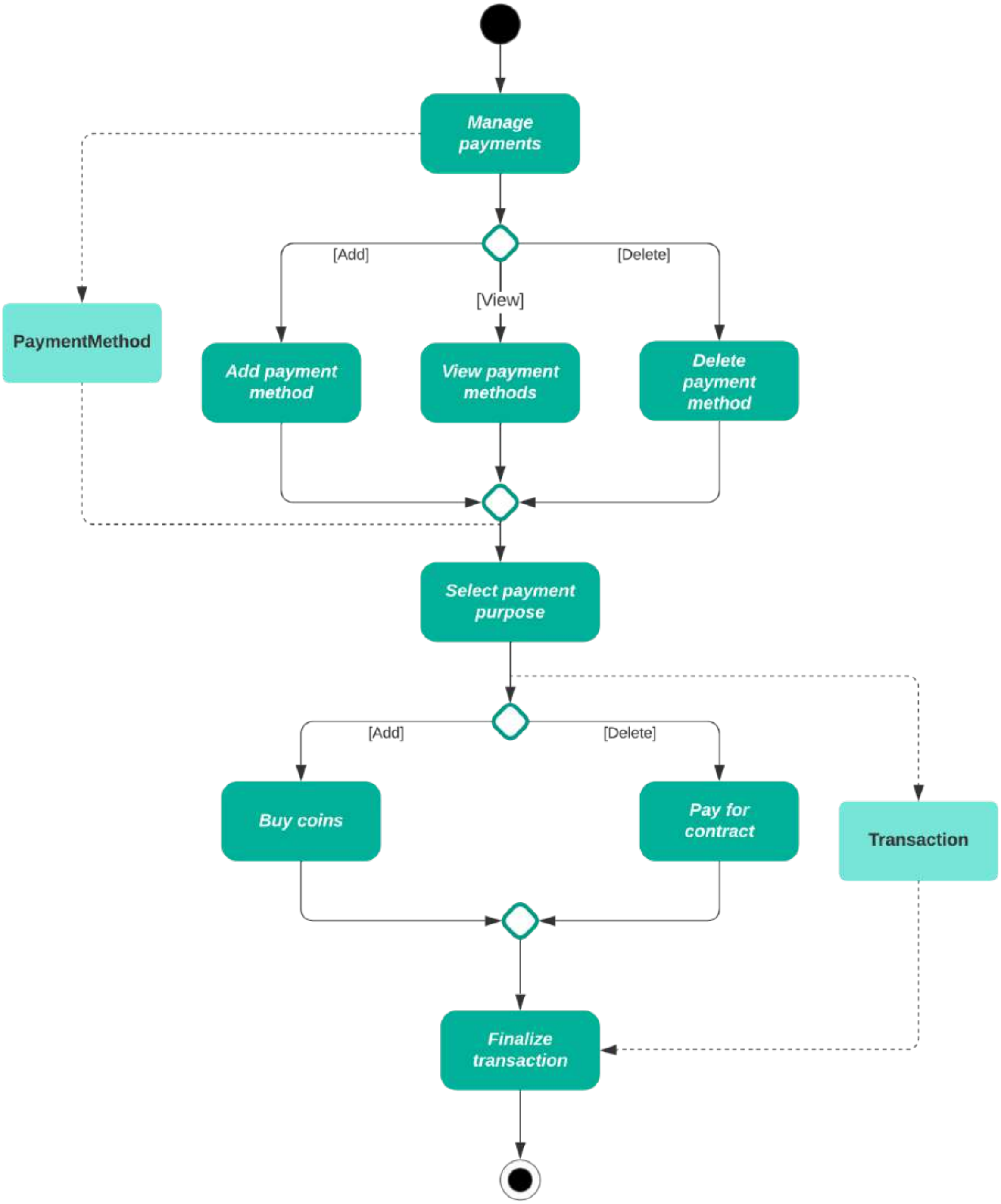
- **Chat Messaging**



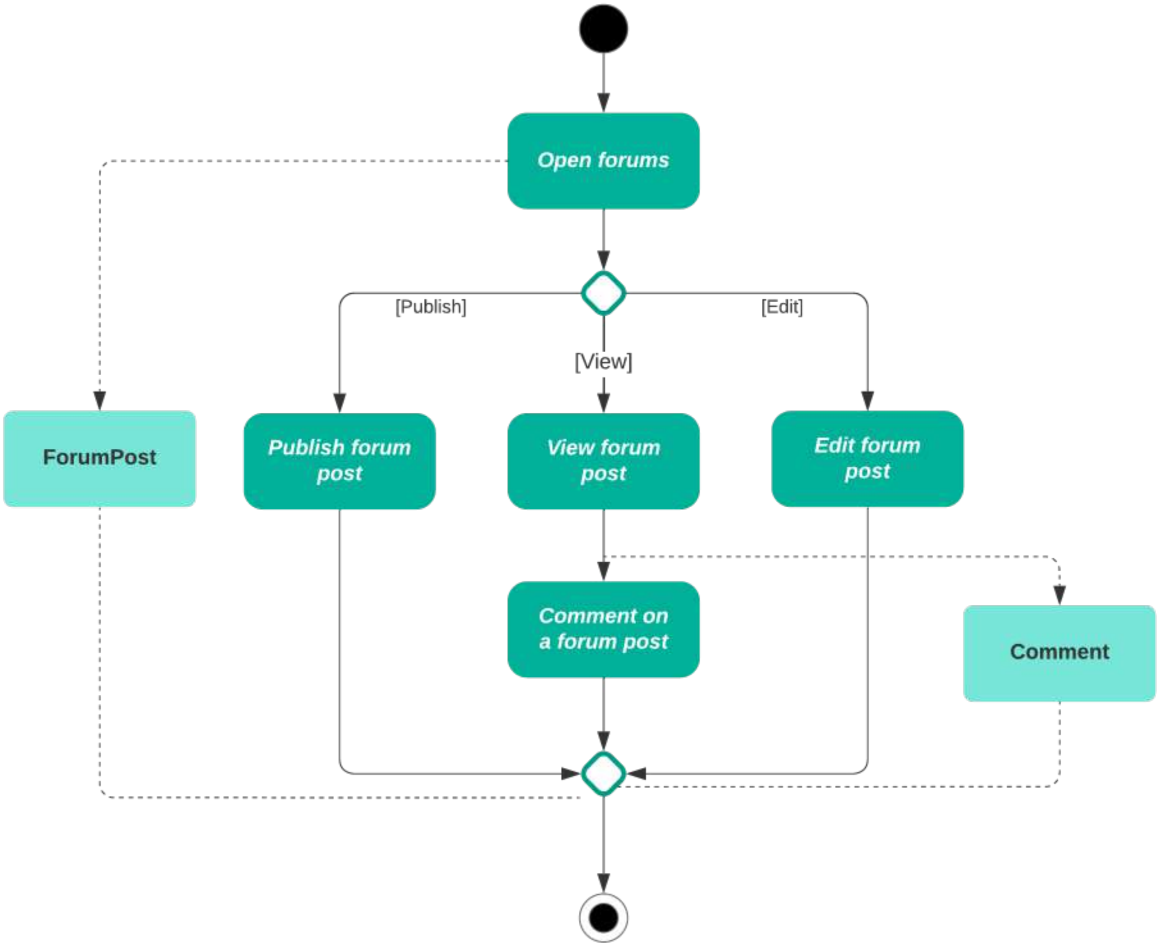
- **Creating Jobs, Proposals and Contracts**


Student creates jobs
[View]
[Edit]
Job
View job posting
Edit job posting
Tutor applies on job
Student views proposals
Proposal
[View]
[Shortlist]
View proposal
Shortlist proposal
Student hires tutor
Create contract
Contract
[Create]
[End]
Create new milestone
Edit milestone
Milestone
End contract

- **Making Payments**



- **Using Forums**



Data Flow Diagrams

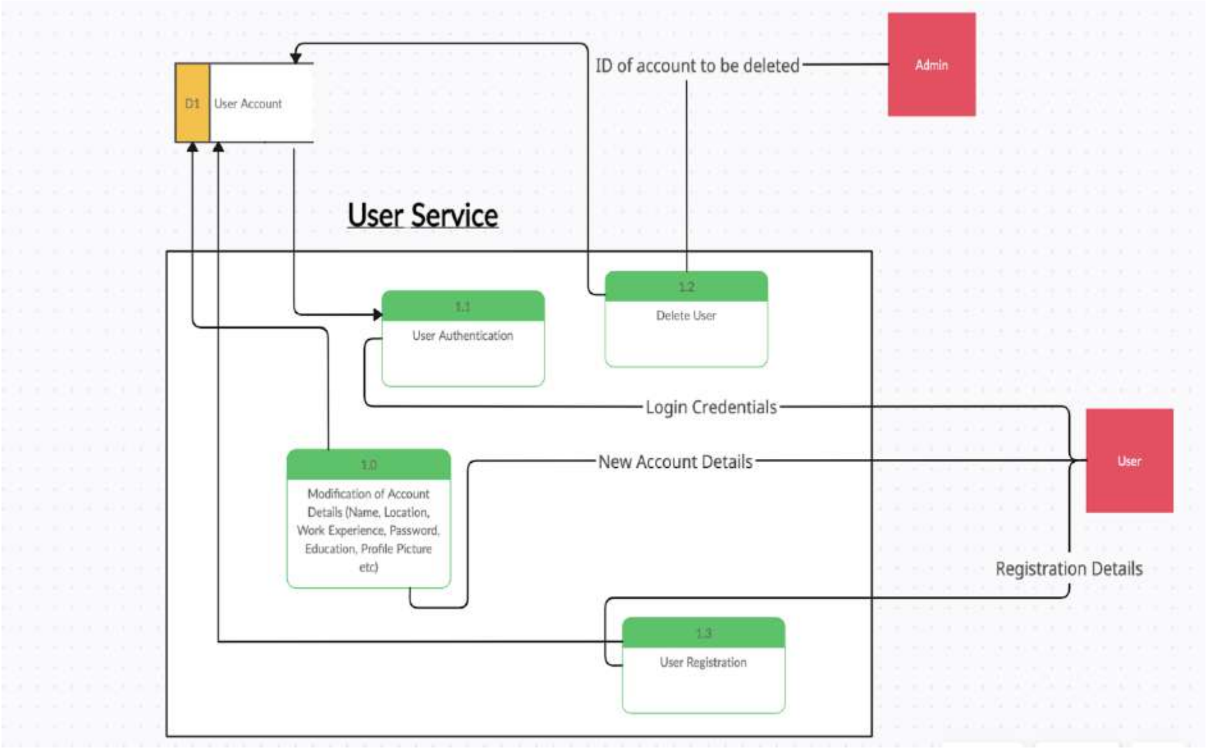
Data flow diagrams (DFDs) show how data moves through a system or process graphically. They are often used in software design specifications. They're useful for pinpointing a system's inputs, processes, and outputs, as well as the connections between them. This aids in communicating the design to stakeholders and the development team by providing a clear and concise representation of the functional needs of the system. Data flow diagrams (DFDs) are comprised of three primary components:

- **Processes:** Internal system changes are known as processes. They are used to illustrate the way in which data is transformed across the system and are depicted as rectangles.

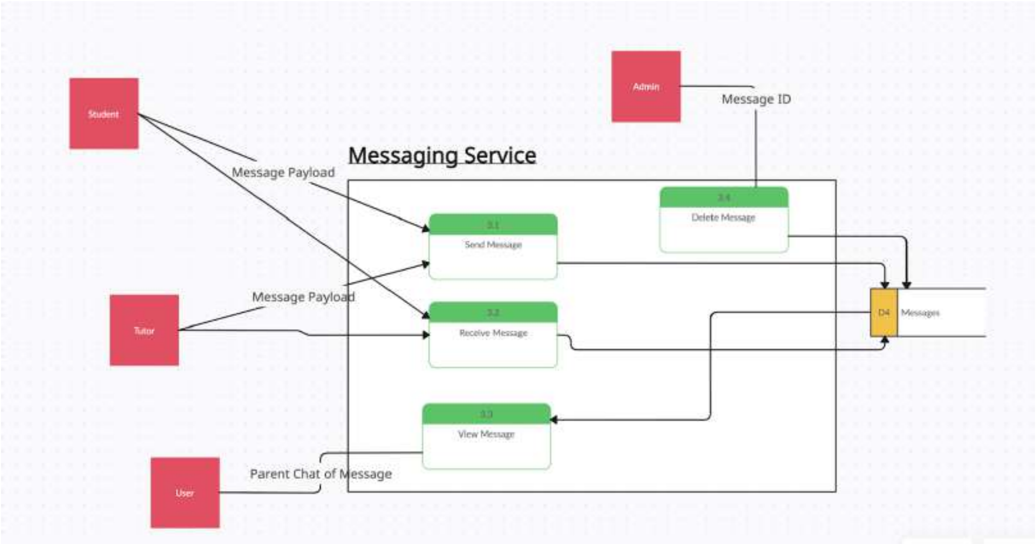
- **Data Flows:** The arrows in a data flow diagram show the flow of information from one part of the system to another. They reveal how one process is related to another, as well as its inputs and outputs.
- **Data Stores:** Data stores are the containers used to store information within the system. They are used to symbolise information that is being temporarily stored or needs to be stored persistently, and they take the shape of rounded rectangles.

Since we are employing a microservices architecture in this project, we will be presenting the DFD for each microservice independently.

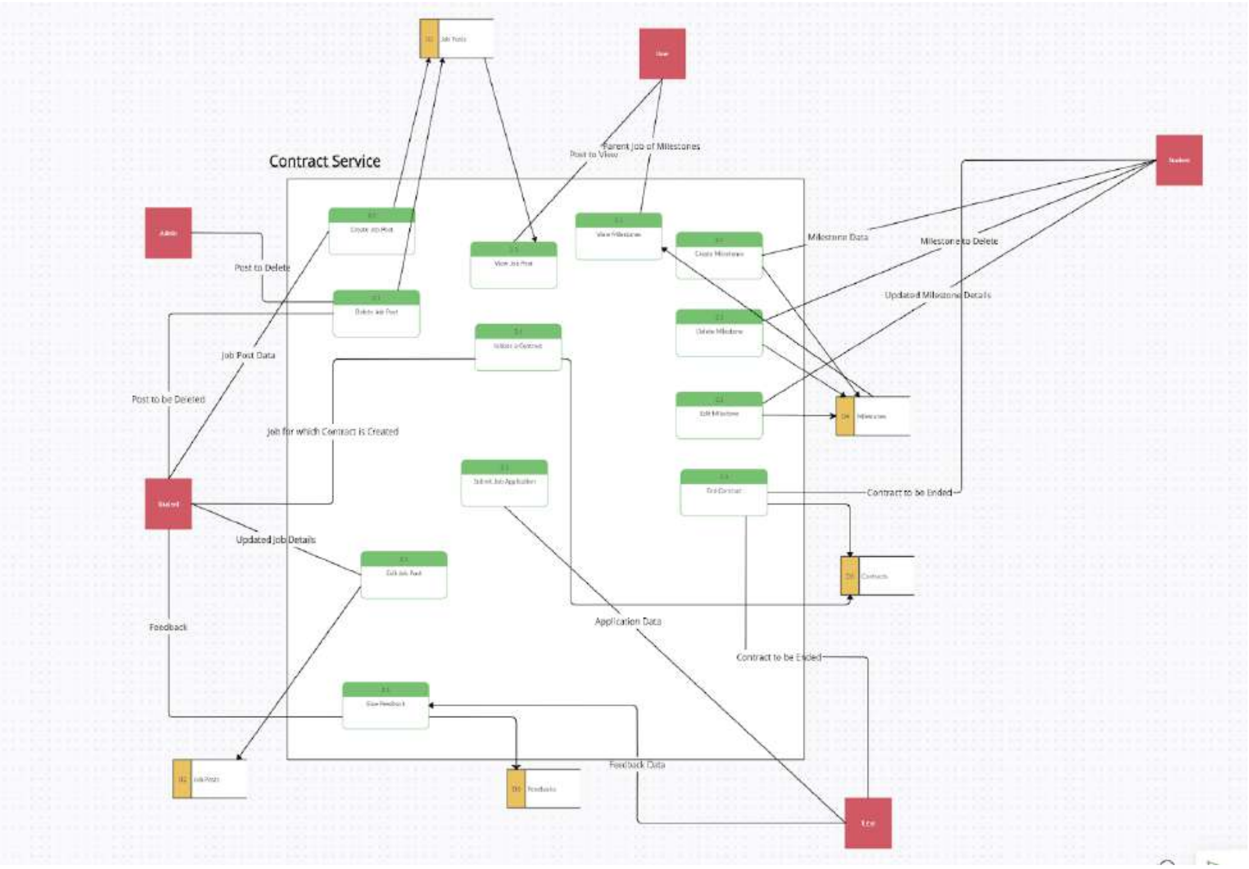
- **User Service**



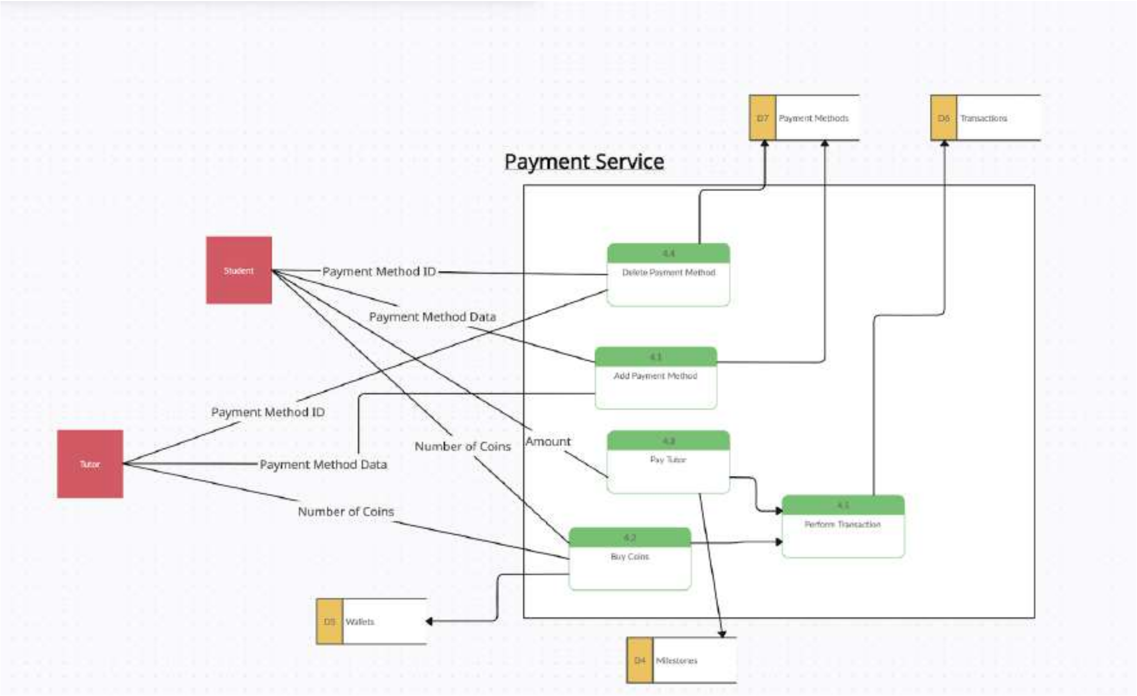
- **Messaging Service**



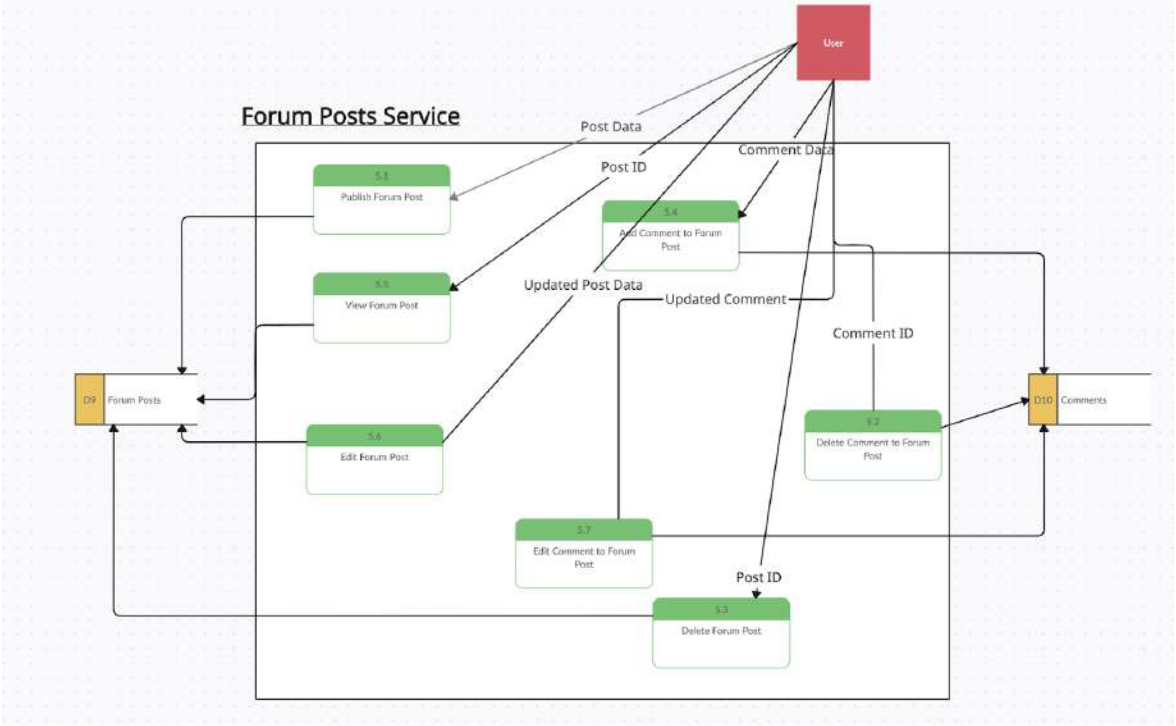
- **Contract Service**



- **Payment Service**



- **Forum Posts Service**



These diagrams can also be viewed in better quality here:

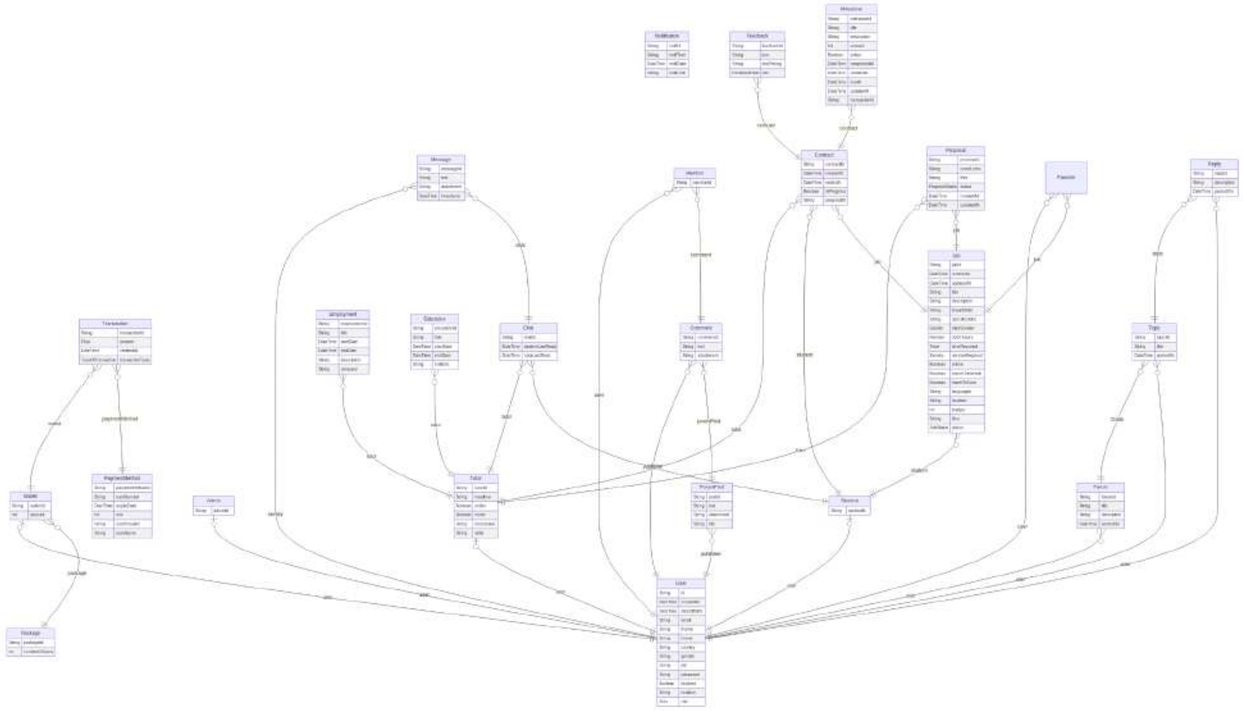
https://drive.google.com/drive/folders/1-wuyKrIgJBN_K5HZZV8j6NV-eI_YC8zl?usp=share_link

Data Design

The given ERD diagram shows how different logical entities have been mapped to database tables, as well as the characteristics and types of each of these tables.

There are the following three kinds of connections between tables:

- One-to-one relationship (1:1)
- One-to-many relationship (1:M)
- many-to-many relationship (M:M)



Also, relationships between different tables have been moved to foreign key relationships or extra tables for many-to-many (M:M) relationships.

For a more detailed view, visit the given link:

https://drive.google.com/file/d/1s84bYNcaneXZf1O6q-hDs2FsvWwz9t2y/view?usp=sharing

Deployment and Integration

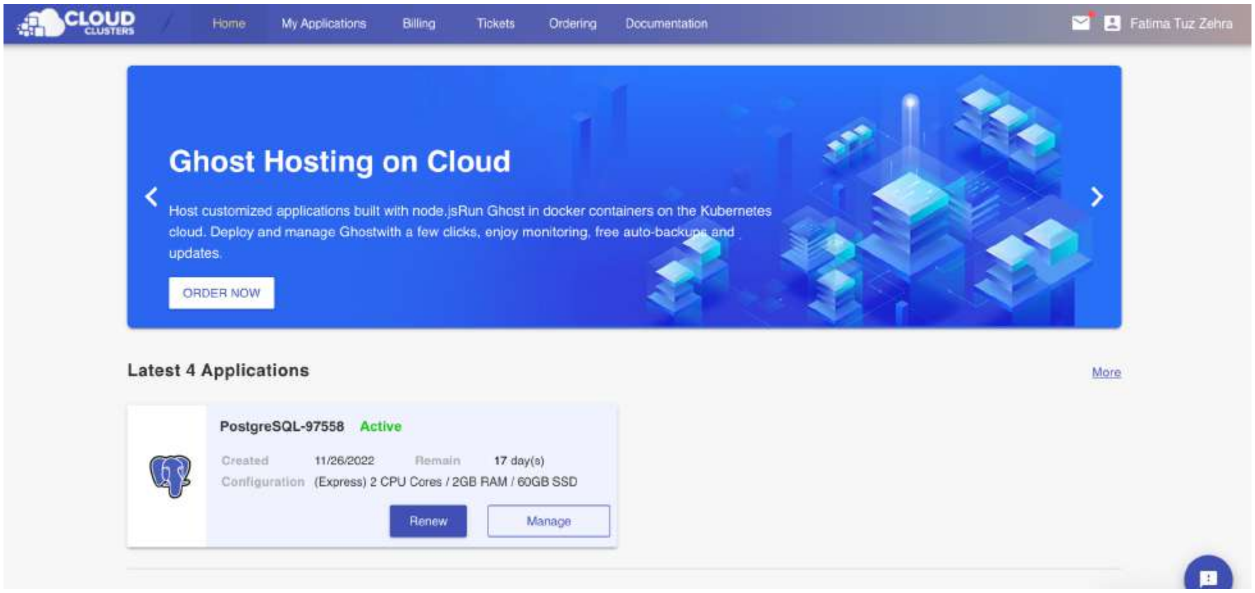
System Integration

On the back end of Tutorania, system interaction is done through an API gateway built with Apollo GraphQL. This architecture makes it easy for the frontend and backend parts of the app to talk to each other. The frontend is responsible for communicating with the backend by submitting Apollo GraphQL client requests to the API gateway. The API gateway is responsible for handling the request routing to the right microservice by utilizing axios. The API gateway is where all calls from clients come together. It is in charge of getting GraphQL requests and mutations from the front end and handling them. The specific data needs from the front end are wrapped up in the Apollo GraphQL client calls. This lets declarative data get work. When a GraphQL request comes in from the front end, the API gateway processes the request and figures out which microservice needs to handle it. This is done based on what action or data is being asked for. A famous HTTP client called axios is used by the API gateway to send the client call to the right microservice.

Tutorania gets a lot out of using Apollo GraphQL and the API gateway pattern. In the first place, it makes it possible to get data in an effective and optimized manner by enabling the frontend to specify the precise data that is required, hence lowering the likelihood of either over-fetching or under-fetching. This makes the system run faster and puts less stress on the network traffic. Secondly, the API gateway makes it easier for the frontend and server to talk to each other. It works as an abstraction layer, hiding the frontend from the complexity of the microservices architecture below. The frontend only needs to talk to the API gateway, and the gateway handles sending calls to the right microservice.

Database Deployment

Instead of a typical locally managed database, pgclusters were used to build Tutorania's database at first. Because of this choice, it will be easier to upgrade and grow the globe. pgclusters is used to control databases and copy them across a group of machines, called "nodes." When changes are made to a distributed database, the data will be the same everywhere and be easy to reach. This makes sure that the data are always up to date and in sync as they are being used.



Ghost Hosting on Cloud

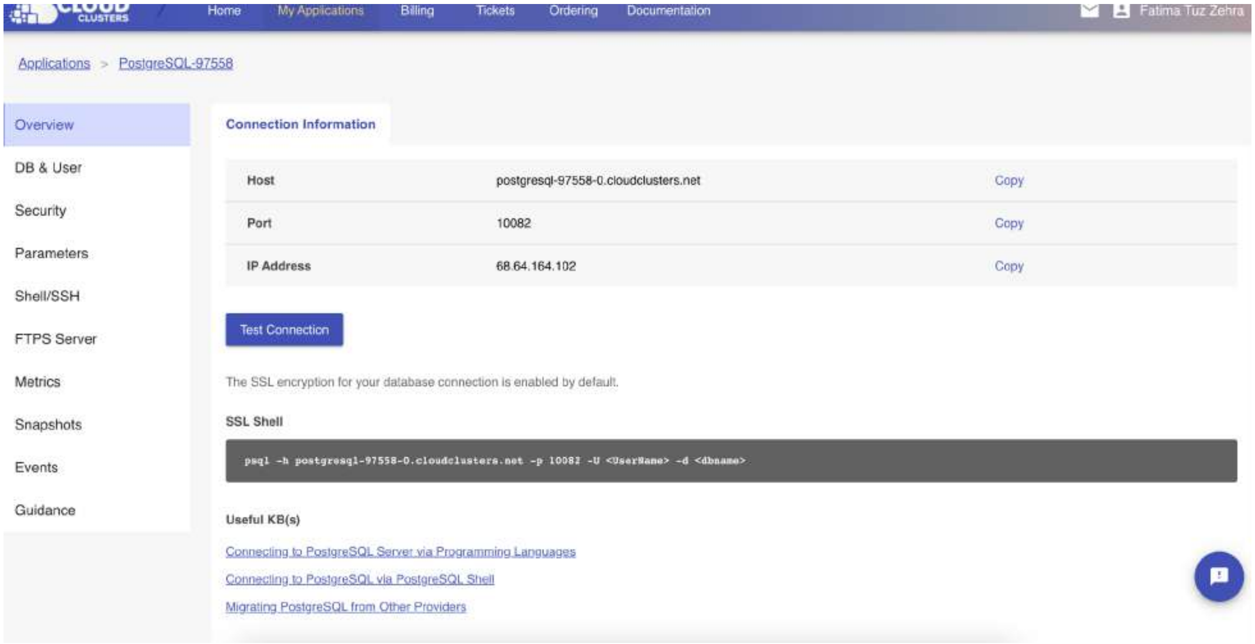
Host customized applications built with node.jsRun Ghost in docker containers on the Kubernetes cloud. Deploy and manage Ghostwith a few clicks, enjoy monitoring, free auto-backups and updates.

Latest 4 Applications

PostgreSQL-97558 Active

Created 11/26/2022 Remain 17 day(s)

Configuration (Express) 2 CPU Cores / 2GB RAM / 60GB SSD



Host	postgresql-97558-0.cloudclusters.net	Copy
Port	10082	Copy
IP Address	68.64.164.102	Copy

The SSL encryption for your database connection is enabled by default.

SSL Shell

```
psql -h postgresql-97558-0.cloudclusters.net -p 10082 -U <UserName> -d <dbname>
```

Useful KB(s)

Connecting to PostgreSQL Server via Programming Languages

Connecting to PostgreSQL via PostgreSQL Shell

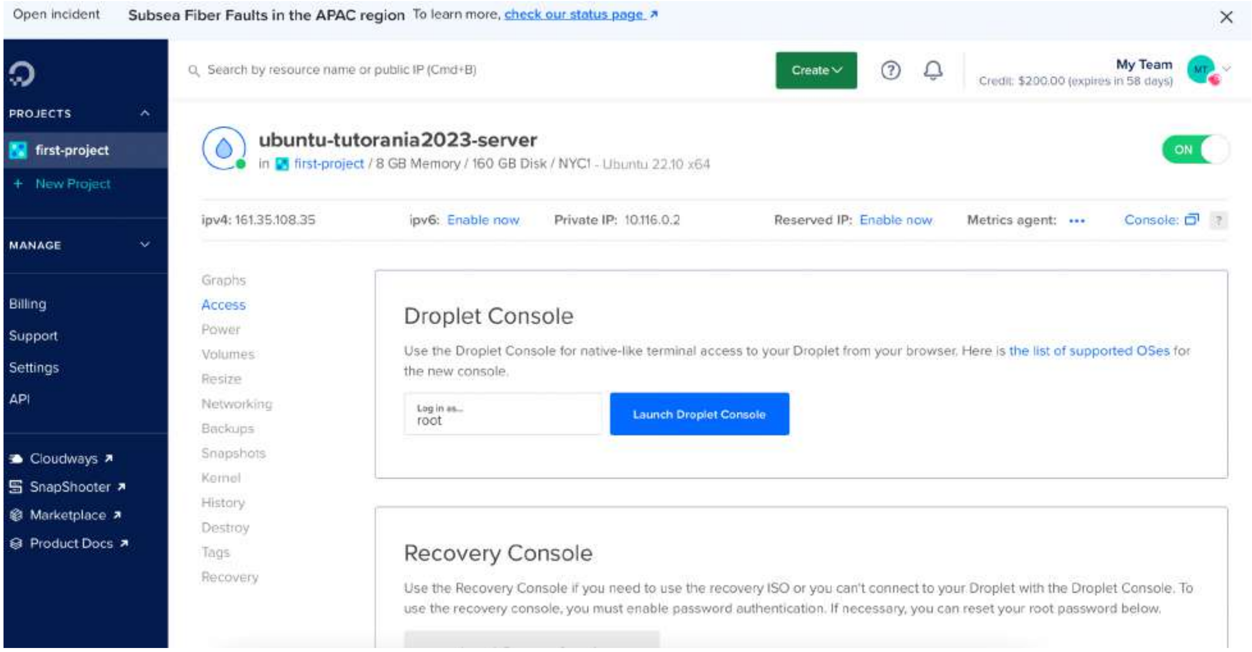
Migrating PostgreSQL from Other Providers

Using pgclusters makes the process of making things easier. With a distributed database design, developers can try out new features and upgrades on different clusters before adding them to the main database. Developers can work on groups or areas by themselves on their own time. This helps people grow faster and makes them more flexible as a whole. The pgclusters database system can keep growing and working regularly even if there is a problem. The database was split into many groups, or "clusters," so that the system could handle the growing number of people and data. Even if one of the database's groups fails, users will still be able to easily get to it. When something bad happens, open systems make it much easier to get back on your feet. In case of a disaster or data loss, data can be retrieved from other groups. This makes sure the data is correct and cuts down on the time the platform is down.

Users in any part of the world can use Tutorania because pgclusters gives them access to information from all over the world. Because data can be seen from any cluster, the person has the same experience no matter where they are. Tutorania's pgclusters database management system has a number of advantages. It makes the facts change right away in all groups. Using parallel work in groups makes the process of making something easier. Scalability, fault tolerance, and disaster recovery all work together to make sure that data security is kept and the system is always up and running. pgclusters makes it possible for Tutorania's database management to meet all of its needs by giving access to data all over the world.

System Deployment

The back end of the Tutorania project is set up on a **DigitalOcean droplet** that runs Ubuntu. On the droplet, the backend's dependencies are loaded, and Docker is used to run all the services at the same time. This method makes it easy to handle the back-end services and makes sure they all run in the same environment. **Similarly, the frontend was also deployed on a different port on the same server.**



Open incident Subsea Fiber Faults in the APAC region To learn more, check our status page

Credit: $200.00 (expires in 58 days)

ubuntu-tutorania2023-server

in first-project / 8 GB Memory / 160 GB Disk / NYC1 - Ubuntu 22.10 x64

ipv4: 161.35.108.35 ipv6: Enable now Private IP: 10.116.0.2 Reserved IP: Enable now Metrics agent: ••• Console:

Droplet Console

Use the Droplet Console for native-like terminal access to your Droplet from your browser. Here is the list of supported OSes for the new console.

Log in as... root

Recovery Console

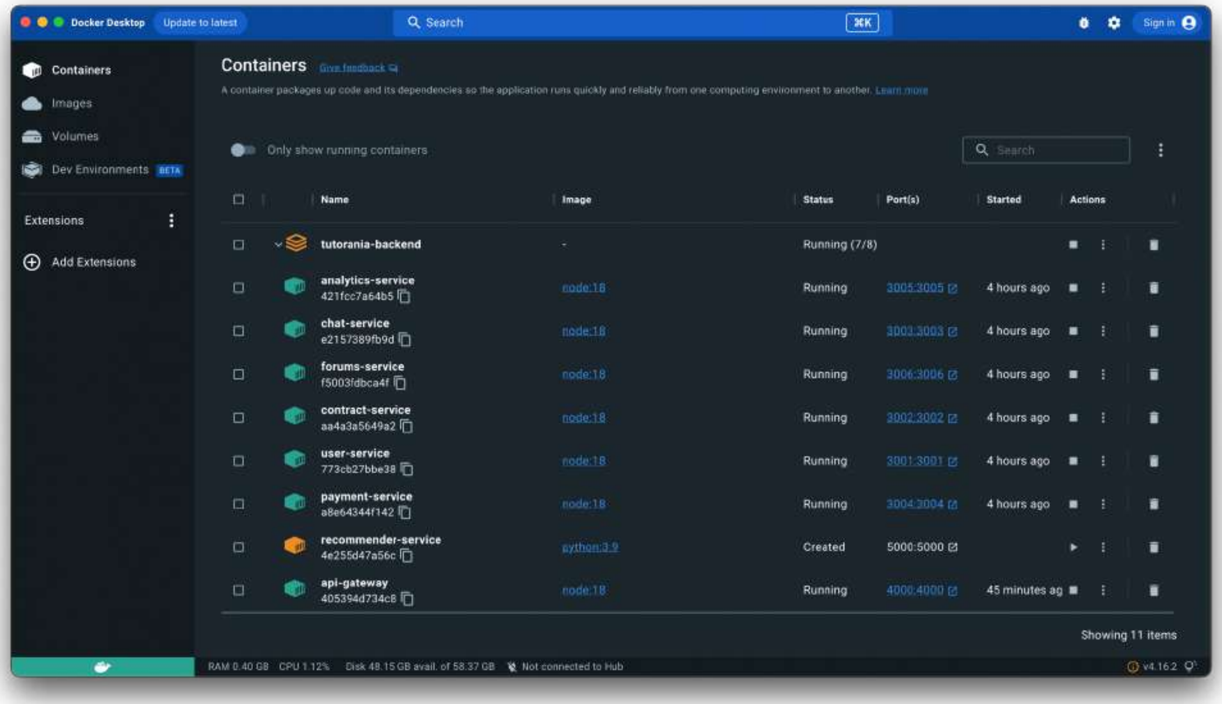
Use the Recovery Console if you need to use the recovery ISO or you can't connect to your Droplet with the Droplet Console. To use the recovery console, you must enable password authentication. If necessary, you can reset your root password below.

Containerization

Docker was used to containerize each microservice from Tutorania. After the requirements were loaded, a Docker Compose file was made with the commands for the microservices. With this approach, the backend is put through its paces. There is no need to change the individual microservices.

Docker containers can hold applications and all of their dependencies, so they can run in any setting. Docker Compose is in charge of how containerized apps are defined and run. This paper talks about what application services are and how they work together.

The Tutorania Docker Compose file has a list of background services and what they need. The requirements of a microservice must be set up first, before the service itself is run. This way of doing things makes sure that every microservice works in a consistent context and installs any dependencies it needs. Docker and Docker Compose are tools that make it easier for microservices to scale, be maintained, and change. It makes creation and maintenance easier, and it makes it easier to use resources and cut costs.



Containers

A container packages up code and its dependencies so the application runs quickly and reliably from one computing environment to another. Learn more

Only show running containers

Name	Image	Status	Port(s)	Started
tutorania-backend	-	Running (7/8)		
analytics-service 421fcc7a64b5	node:18	Running	3005:3005	4 hours ago
chat-service e2157389fb9d	node:18	Running	3003:3003	4 hours ago
forums-service f5003fdbca4f	node:18	Running	3006:3006	4 hours ago
contract-service aa4a3a5649a2	node:18	Running	3002:3002	4 hours ago
user-service 773cb27bbe38	node:18	Running	3001:3001	4 hours ago
payment-service a8e64344f142	node:18	Running	3004:3004	4 hours ago
recommender-service 4e255d47a56c	python:3.9	Created	5000:5000	
api-gateway 405394d734c8	node:18	Running	4000:4000	45 minutes ag

Showing 11 items

RAM 0.40 GB CPU 1.12% Disk 48.15 GB avail. of 58.37 GB Not connected to Hub v4.16.2

System Testing

Testing Plan

A testing plan needs to be put in place to make sure that the Tutorania application works as planned and meets the needs of the users. The testing plan has several steps, such as testing the back-end API with Postman and Swagger, testing the API Gateway with Apollo GraphQL Playground, testing the system and integration with Selenium, and testing the load with JMeter. By sticking to this testing plan, the Tutorania application can be tested in depth and made sure to work as intended.

Backend API Testing and Unit Testing using Postman / Swagger, Jest and Supertest

- Postman is a platform that is often used to create, build, and test APIs. It has a user interface that is easy to understand and use, and it has a lot of tools for testing REST APIs.

- Swagger is another HTTP client used to test and document APIs. It simplifies REST API testing.

- To ensure Tutorania's backend APIs operate, utilise Postman and Swagger. Test all CRUD (create, retrieve, update, and delete) actions.

- Jest and Supertest can develop test suites and generate code coverage reports for API routes in microservices.

API Gateway GraphQL Testing using GraphQL Playground

- Apollo GraphQL is a well-known place to make and test GraphQL APIs. Apollo also has GraphQL Playground, which is a simple way to test out GraphQL APIs.

- Tutorania's API gateway can be tested with GraphQL Playground to make sure it works as planned. In GraphQL, we should test all queries and mutations.

System and Integration Testing using Selenium

- Selenium tests web apps. It automates browsers and tests web applications.

- Selenium can test Tutorania's system and interface to ensure proper operation and test all user flows and scenarios.

Load Testing using JMeter

- JMeter tests web apps' traffic handling. It simulates many people and tests web applications with different loads.

- JMeter can be used to test Tutorania's load capacity. All user flows and circumstances that may occur with many users should be tested.

In short, the testing plan for the Tutorania app includes testing the backend API with Postman and Swagger, testing the API Gateway GraphQL with Apollo GraphQL UI, testing the system and interaction with Selenium, and testing the load with JMeter. These testing tools will make sure the app works as expected, works well when it's busy, and meets the needs of the users.

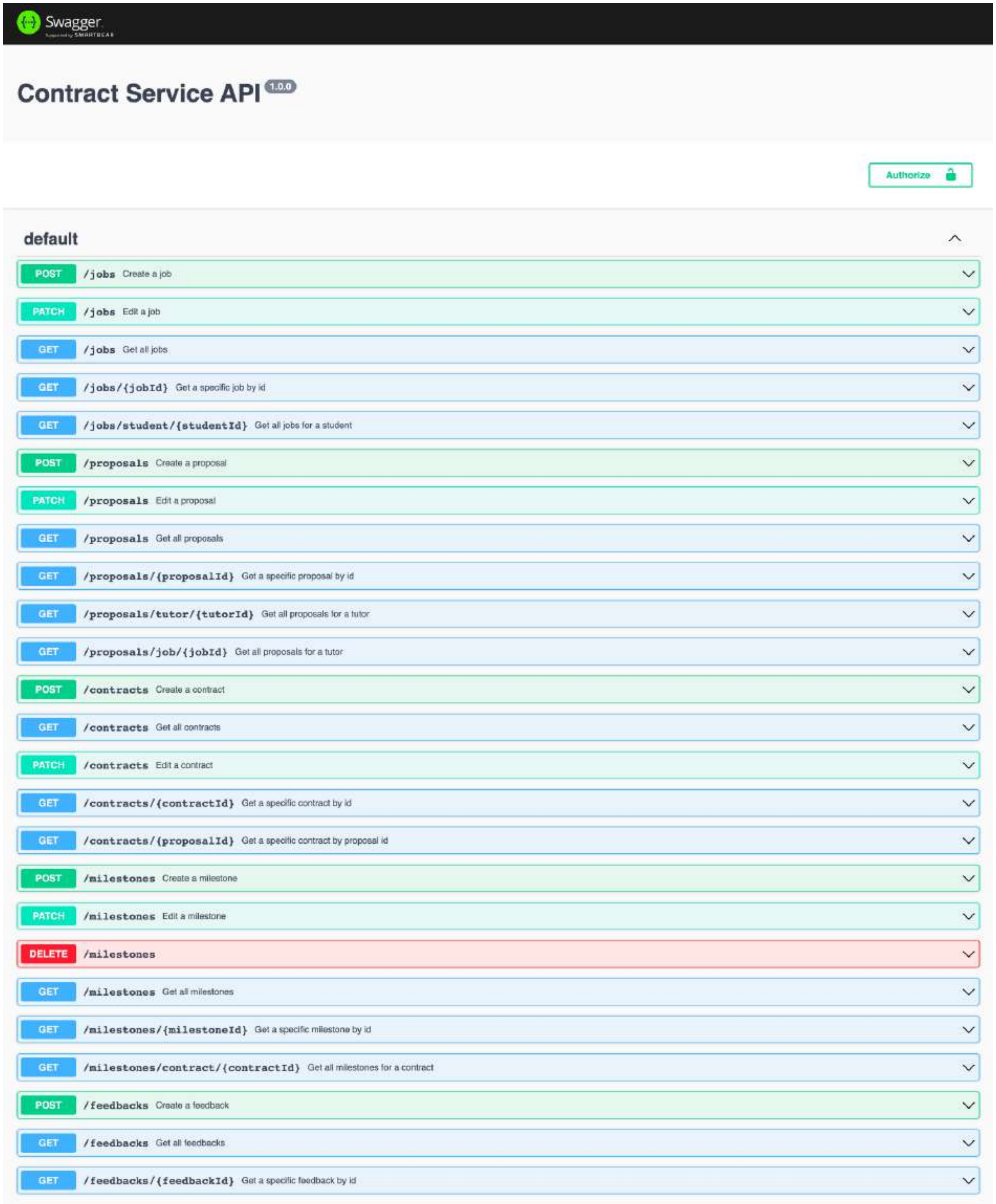
Unit Testing

Unit testing is an important step in the process of developing Tutorania application. It means checking each part of the application that can be tested to see if each piece of code works as it should. Unit testing is generally done by writing test cases that run individual functions, methods, and classes. This makes it easy to fix bugs before putting the code into the main code.

For Tutorania, the backend development consisted of a sequence of important steps. First of all the CRUD routes for each entity were registered in the corresponding microservice. Their correct operation was ensured by documenting and testing them instantly by using Swagger. Then the route was registered in the API gateway which redirects the request to the corresponding microservice. In API gateway, GraphQL schemas or type definitions were defined and corresponding queries and mutations were added, then the resolvers were added and then finally registered in the corresponding microservice adapter. Finally each microservice was fully tested with separate test files for each written using Jest and Supetest to ensure maximum code coverage. The detail of each testing tool is given further.

Unit testing was done with Jest and Supertest to make sure that all API routes in Tutorania work well. Jest is a testing tool built on JavaScript that lets you test both front-end and back-end apps. It is mostly made for apps that use React, but you can write automation scenarios for any JavaScript-based codebases with it. Supertest is a library for Node.js that lets you test HTTP requests and replies. It is used to test API endpoints and make sure they work the way they should. For each microservice, separate test files were made to make sure that the CRUD operations work correctly. This method made it easy to test each part and make sure it worked the way it was supposed to.

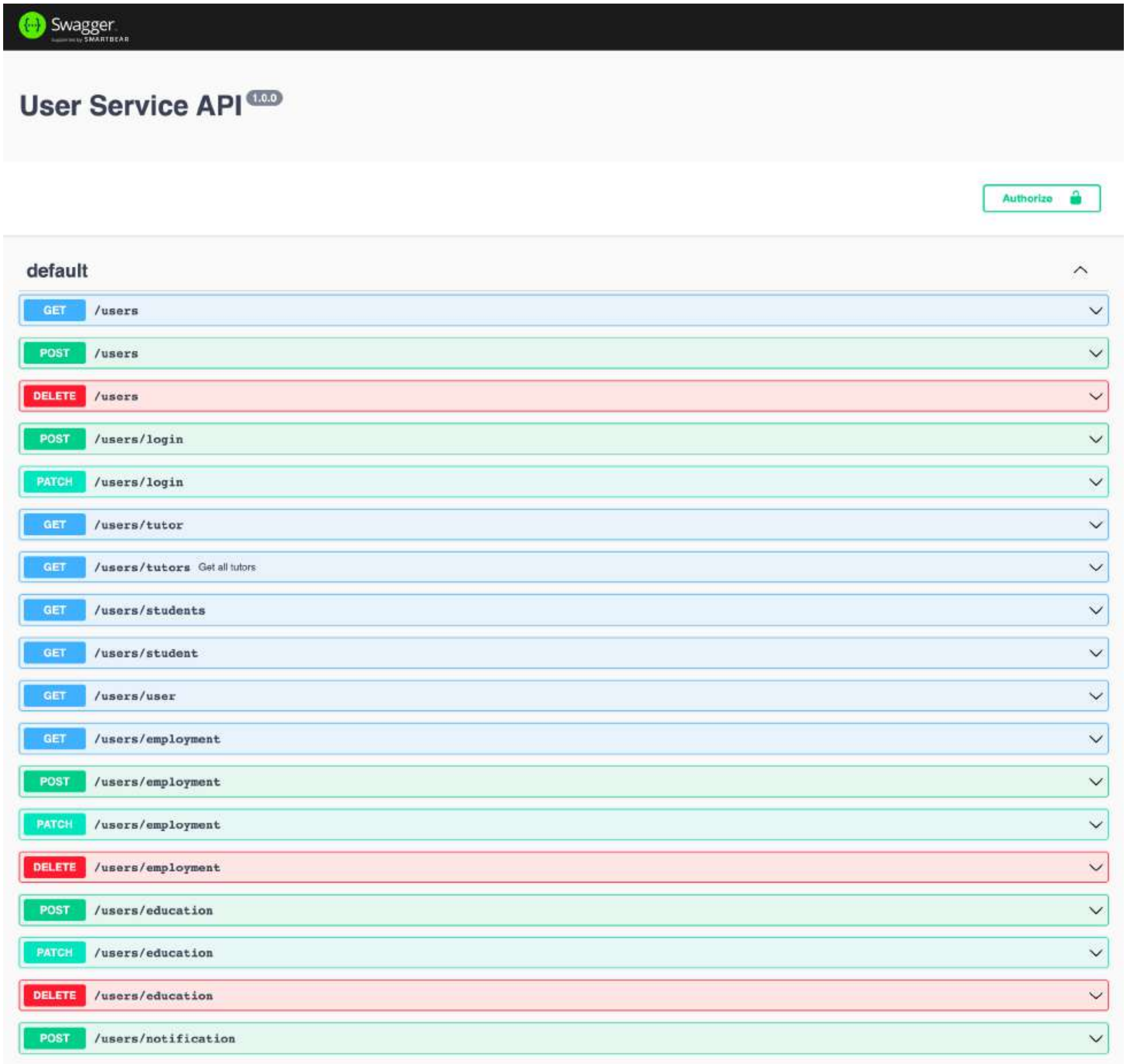
Along with Jest and Supertest, Swagger was also used to test APIs. Swagger is a free tool for developers that helps them plan, build, document the requests and responses, and test RESTful APIs.



Contract Service API 1.0.0

default

POST /jobs Create a job
PATCH /jobs Edit a job
GET /jobs Get all jobs
GET /jobs/{jobId} Get a specific job by id
GET /jobs/student/{studentId} Get all jobs for a student
POST /proposals Create a proposal
PATCH /proposals Edit a proposal
GET /proposals Get all proposals
GET /proposals/{proposalId} Get a specific proposal by id
GET /proposals/tutor/{tutorId} Get all proposals for a tutor
GET /proposals/job/{jobId} Get all proposals for a tutor
POST /contracts Create a contract
GET /contracts Get all contracts
PATCH /contracts Edit a contract
GET /contracts/{contractId} Get a specific contract by id
GET /contracts/{proposalId} Get a specific contract by proposal id
POST /milestones Create a milestone
PATCH /milestones Edit a milestone
DELETE /milestones
GET /milestones Get all milestones
GET /milestones/{milestoneId} Get a specific milestone by id
GET /milestones/contract/{contractId} Get all milestones for a contract
POST /feedbacks Create a feedback
GET /feedbacks Get all feedbacks
GET /feedbacks/{feedbackId} Get a specific feedback by id



User Service API 1.0.0

default

GET /users

POST /users

DELETE /users

POST /users/login

PATCH /users/login

GET /users/tutor

GET /users/tutors Get all tutors

GET /users/students

GET /users/student

GET /users/user

GET /users/employment

POST /users/employment

PATCH /users/employment

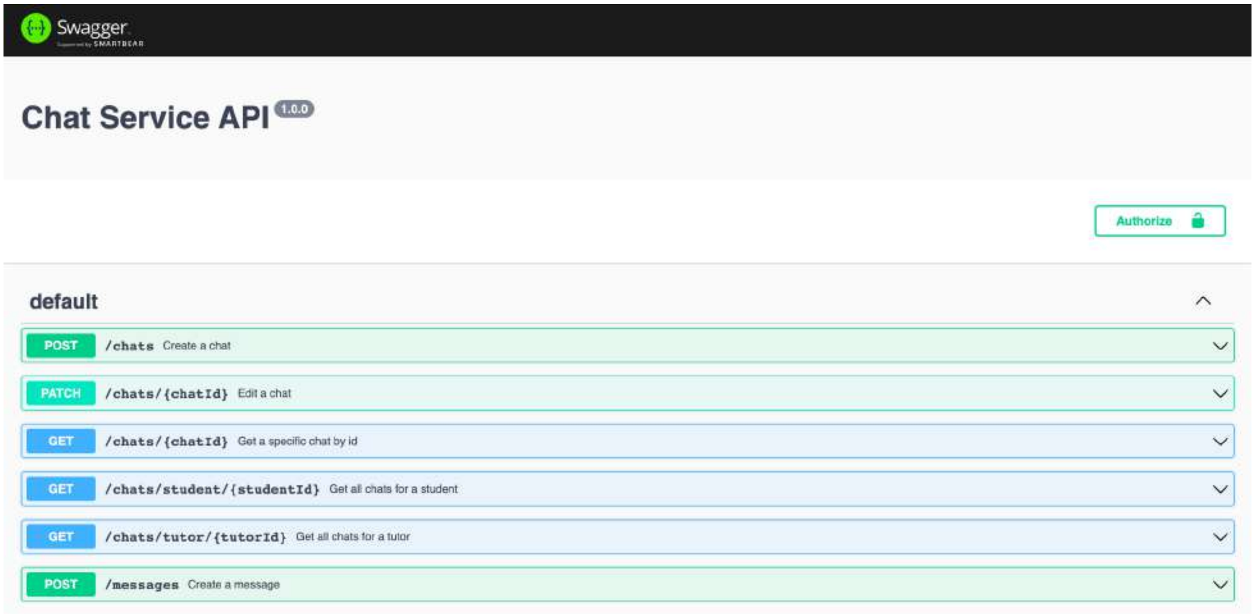
DELETE /users/employment

POST /users/education

PATCH /users/education

DELETE /users/education

POST /users/notification



Chat Service API 1.0.0

default

POST /chats Create a chat

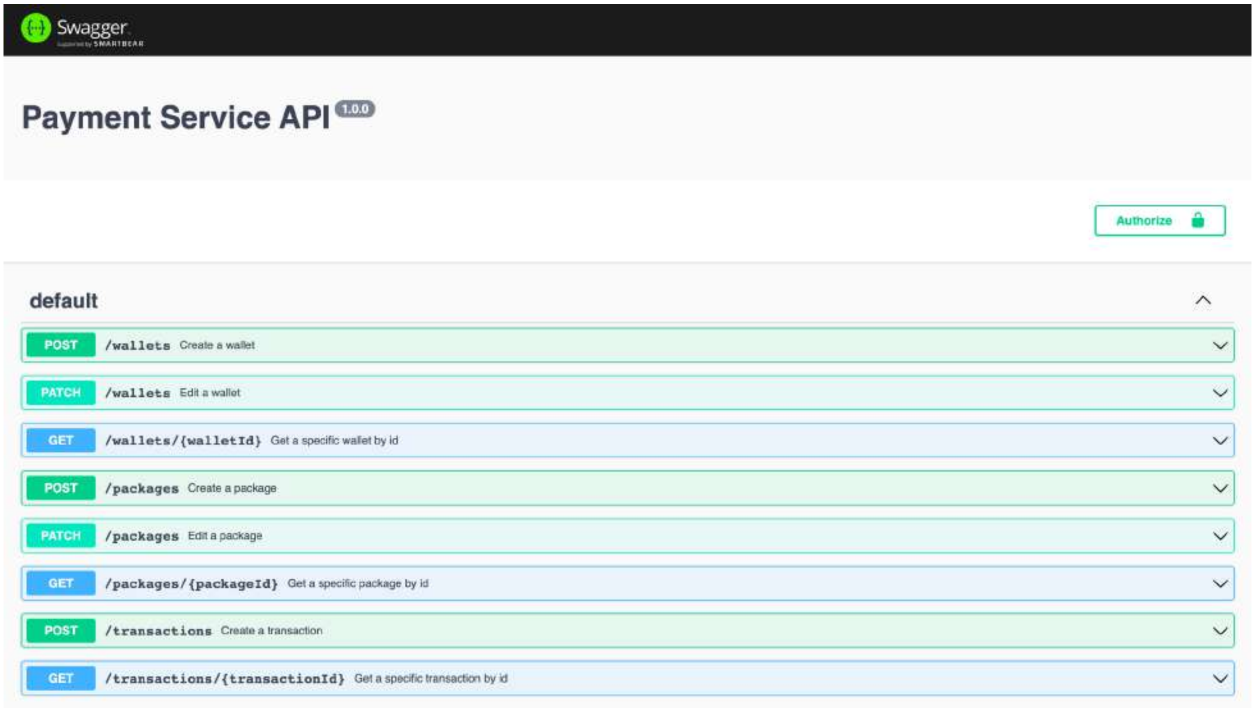
PATCH /chats/{chatId} Edit a chat

GET /chats/{chatId} Get a specific chat by id

GET /chats/student/{studentId} Get all chats for a student

GET /chats/tutor/{tutorId} Get all chats for a tutor

POST /messages Create a message



Payment Service API 1.0.0

default

POST /wallets Create a wallet

PATCH /wallets Edit a wallet

GET /wallets/{walletId} Get a specific wallet by id

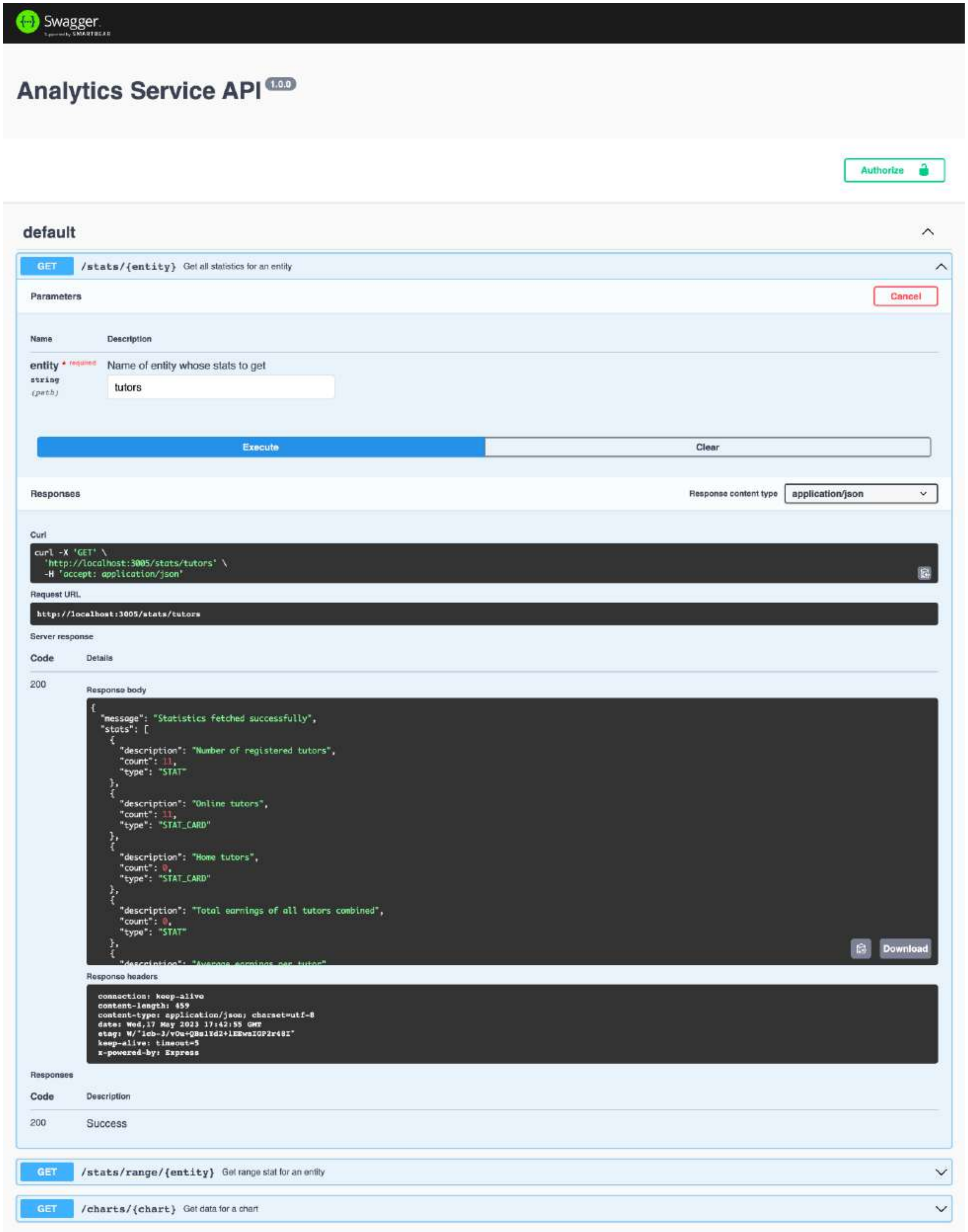
POST /packages Create a package

PATCH /packages Edit a package

GET /packages/{packageId} Get a specific package by id

POST /transactions Create a transaction

GET /transactions/{transactionId} Get a specific transaction by id



Analytics Service API 1.0.0

default

GET /stats/{entity} Get all statistics for an entity

Parameters

Name	Description
entity * required string (path)	Name of entity whose stats to get tutors

Responses

Response content type: application/json

Curl

```
curl -X 'GET' \
  'http://localhost:3005/stats/tutors' \
  -H 'accept: application/json'
```

Request URL

```
http://localhost:3005/stats/tutors
```

Server response

Code | Details

200

Response body

```
{
  "message": "Statistics fetched successfully",
  "stats": [
    {
      "description": "Number of registered tutors",
      "count": 11,
      "type": "STAT"
    },
    {
      "description": "Online tutors",
      "count": 11,
      "type": "STAT_CARD"
    },
    {
      "description": "Home tutors",
      "count": 0,
      "type": "STAT_CARD"
    },
    {
      "description": "Total earnings of all tutors combined",
      "count": 0,
      "type": "STAT"
    },
    {
      "description": "Average earnings per tutor"
```

Response headers

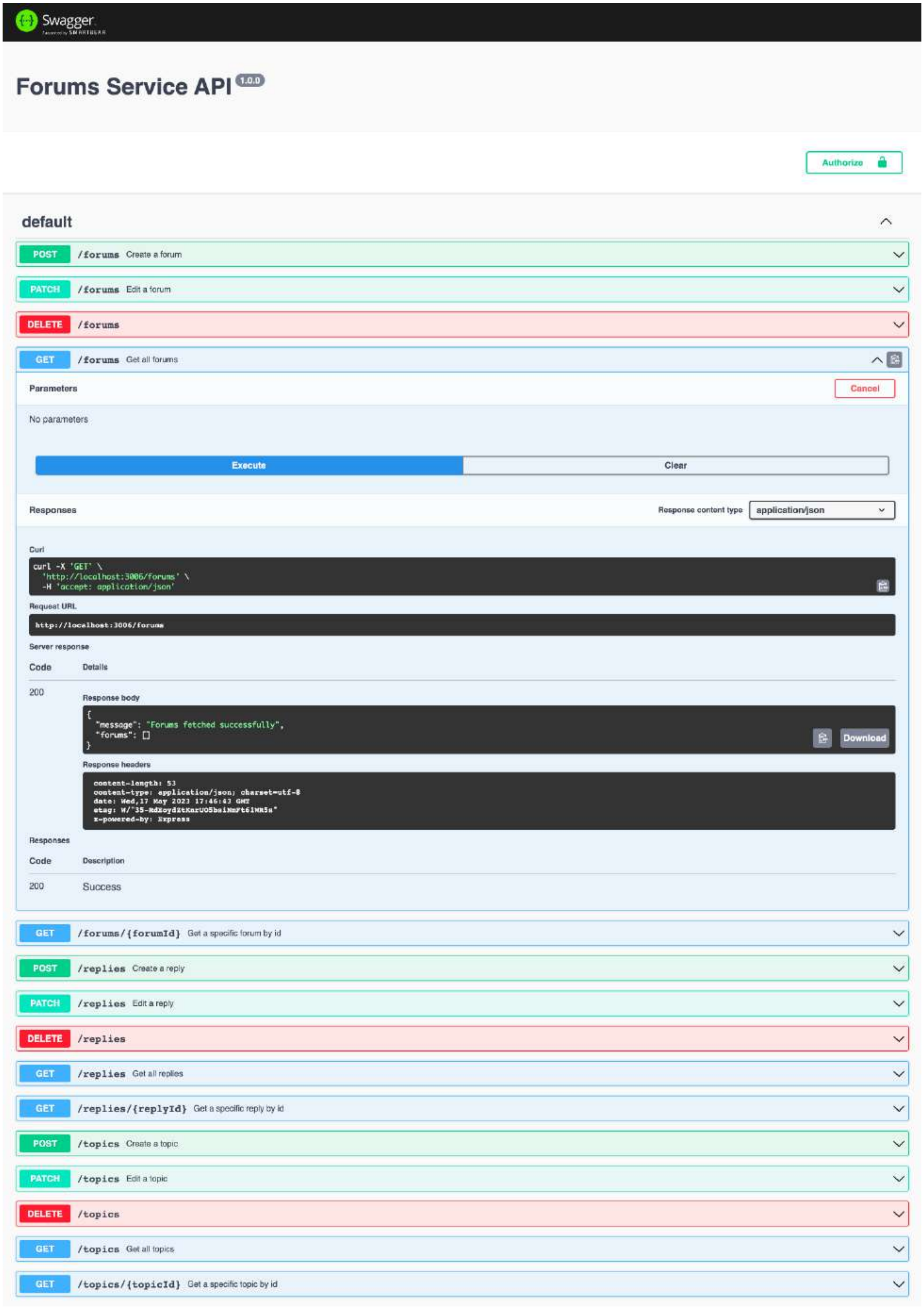
```
connection: keep-alive
content-length: 459
content-type: application/json; charset=utf-8
date: Wed,17 May 2023 17:42:55 GMT
etag: W/"1cb-3/vOu+QBs1Yd2+1EEwsIGP2r48I"
keep-alive: timeout=5
x-powered-by: Express
```

Responses

Code	Description
200	Success

GET /stats/range/{entity} Get range stat for an entity

GET /charts/{chart} Get data for a chart



Forums Service API 1.0.0

default

POST /forums Create a forum

PATCH /forums Edit a forum

DELETE /forums

GET /forums Get all forums

Parameters

No parameters

Responses

Response content type: application/json

Curl

```
curl -X 'GET' \
  'http://localhost:3006/forums' \
  -H 'accept: application/json'
```

Request URL

```
http://localhost:3006/forums
```

Server response

Code	Details
200	Response body

```
{
  "message": "Forums fetched successfully",
  "forums": []
}
```

Response headers

```
 content-length: 53
 content-type: application/json; charset=utf-8
 date: Wed,17 May 2023 17:46:43 GMT
 etag: W/"35-RdXoydZtKarUO5baiNmFt6lWR5s"
 x-powered-by: Express
```

Responses

Code	Description
200	Success

GET /forums/{forumId} Get a specific forum by id

POST /replies Create a reply

PATCH /replies Edit a reply

DELETE /replies

GET /replies Get all replies

GET /replies/{replyId} Get a specific reply by id

POST /topics Create a topic

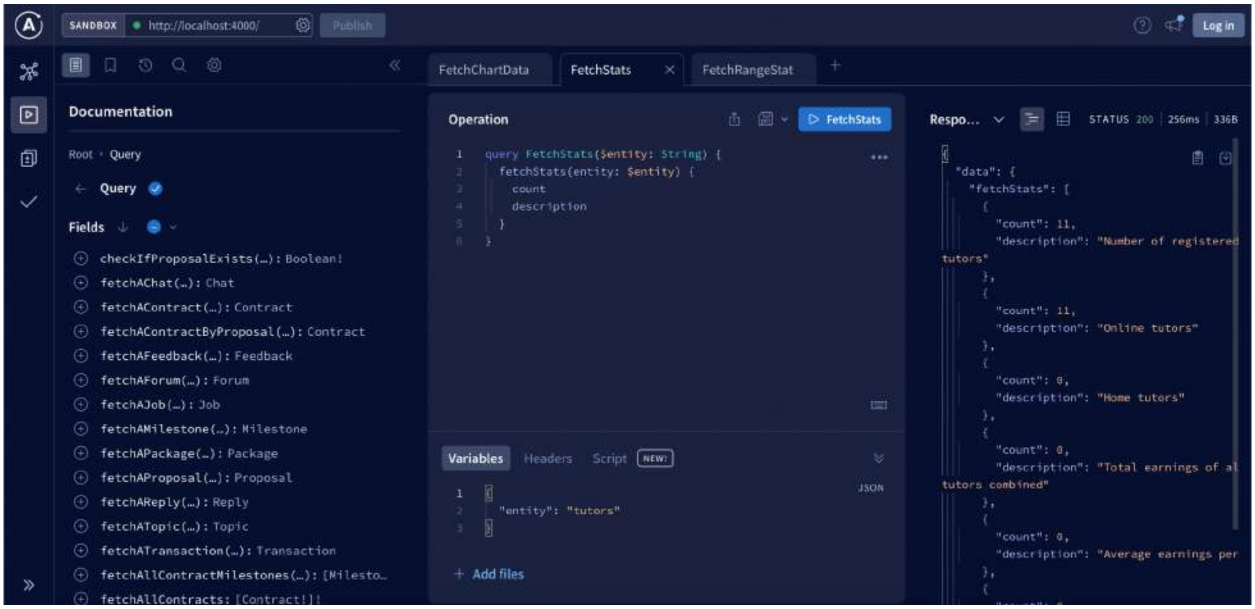
PATCH /topics Edit a topic

DELETE /topics

GET /topics Get all topics

GET /topics/{topicId} Get a specific topic by id

GraphQL Playground was also used to test the GraphQL API interface. GraphQL Playground is an interactive place where GraphQL APIs can be tested. All the schemas, queries and mutations appear there and can be easily tested by specifying the corresponding arguments etc. and query results can be viewed.

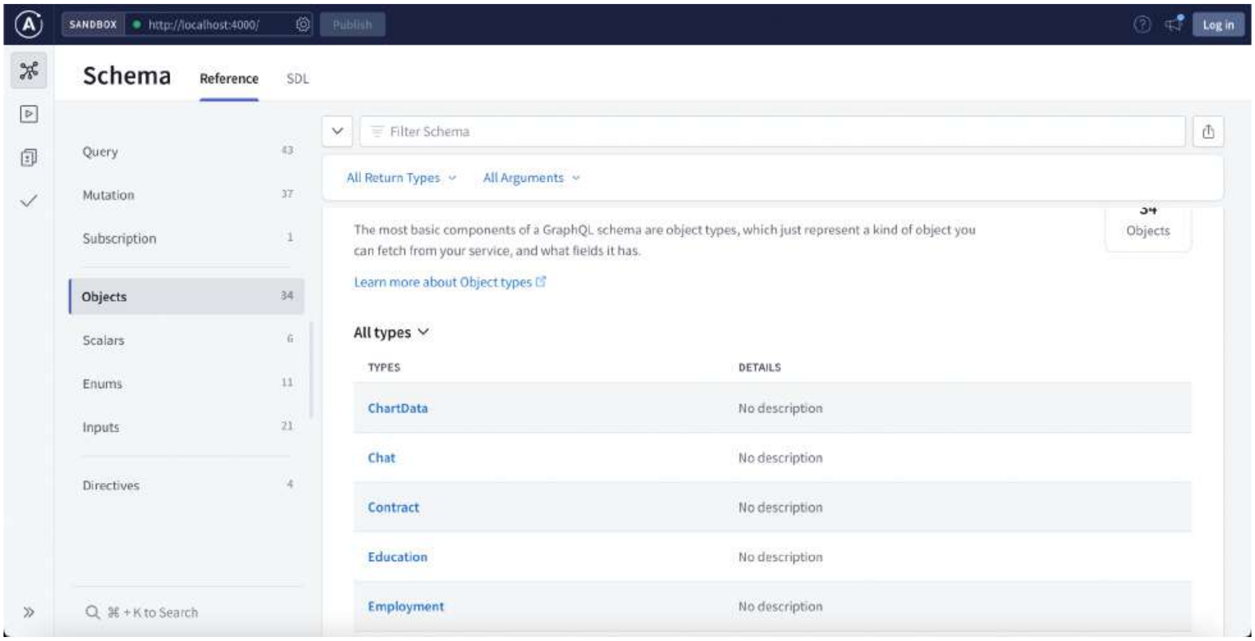


checkIfProposalExists(…): Boolean!
fetchAChat(…): Chat
fetchAContract(…): Contract
fetchAContractByProposal(…): Contract
fetchAFeedback(…): Feedback
fetchAForum(…): Forum
fetchAJob(…): Job
fetchAMilestone(…): Milestone
fetchAPackage(…): Package
fetchAProposal(…): Proposal
fetchAReply(…): Reply
fetchATopic(…): Topic
fetchATransaction(…): Transaction
fetchAllContractMilestones(…): [Milesto…
fetchAllContracts: [Contract!]!

```
query FetchStats($entity: String) {
  fetchStats(entity: $entity) {
    count
    description
  }
}
```

```
{
  "entity": "tutors"
}
```

```
{
  "data": {
    "fetchStats": [
      {
        "count": 11,
        "description": "Number of registered
tutors"
      },
      {
        "count": 11,
        "description": "Online tutors"
      },
      {
        "count": 0,
        "description": "Home tutors"
      },
      {
        "count": 0,
        "description": "Total earnings of al
tutors combined"
      },
      {
        "count": 0,
        "description": "Average earnings per
      },
      {
```



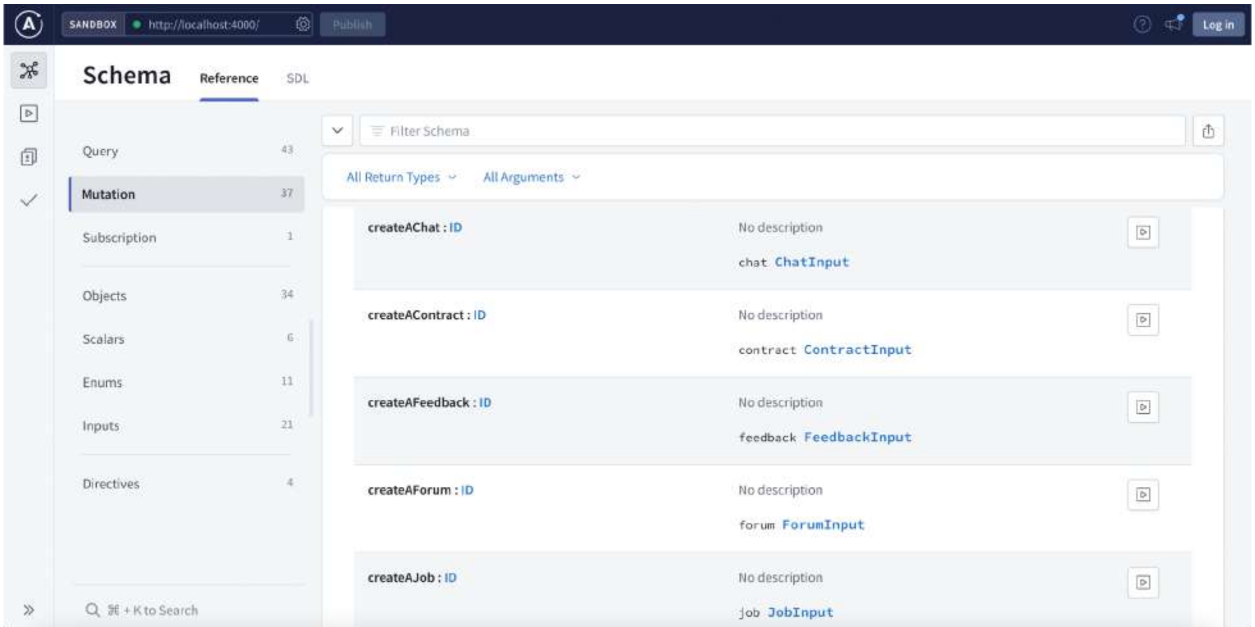
Query 43
Mutation 37
Subscription 1
Objects 34
Scalars 6
Enums 11
Inputs 21
Directives 4

The most basic components of a GraphQL schema are object types, which just represent a kind of object you can fetch from your service, and what fields it has.

Learn more about Object types

All types

TYPES	DETAILS
ChartData	No description
Chat	No description
Contract	No description
Education	No description
Employment	No description



createAChat : ID — No description — chat ChatInput

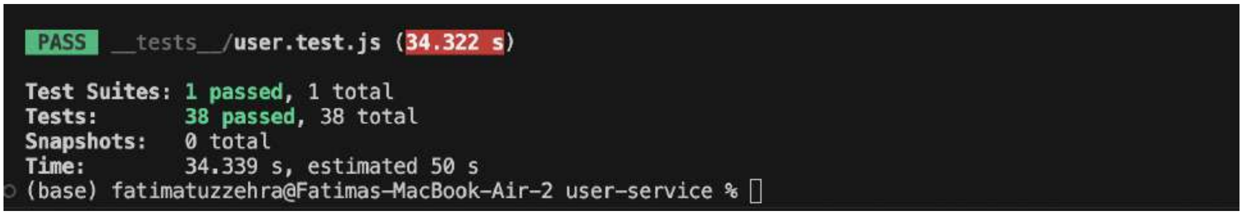
createAContract : ID — No description — contract ContractInput

createAFeedback : ID — No description — feedback FeedbackInput

createAForum : ID — No description — forum ForumInput

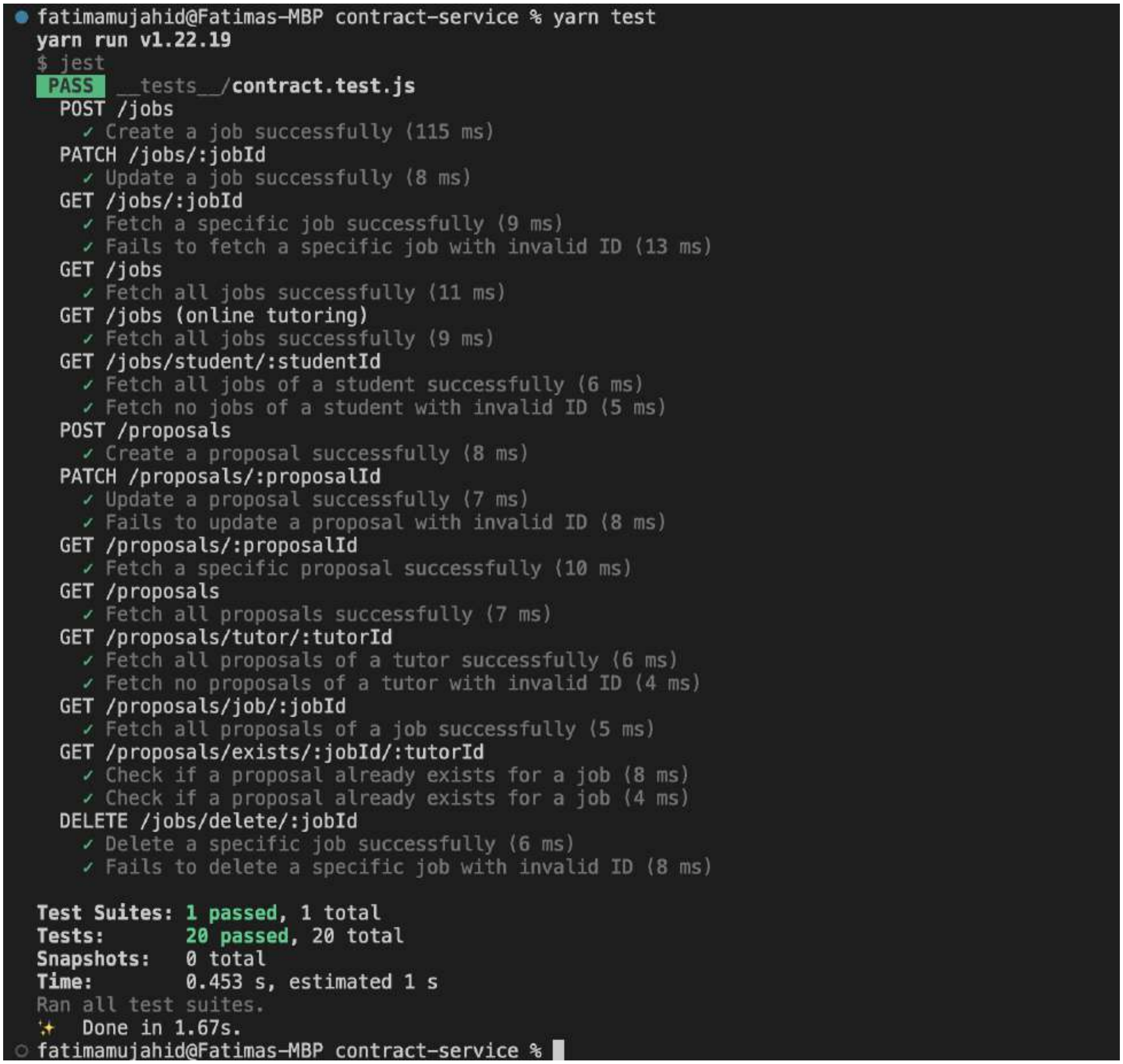
createAJob : ID — No description — job JobInput

Overall, unit testing with Jest, Supertest, Swagger, and GraphQL Playground was very important for making sure that all API routes in Tutorania worked well. The logs from jest tests are mentioned below:



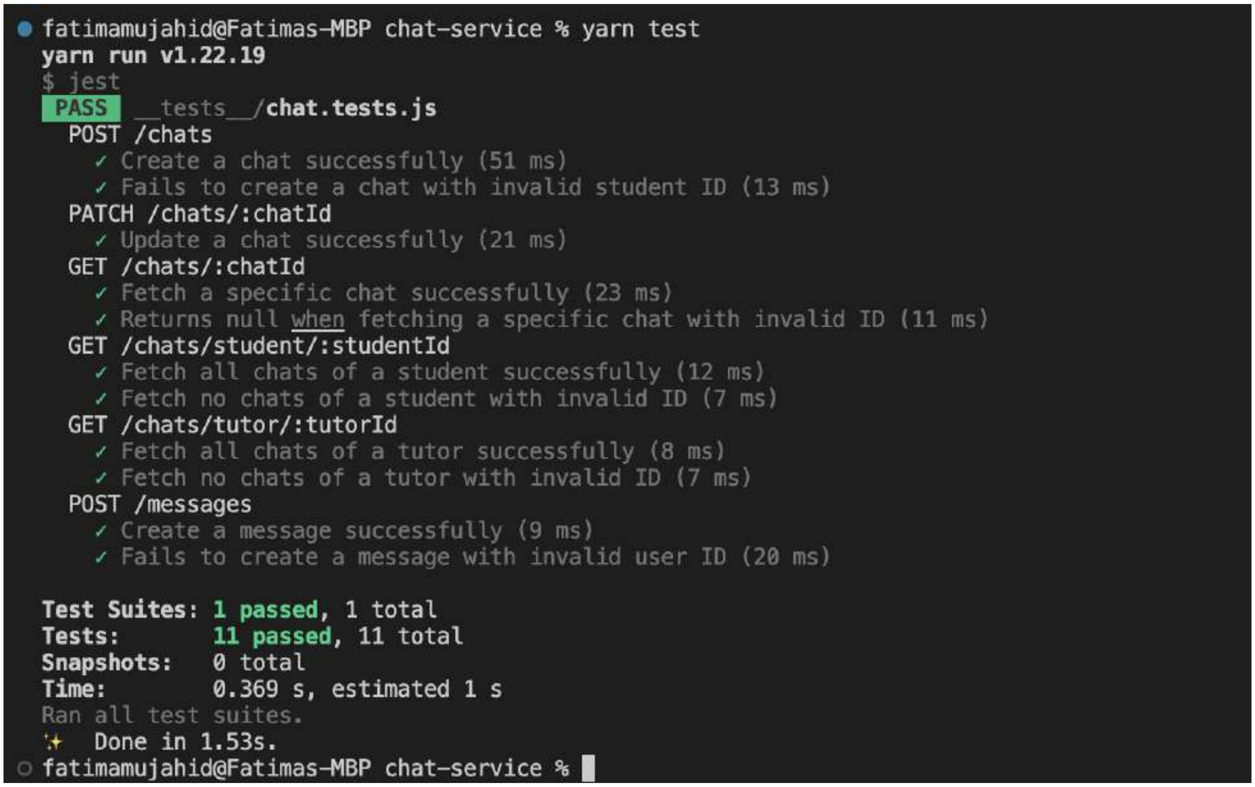
```
PASS  __tests__/user.test.js (34.322 s)

Test Suites: 1 passed, 1 total
Tests:       38 passed, 38 total
Snapshots:   0 total
Time:        34.339 s, estimated 50 s
(base) fatimatuzzehra@Fatimas-MacBook-Air-2 user-service % 
```



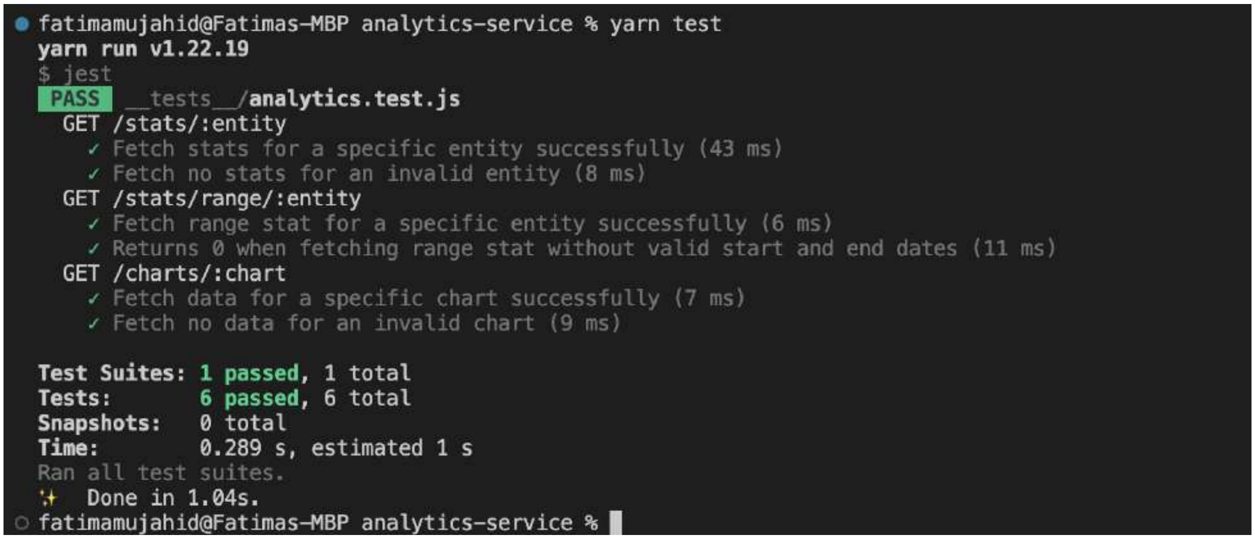
```
● fatimamujahid@Fatimas-MBP contract-service % yarn test
  yarn run v1.22.19
  $ jest
   PASS  __tests__/contract.test.js
    POST /jobs
      ✓ Create a job successfully (115 ms)
    PATCH /jobs/:jobId
      ✓ Update a job successfully (8 ms)
    GET /jobs/:jobId
      ✓ Fetch a specific job successfully (9 ms)
      ✓ Fails to fetch a specific job with invalid ID (13 ms)
    GET /jobs
      ✓ Fetch all jobs successfully (11 ms)
    GET /jobs (online tutoring)
      ✓ Fetch all jobs successfully (9 ms)
    GET /jobs/student/:studentId
      ✓ Fetch all jobs of a student successfully (6 ms)
      ✓ Fetch no jobs of a student with invalid ID (5 ms)
    POST /proposals
      ✓ Create a proposal successfully (8 ms)
    PATCH /proposals/:proposalId
      ✓ Update a proposal successfully (7 ms)
      ✓ Fails to update a proposal with invalid ID (8 ms)
    GET /proposals/:proposalId
      ✓ Fetch a specific proposal successfully (10 ms)
    GET /proposals
      ✓ Fetch all proposals successfully (7 ms)
    GET /proposals/tutor/:tutorId
      ✓ Fetch all proposals of a tutor successfully (6 ms)
      ✓ Fetch no proposals of a tutor with invalid ID (4 ms)
    GET /proposals/job/:jobId
      ✓ Fetch all proposals of a job successfully (5 ms)
    GET /proposals/exists/:jobId/:tutorId
      ✓ Check if a proposal already exists for a job (8 ms)
      ✓ Check if a proposal already exists for a job (4 ms)
    DELETE /jobs/delete/:jobId
      ✓ Delete a specific job successfully (6 ms)
      ✓ Fails to delete a specific job with invalid ID (8 ms)

  Test Suites: 1 passed, 1 total
  Tests:       20 passed, 20 total
  Snapshots:   0 total
  Time:        0.453 s, estimated 1 s
  Ran all test suites.
  ✨  Done in 1.67s.
○ fatimamujahid@Fatimas-MBP contract-service % 
```



```
● fatimamujahid@Fatimas-MBP chat-service % yarn test
  yarn run v1.22.19
  $ jest
  PASS  __tests__/chat.tests.js
    POST /chats
      ✓ Create a chat successfully (51 ms)
      ✓ Fails to create a chat with invalid student ID (13 ms)
    PATCH /chats/:chatId
      ✓ Update a chat successfully (21 ms)
    GET /chats/:chatId
      ✓ Fetch a specific chat successfully (23 ms)
      ✓ Returns null when fetching a specific chat with invalid ID (11 ms)
    GET /chats/student/:studentId
      ✓ Fetch all chats of a student successfully (12 ms)
      ✓ Fetch no chats of a student with invalid ID (7 ms)
    GET /chats/tutor/:tutorId
      ✓ Fetch all chats of a tutor successfully (8 ms)
      ✓ Fetch no chats of a tutor with invalid ID (7 ms)
    POST /messages
      ✓ Create a message successfully (9 ms)
      ✓ Fails to create a message with invalid user ID (20 ms)

  Test Suites: 1 passed, 1 total
  Tests:       11 passed, 11 total
  Snapshots:   0 total
  Time:        0.369 s, estimated 1 s
  Ran all test suites.
  ✨  Done in 1.53s.
○ fatimamujahid@Fatimas-MBP chat-service %
```

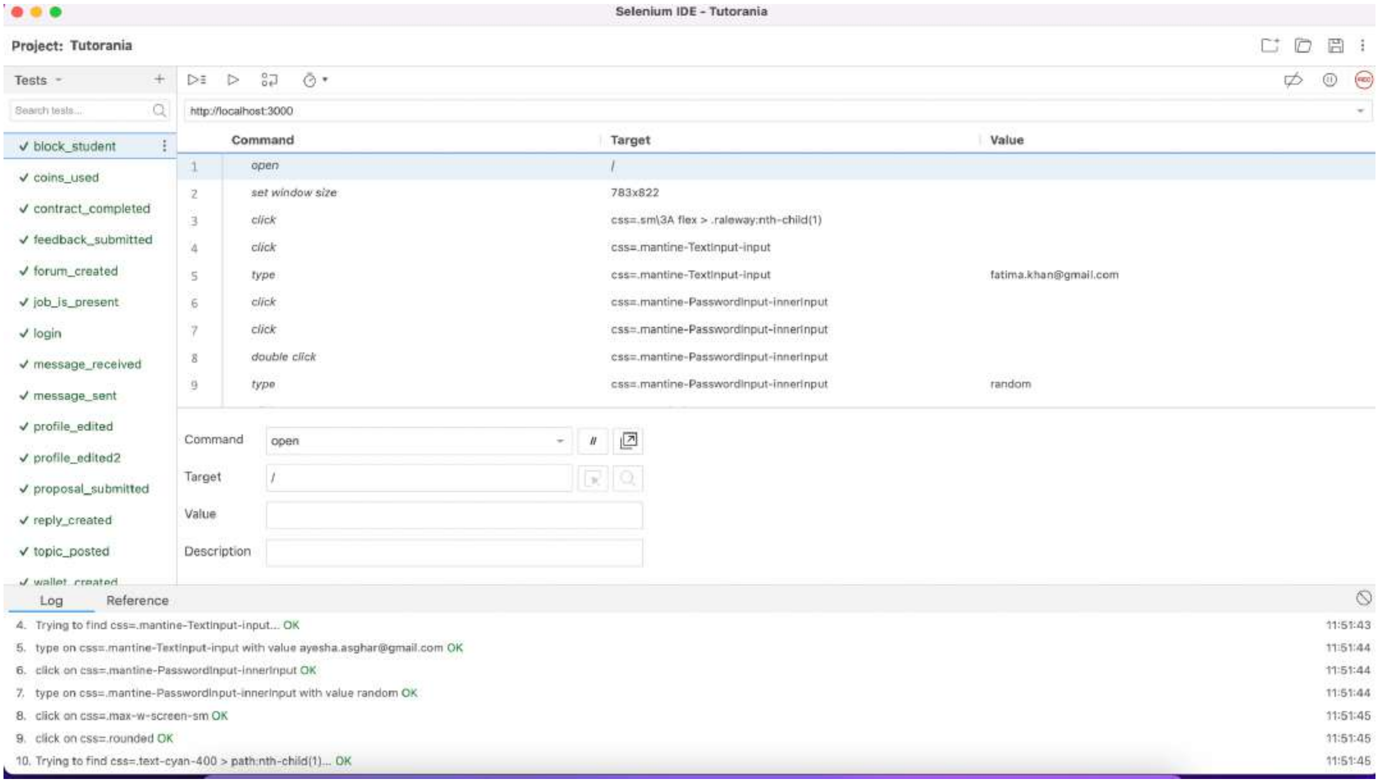


```
● fatimamujahid@Fatimas-MBP analytics-service % yarn test
  yarn run v1.22.19
  $ jest
  PASS  __tests__/analytics.test.js
    GET /stats/:entity
      ✓ Fetch stats for a specific entity successfully (43 ms)
      ✓ Fetch no stats for an invalid entity (8 ms)
    GET /stats/range/:entity
      ✓ Fetch range stat for a specific entity successfully (6 ms)
      ✓ Returns 0 when fetching range stat without valid start and end dates (11 ms)
    GET /charts/:chart
      ✓ Fetch data for a specific chart successfully (7 ms)
      ✓ Fetch no data for an invalid chart (9 ms)

  Test Suites: 1 passed, 1 total
  Tests:       6 passed, 6 total
  Snapshots:   0 total
  Time:        0.289 s, estimated 1 s
  Ran all test suites.
  ✨  Done in 1.04s.
○ fatimamujahid@Fatimas-MBP analytics-service %
```

Integration and System Testing

As the testing tool for Tutorania, Selenium was used to test the system and how it works with other parts. Selenium is a free tool that works with many browsers and computer languages. During the testing process, test cases were made to check how the application worked, including how the menus, forms, links, and buttons worked. Selenium was used to run the test cases, and the results were looked at to find any problems or bugs that needed to be solved. Selenium also allows parallel testing, which lets testers run more than one test at the same time to save time and effort. By using Selenium to do system and integration testing, the Tutorania application was tested thoroughly to make sure it worked as planned. Selenium's testing results helped find any problems or bugs that needed to be fixed before the app is released to the public. The testing method was important for making a successful and high-quality product because it made sure that all the parts of the system worked well together.



http://localhost:3000

	Command	Target	Value
1	open	/	
2	set window size	783x822	
3	click	css=.sm\3A flex > .raleway:nth-child(1)	
4	click	css=.mantine-TextInput-input	
5	type	css=.mantine-TextInput-input	fatima.khan@gmail.com
6	click	css=.mantine-PasswordInput-innerInput	
7	click	css=.mantine-PasswordInput-innerInput	
8	double click	css=.mantine-PasswordInput-innerInput	
9	type	css=.mantine-PasswordInput-innerInput	random

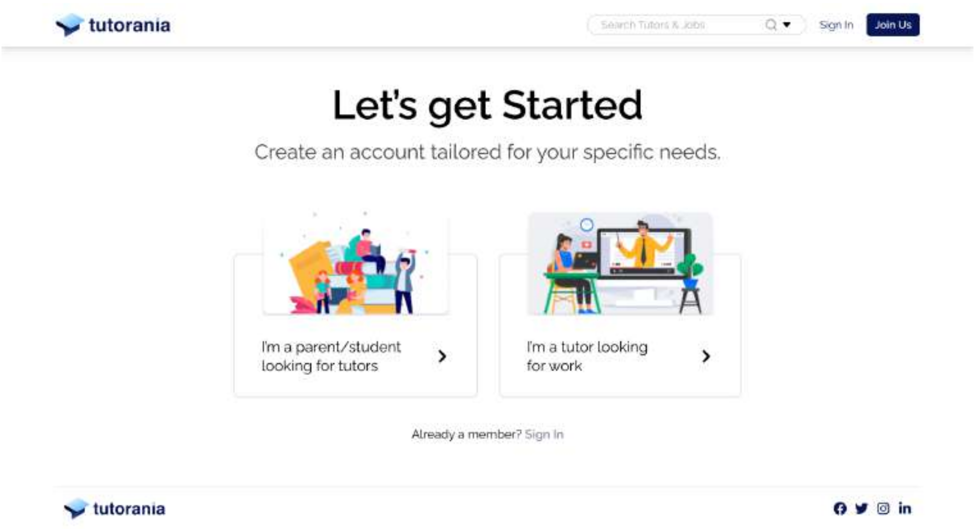
Command: open
Target: /
Value:
Description:

```
4.  Trying to find css=.mantine-TextInput-input... OK                                  11:51:43
5.  type on css=.mantine-TextInput-input with value ayesha.asghar@gmail.com OK         11:51:44
6.  click on css=.mantine-PasswordInput-innerInput OK                                  11:51:44
7.  type on css=.mantine-PasswordInput-innerInput with value random OK                 11:51:44
8.  click on css=.max-w-screen-sm OK                                                   11:51:45
9.  click on css=.rounded OK                                                           11:51:45
10. Trying to find css=.text-cyan-400 > path:nth-child(1)... OK                        11:51:45
```

Non-Functional Testing

Tutorania is a Node.js and Next.js app with a microservices architecture. JMeter was used to test its non-functional and performance features. JMeter is an open-source, pure Java platform software that is used to test how well an application works under a lot of stress and to measure speed and response times. The project team chose to test performance with JMeter. Using JMeter, the performance test scripts were made for the situations that match the identified peak load as well to check how well and quickly the APIs respond with data. On the servers, performance tracking was set up to watch the test and find the bottlenecks while the test was running. JMeter was used to analyse and measure how well Tutorania worked, and it made a report that used statistical information to describe how well the system worked. We made sure the Tutorania app works as intended and meets the requirements for speed, scalability, and stability by using JMeter for non-functional and performance testing.

User Interfaces



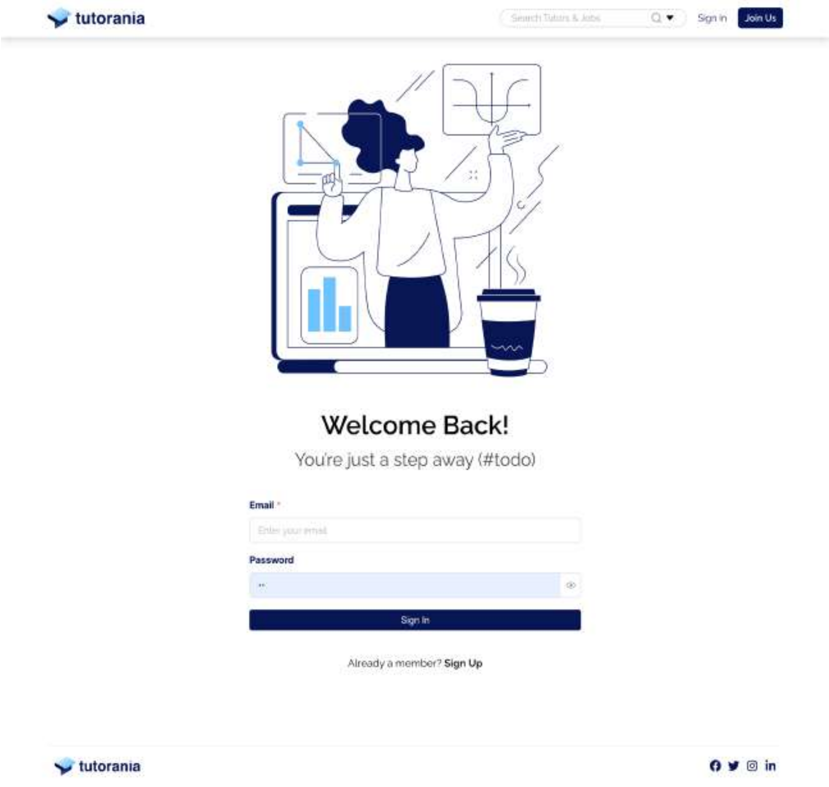
Let's get Started

Create an account tailored for your specific needs.

I'm a parent/student looking for tutors

I'm a tutor looking for work

Already a member? Sign In



Welcome Back!

You're just a step away (#todo)

Email *

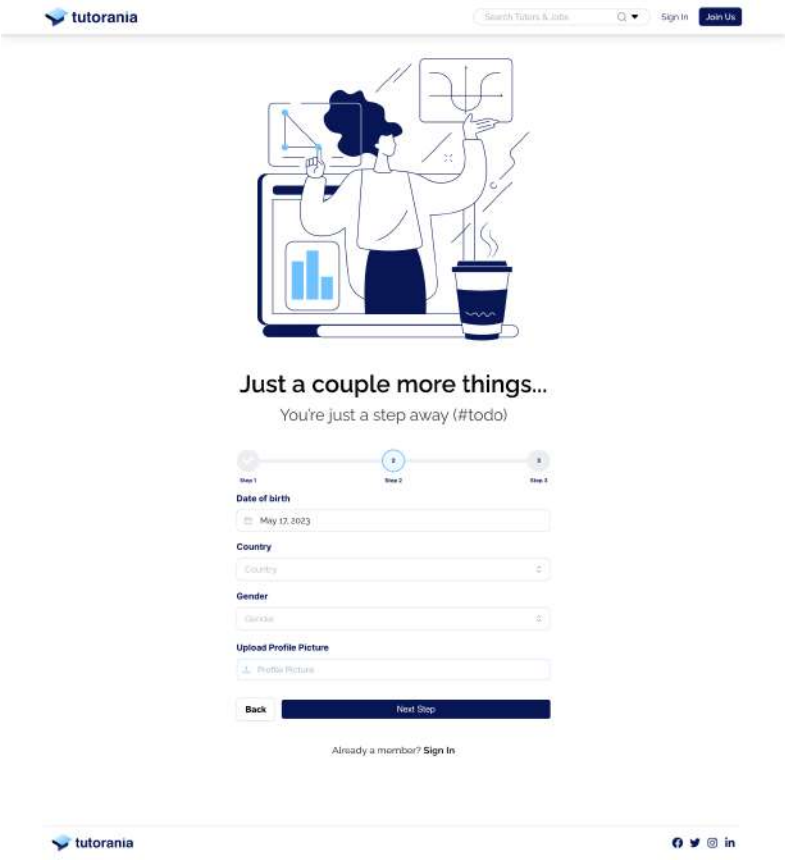
Enter your email

Password

••

Sign In

Already a member? Sign Up



Just a couple more things...

You're just a step away (#todo)

Date of birth

May 17, 2023

Country

Country

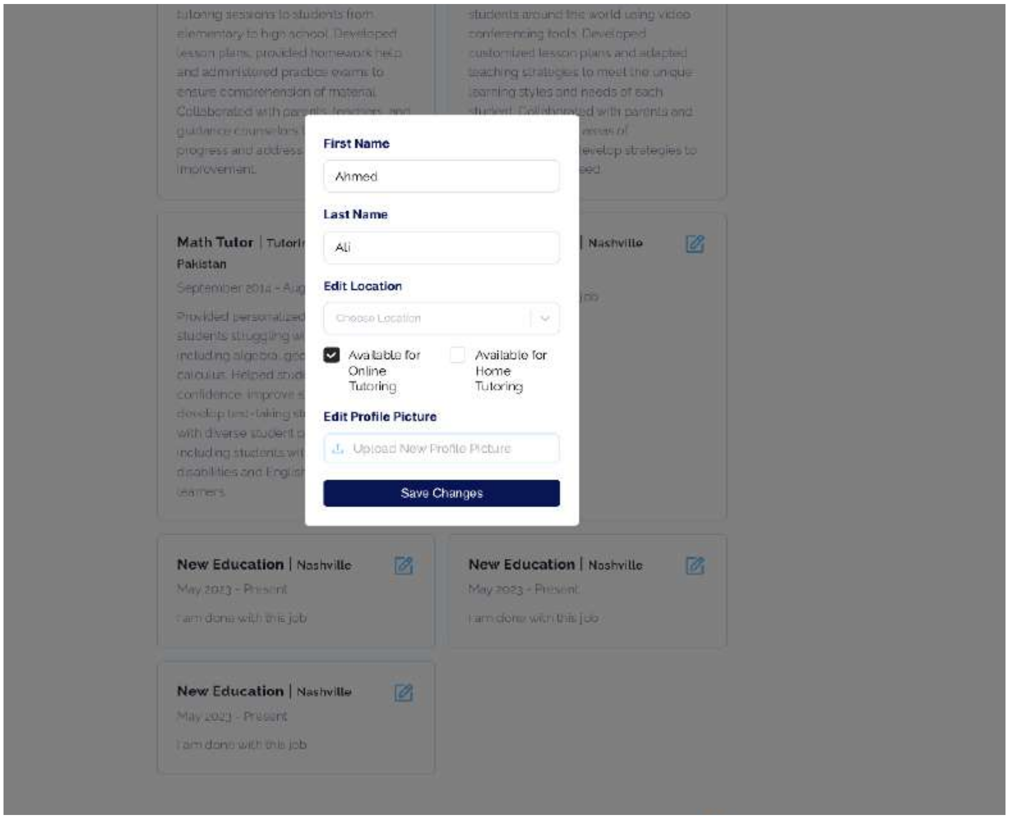
Gender

Gender

Upload Profile Picture

Profile Picture

Already a member? **Sign In**



First Name

Ahmed

Last Name

Ali

Edit Location

Choose Location

☑ Available for Online Tutoring □ Available for Home Tutoring

Edit Profile Picture

Upload New Profile Picture

Save Changes

New Education | Nashville

May 2023 - Present

I am done with this job

New Education | Nashville

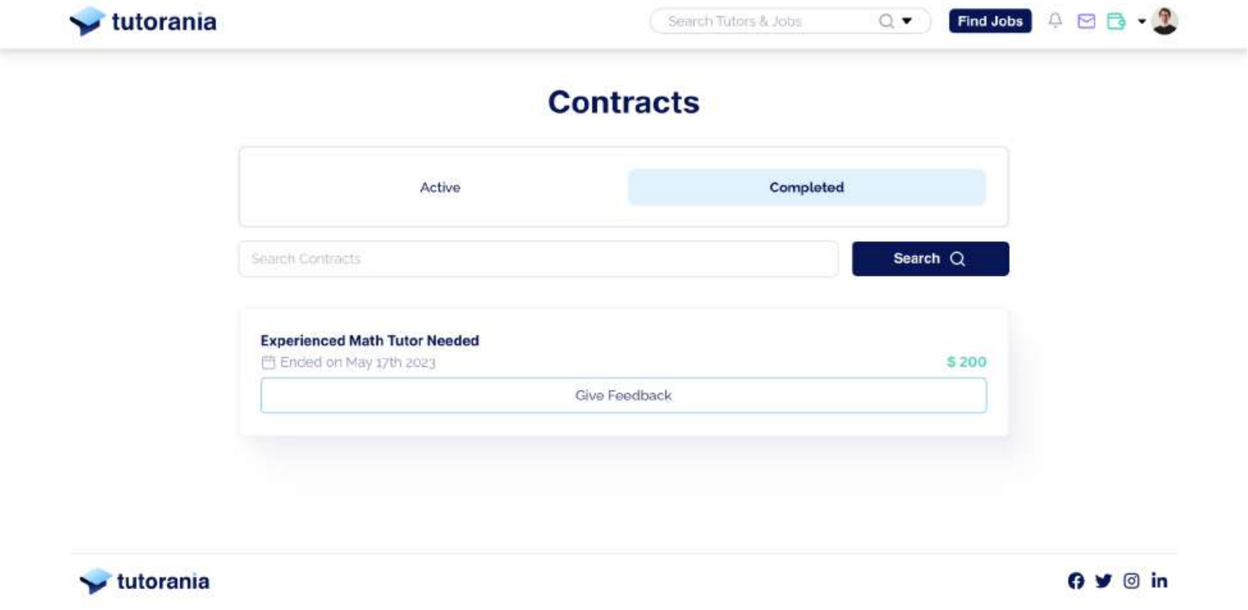
May 2023 - Present

I am done with this job

New Education | Nashville

May 2023 - Present

I am done with this job



Contracts

Active | Completed

Search Contracts

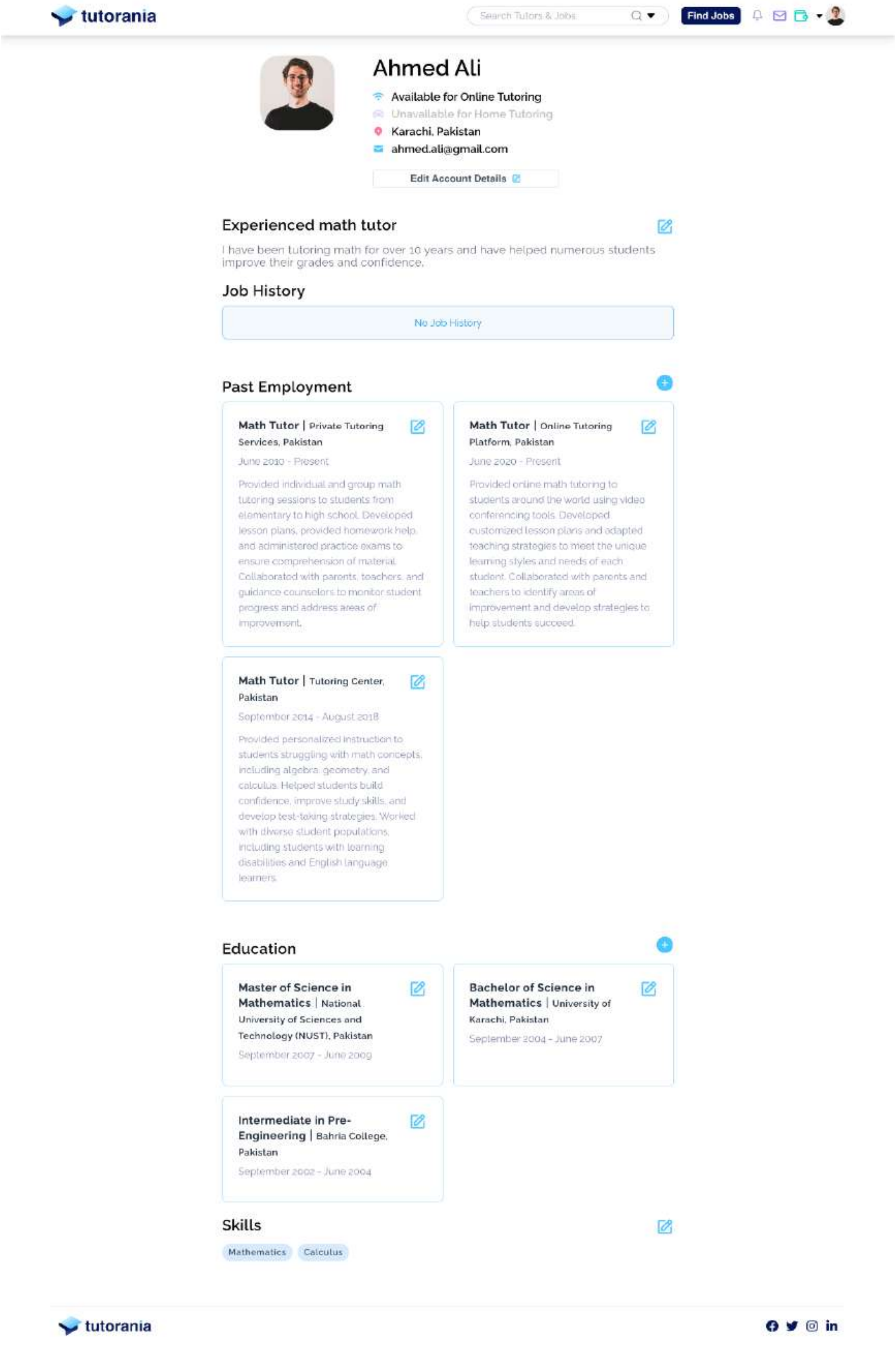
Search

Experienced Math Tutor Needed

Ended on May 17th 2023

$ 200

Give Feedback



Ahmed Ali

Available for Online Tutoring

Unavailable for Home Tutoring

Karachi, Pakistan

ahmed.ali@gmail.com

Experienced math tutor

I have been tutoring math for over 10 years and have helped numerous students improve their grades and confidence.

Job History

No Job History

Past Employment

Math Tutor | Private Tutoring Services, Pakistan

June 2010 - Present

Provided individual and group math tutoring sessions to students from elementary to high school. Developed lesson plans, provided homework help, and administered practice exams to ensure comprehension of material. Collaborated with parents, teachers, and guidance counselors to monitor student progress and address areas of improvement.

Math Tutor | Online Tutoring Platform, Pakistan

June 2020 - Present

Provided online math tutoring to students around the world using video conferencing tools. Developed customized lesson plans and adapted teaching strategies to meet the unique learning styles and needs of each student. Collaborated with parents and teachers to identify areas of improvement and develop strategies to help students succeed.

Math Tutor | Tutoring Center, Pakistan

September 2014 - August 2018

Provided personalized instruction to students struggling with math concepts, including algebra, geometry, and calculus. Helped students build confidence, improve study skills, and develop test-taking strategies. Worked with diverse student populations, including students with learning disabilities and English language learners.

Education

Master of Science in Mathematics | National University of Sciences and Technology (NUST), Pakistan

September 2007 - June 2009

Bachelor of Science in Mathematics | University of Karachi, Pakistan

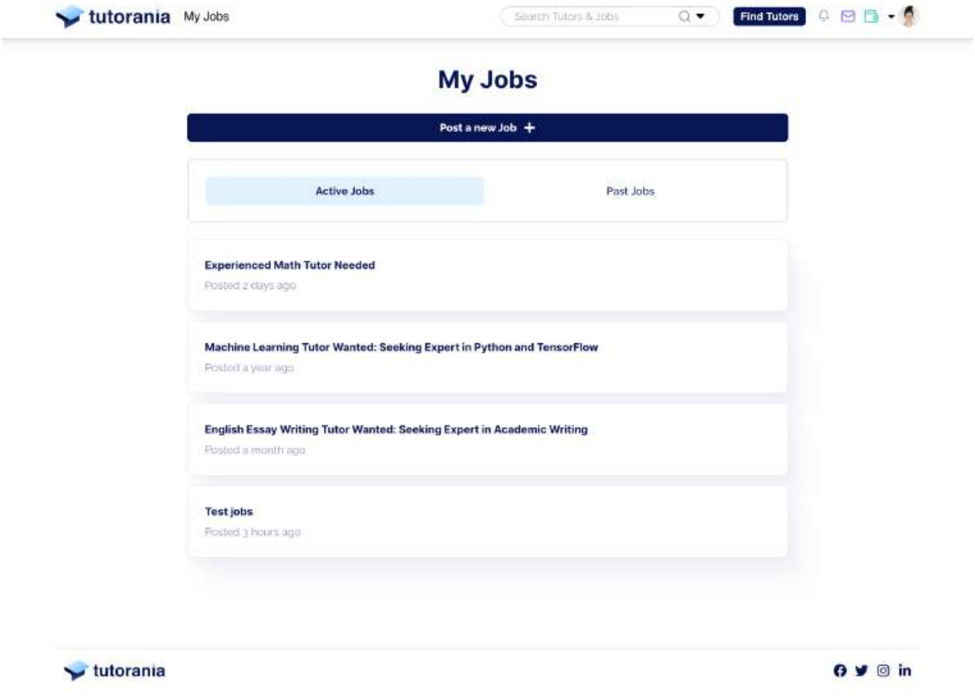
September 2004 - June 2007

Intermediate in Pre-Engineering | Bahria College, Pakistan

September 2002 - June 2004

Skills

Mathematics Calculus



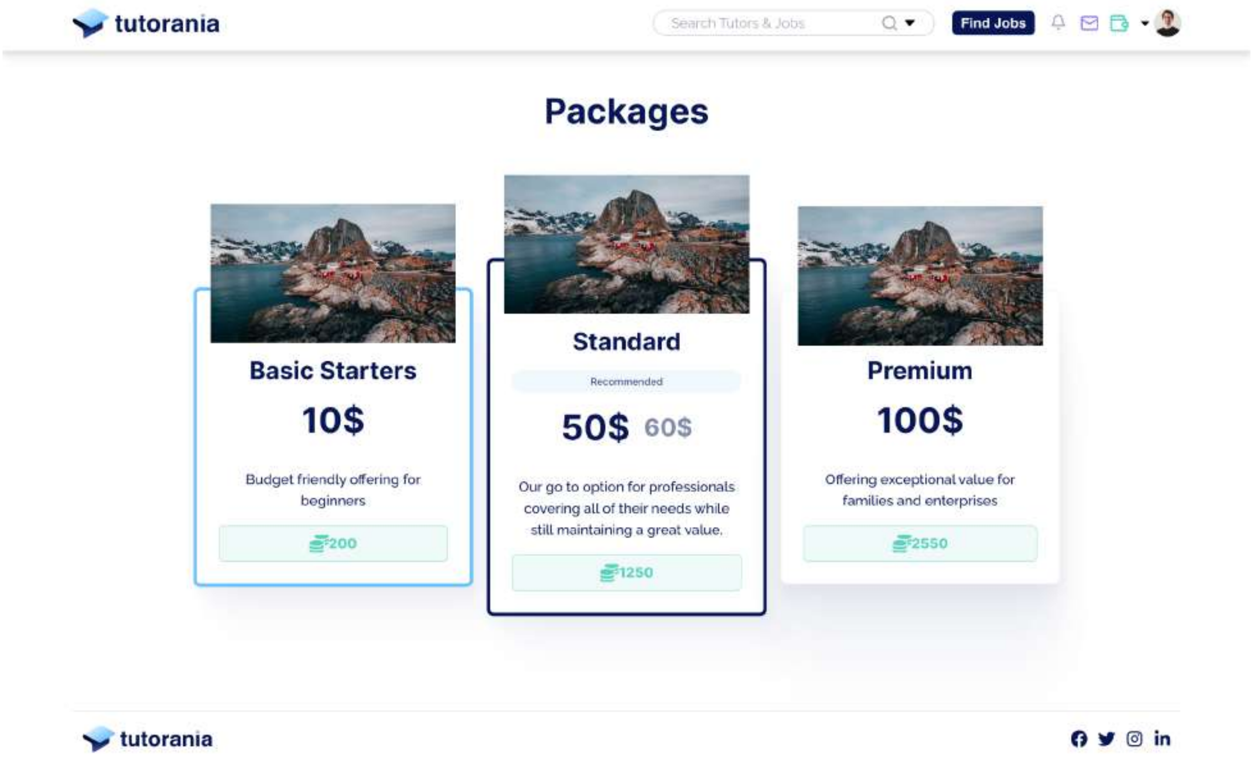
My Jobs

Experienced Math Tutor Needed
Posted 2 days ago

Machine Learning Tutor Wanted: Seeking Expert in Python and TensorFlow
Posted a year ago

English Essay Writing Tutor Wanted: Seeking Expert in Academic Writing
Posted a month ago

Test jobs
Posted 3 hours ago

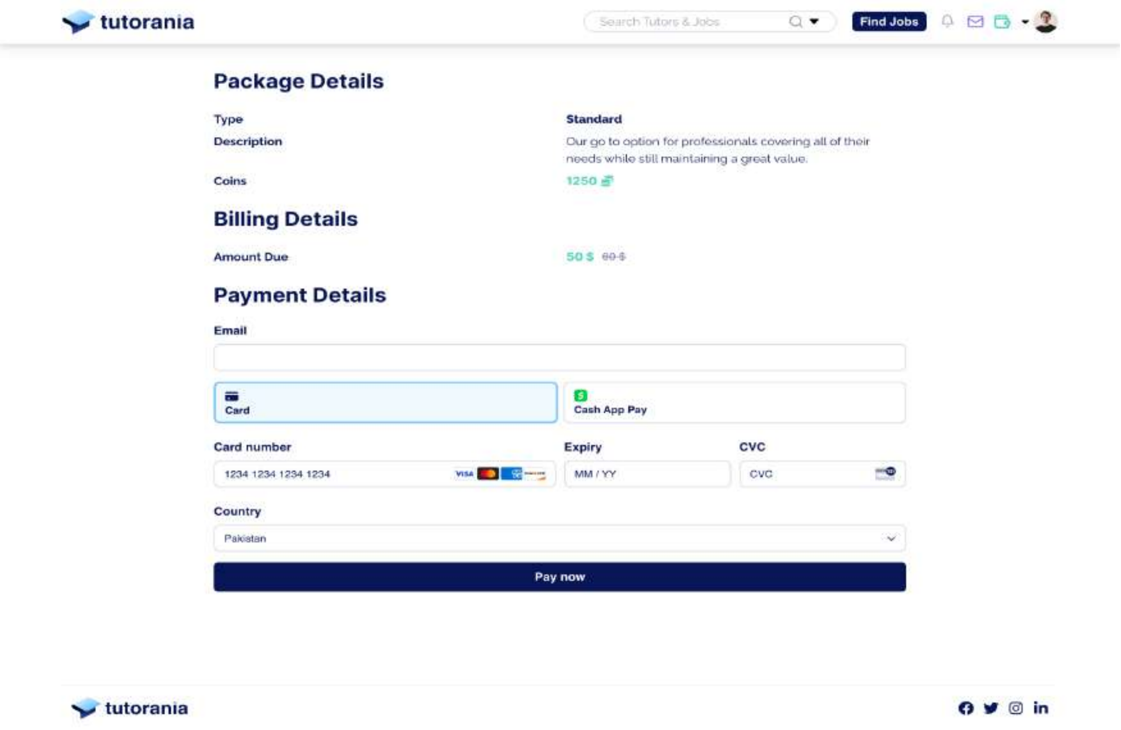


Packages

Basic Starters
10$
Budget friendly offering for beginners
200

Standard
Recommended
50$ 60$
Our go to option for professionals covering all of their needs while still maintaining a great value.
1250

Premium
100$
Offering exceptional value for families and enterprises
2550



Package Details

Type	Standard
Description	Our go to option for professionals covering all of their needs while still maintaining a great value.
Coins	1250

Billing Details

Amount Due	50 $ ~~60 $~~

Payment Details

Email

Card | Cash App Pay

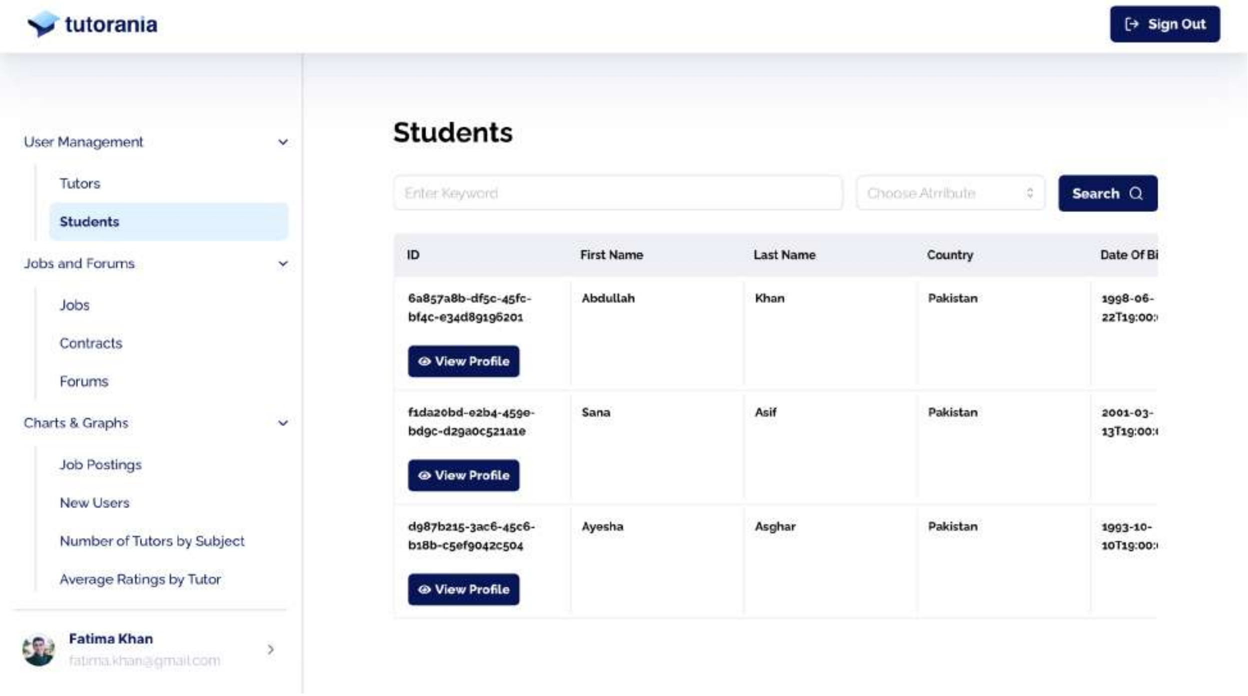
Card number: 1234 1234 1234 1234

Expiry: MM / YY

CVC: CVC

Country: Pakistan

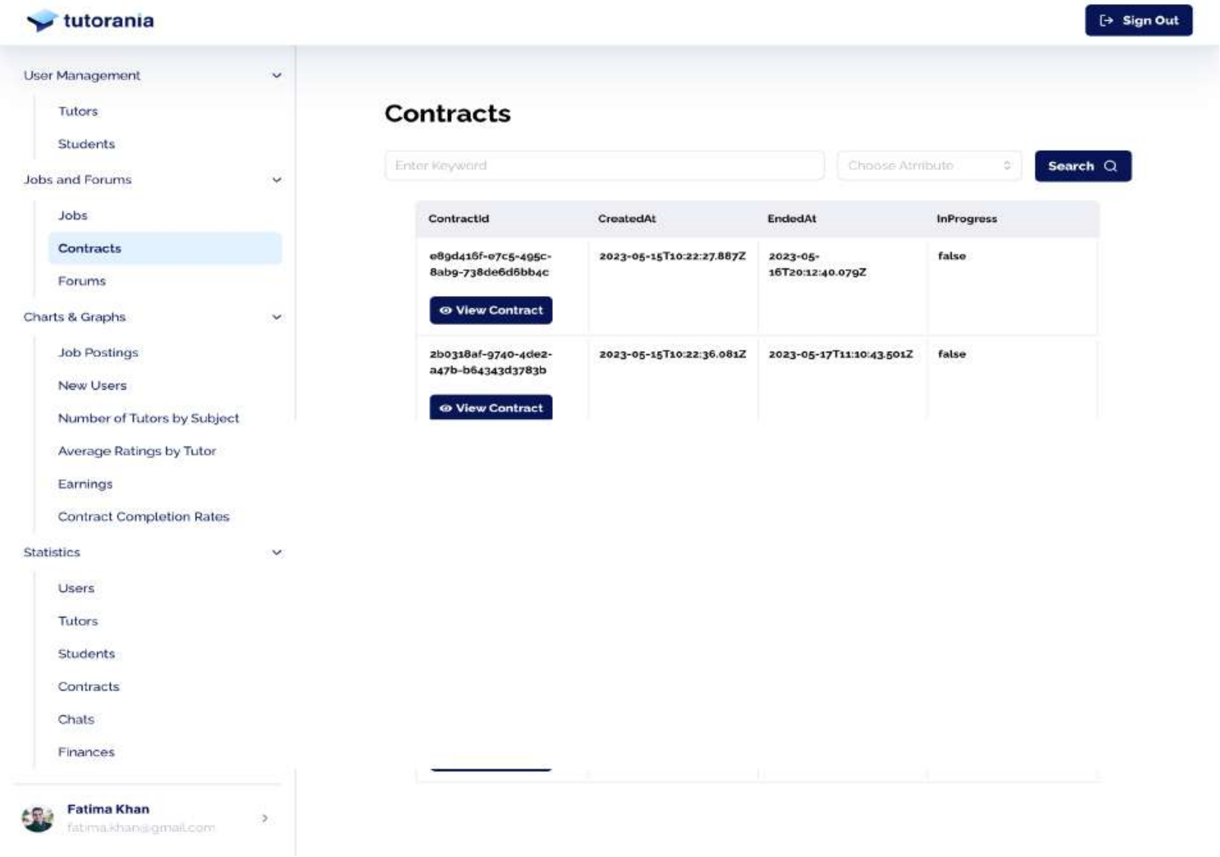
Pay now



Students

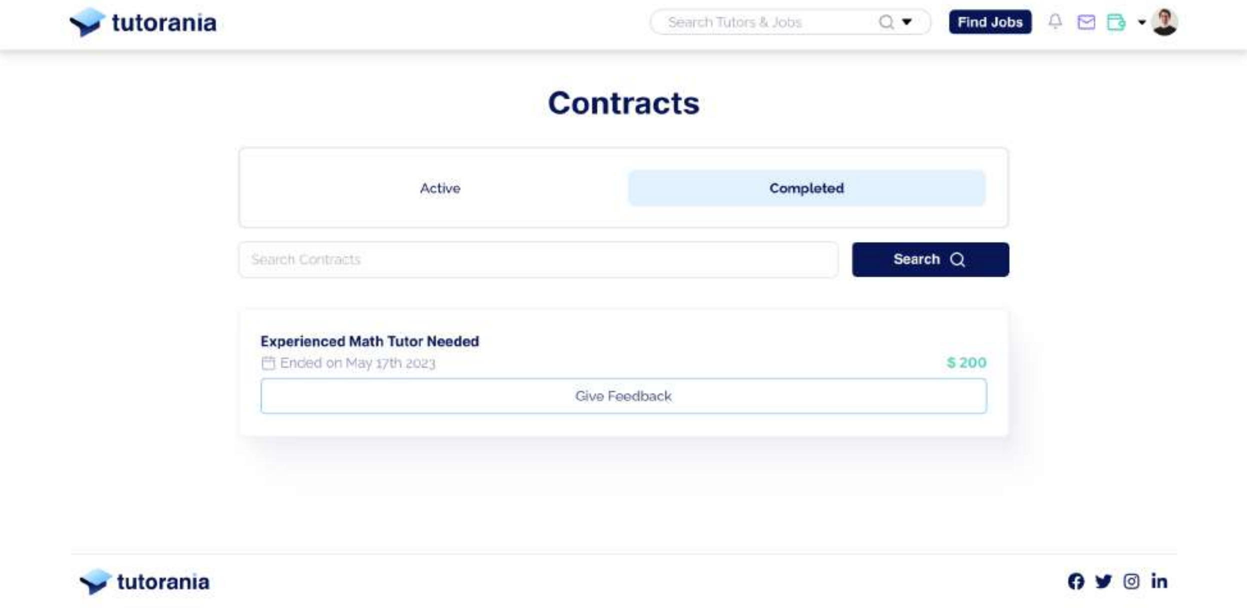
Enter Keyword | Choose Attribute | Search

ID	First Name	Last Name	Country	Date Of Bi
6a857a8b-df5c-45fc-bf4c-e34d89196201 View Profile	Abdullah	Khan	Pakistan	1998-06-22T19:00:
f1da20bd-e2b4-459e-bd9c-d29a0c521a1e View Profile	Sana	Asif	Pakistan	2001-03-13T19:00:
d987b215-3ac6-45c6-b18b-c5ef9042c504 View Profile	Ayesha	Asghar	Pakistan	1993-10-10T19:00:



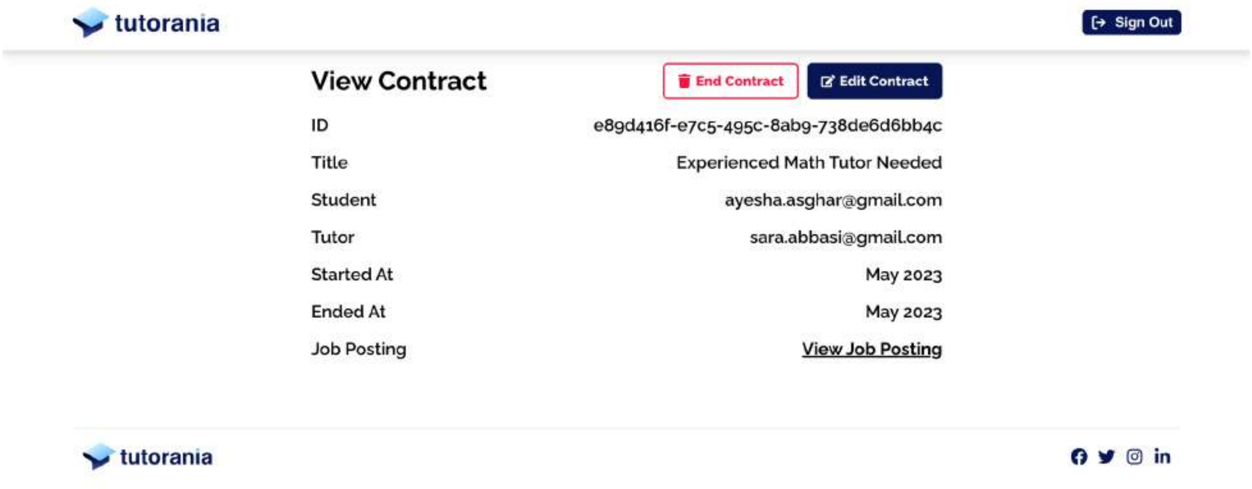
Contracts

ContractId	CreatedAt	EndedAt	InProgress
e89d416f-e7c5-495c-8ab9-738de6d6bb4c View Contract	2023-05-15T10:22:27.887Z	2023-05-16T20:12:40.079Z	false
2b0318af-9740-4de2-a47b-b64343d3783b View Contract	2023-05-15T10:22:36.081Z	2023-05-17T11:10:43.501Z	false



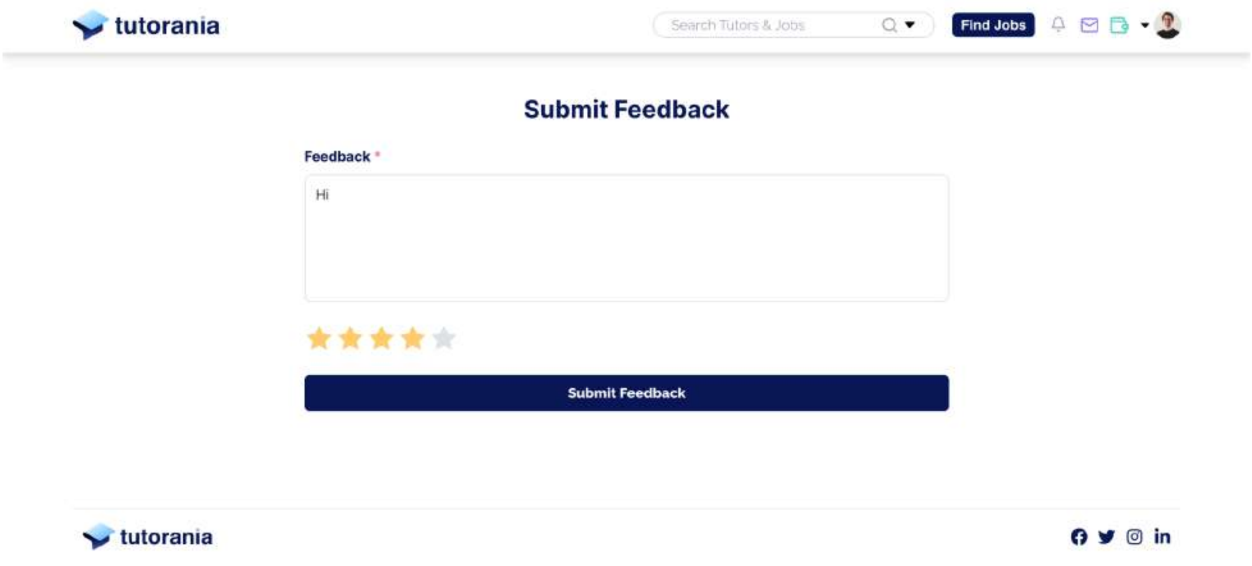
Contracts

Experienced Math Tutor Needed
Ended on May 17th 2023 $ 200



View Contract

ID	e8gd416f-e7c5-495c-8abg-738de6d6bb4c
Title	Experienced Math Tutor Needed
Student	ayesha.asghar@gmail.com
Tutor	sara.abbasi@gmail.com
Started At	May 2023
Ended At	May 2023
Job Posting	View Job Posting

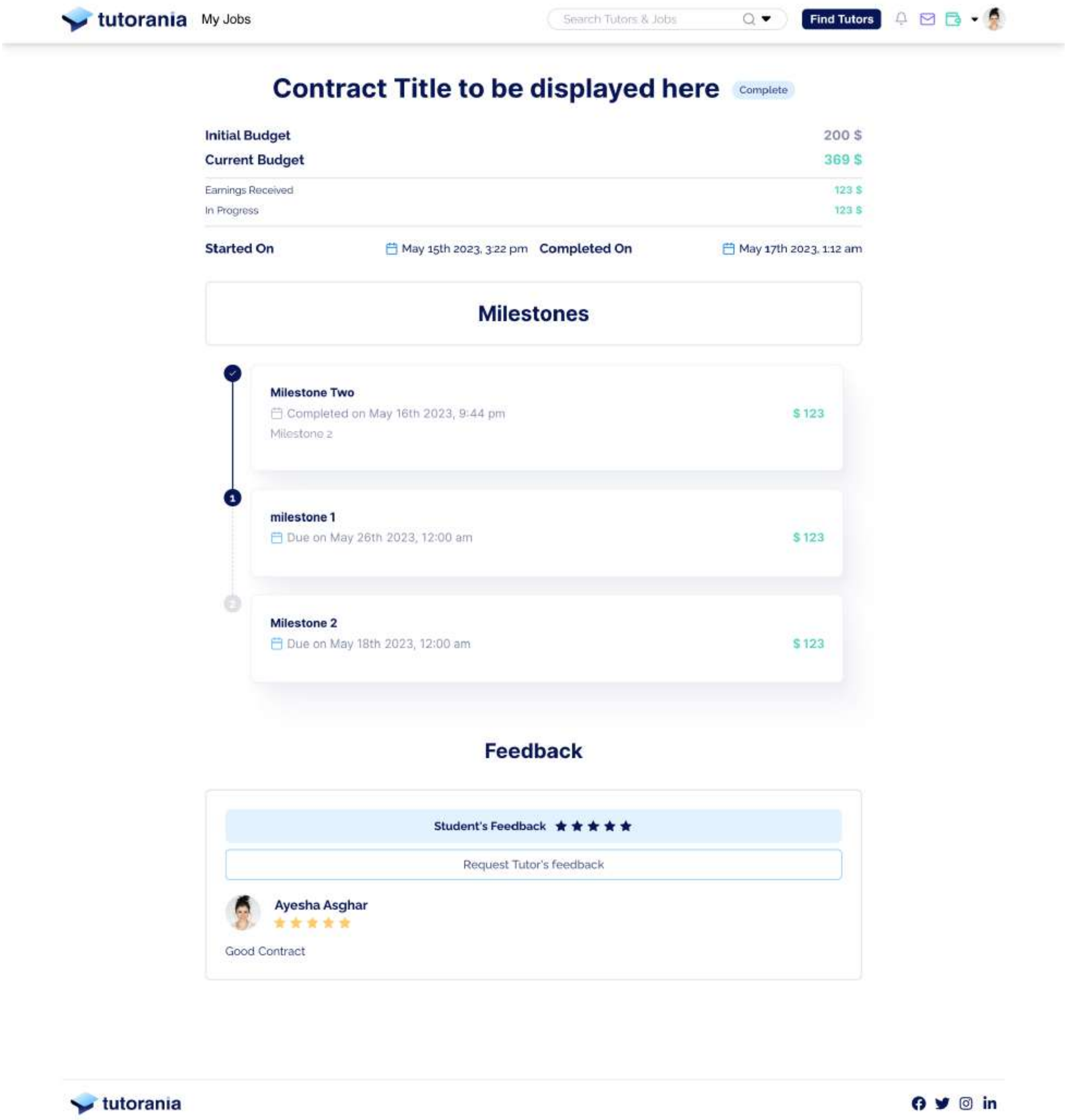


Submit Feedback

Feedback *

Hi

★★★★☆



Contract Title to be displayed here Complete

Initial Budget	200 $
Current Budget	369 $
Earnings Received	123 $
In Progress	123 $
Started On	May 15th 2023, 3:22 pm
Completed On	May 17th 2023, 1:12 am

Milestones

Milestone Two
Completed on May 16th 2023, 9:44 pm
Milestone 2
$ 123

milestone 1
Due on May 26th 2023, 12:00 am
$ 123

Milestone 2
Due on May 18th 2023, 12:00 am
$ 123

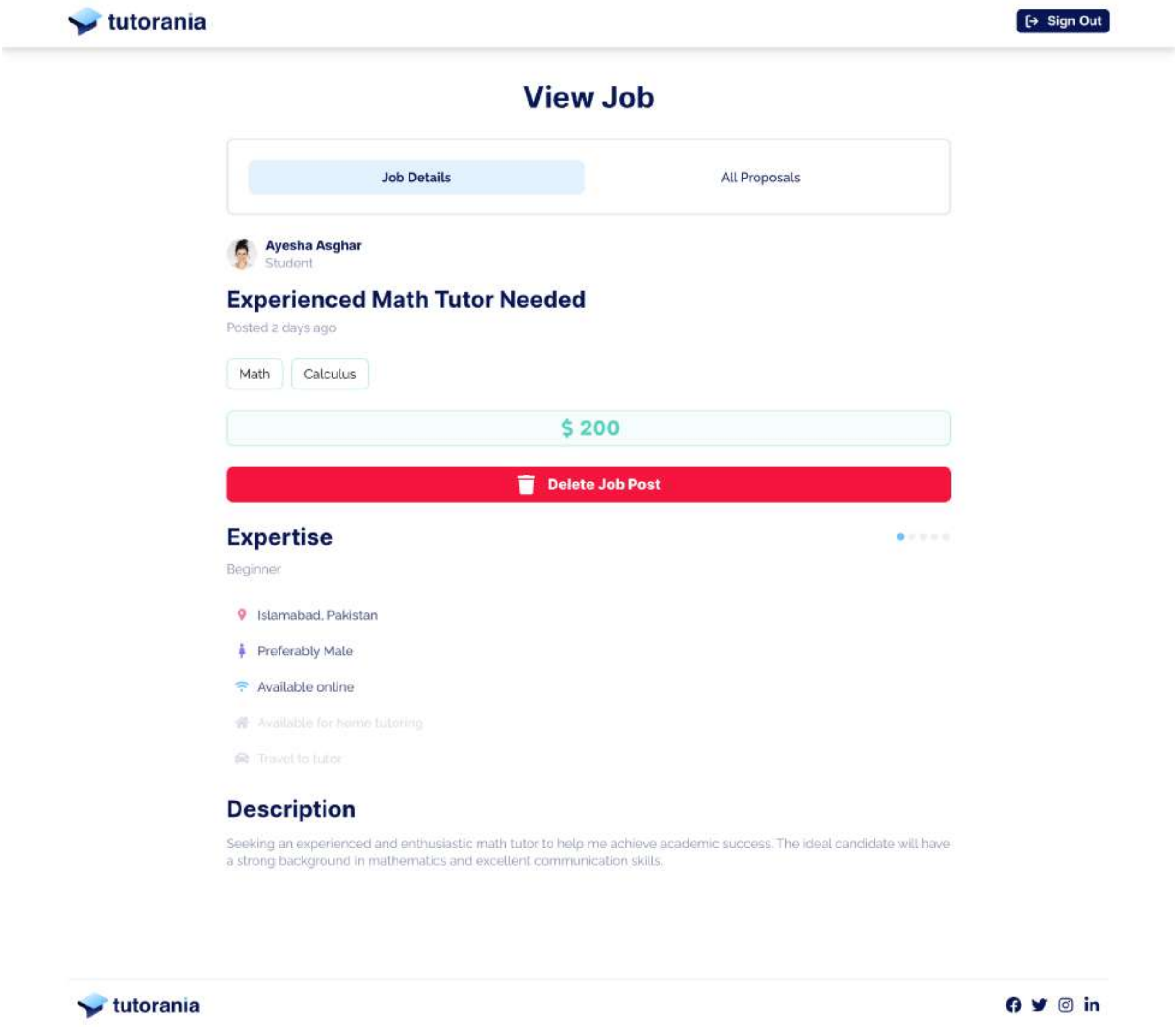
Feedback

Student's Feedback ★★★★★

Request Tutor's feedback

Ayesha Asghar
★★★★★

Good Contract



View Job

Job Details | All Proposals

Ayesha Asghar
Student

Experienced Math Tutor Needed

Posted 2 days ago

Math | Calculus

$ 200

Expertise

Beginner

Islamabad, Pakistan

Preferably Male

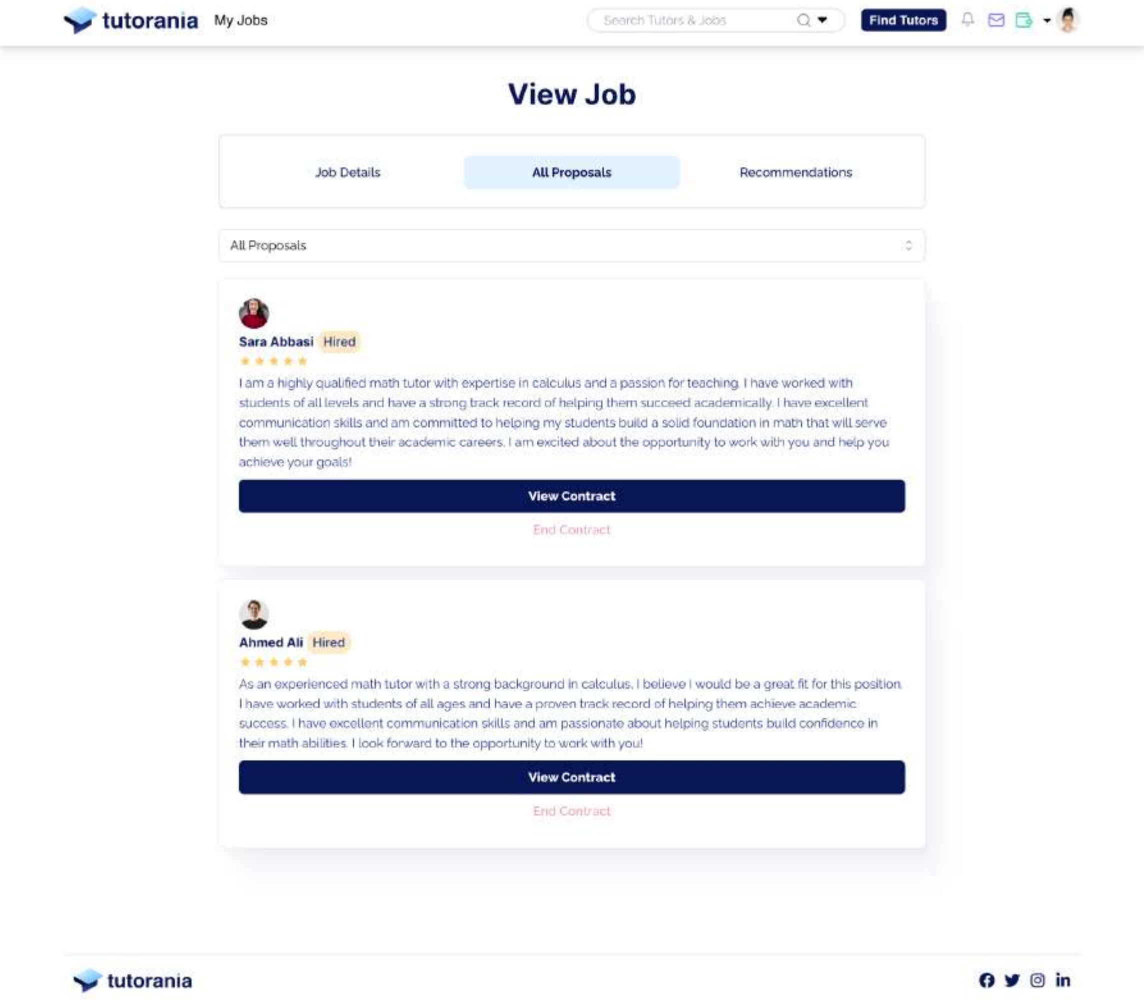
Available online

Available for home tutoring

Travel to tutor

Description

Seeking an experienced and enthusiastic math tutor to help me achieve academic success. The ideal candidate will have a strong background in mathematics and excellent communication skills.



Sara Abbasi Hired

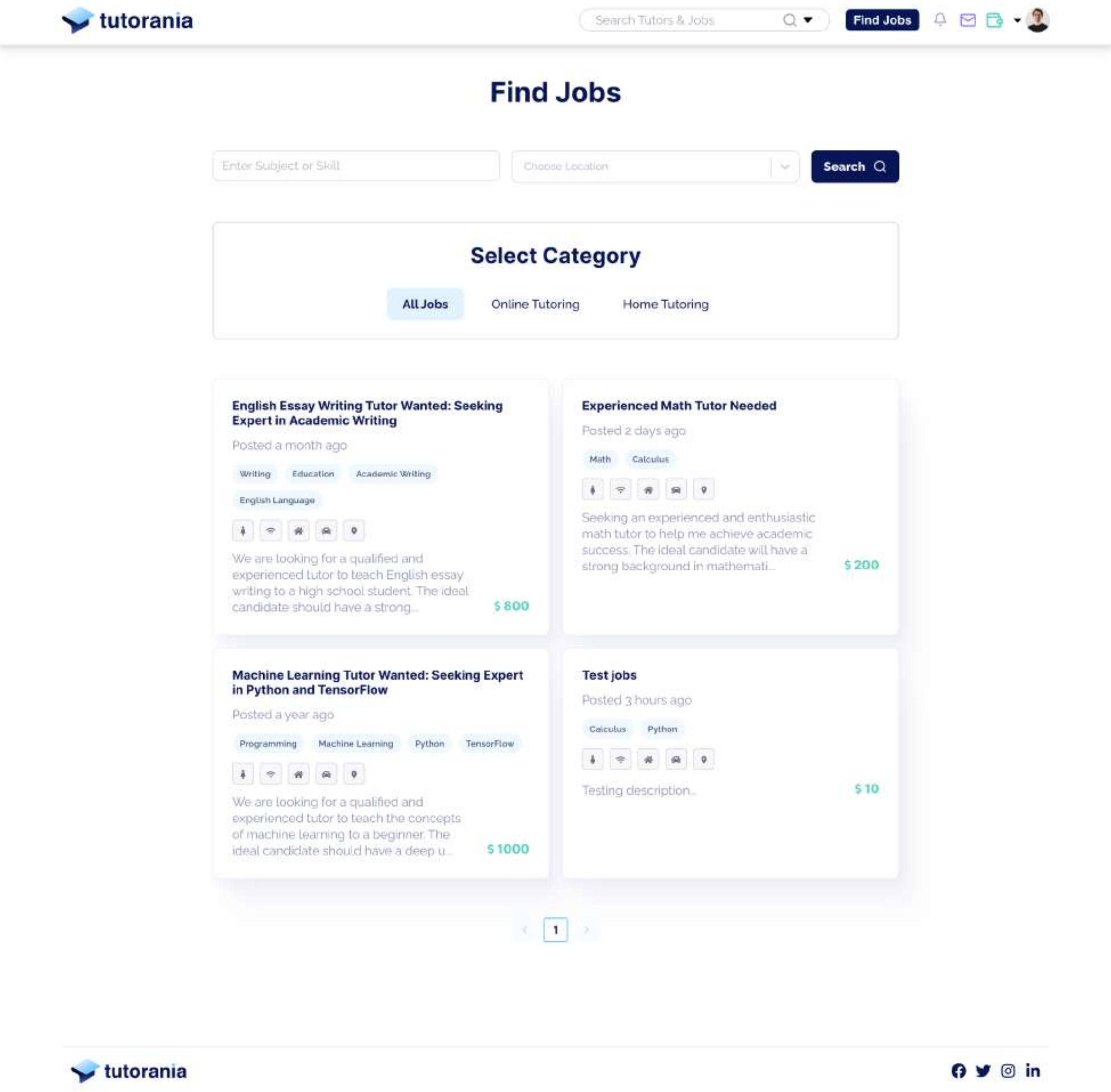
★★★★★

I am a highly qualified math tutor with expertise in calculus and a passion for teaching. I have worked with students of all levels and have a strong track record of helping them succeed academically. I have excellent communication skills and am committed to helping my students build a solid foundation in math that will serve them well throughout their academic careers. I am excited about the opportunity to work with you and help you achieve your goals!

Ahmed Ali Hired

★★★★★

As an experienced math tutor with a strong background in calculus, I believe I would be a great fit for this position. I have worked with students of all ages and have a proven track record of helping them achieve academic success. I have excellent communication skills and am passionate about helping students build confidence in their math abilities. I look forward to the opportunity to work with you!



tutorania

Search Tutors & Jobs

Find Jobs

Find Jobs

Enter Subject or Skill

Choose Location

Search

Select Category

All Jobs | Online Tutoring | Home Tutoring

English Essay Writing Tutor Wanted: Seeking Expert in Academic Writing

Posted a month ago

Writing · Education · Academic Writing · English Language

We are looking for a qualified and experienced tutor to teach English essay writing to a high school student. The ideal candidate should have a strong...

$ 800

Experienced Math Tutor Needed

Posted 2 days ago

Math · Calculus

Seeking an experienced and enthusiastic math tutor to help me achieve academic success. The ideal candidate will have a strong background in mathemati...

$ 200

Machine Learning Tutor Wanted: Seeking Expert in Python and TensorFlow

Posted a year ago

Programming · Machine Learning · Python · TensorFlow

We are looking for a qualified and experienced tutor to teach the concepts of machine learning to a beginner. The ideal candidate should have a deep u...

$ 1000

Test jobs

Posted 3 hours ago

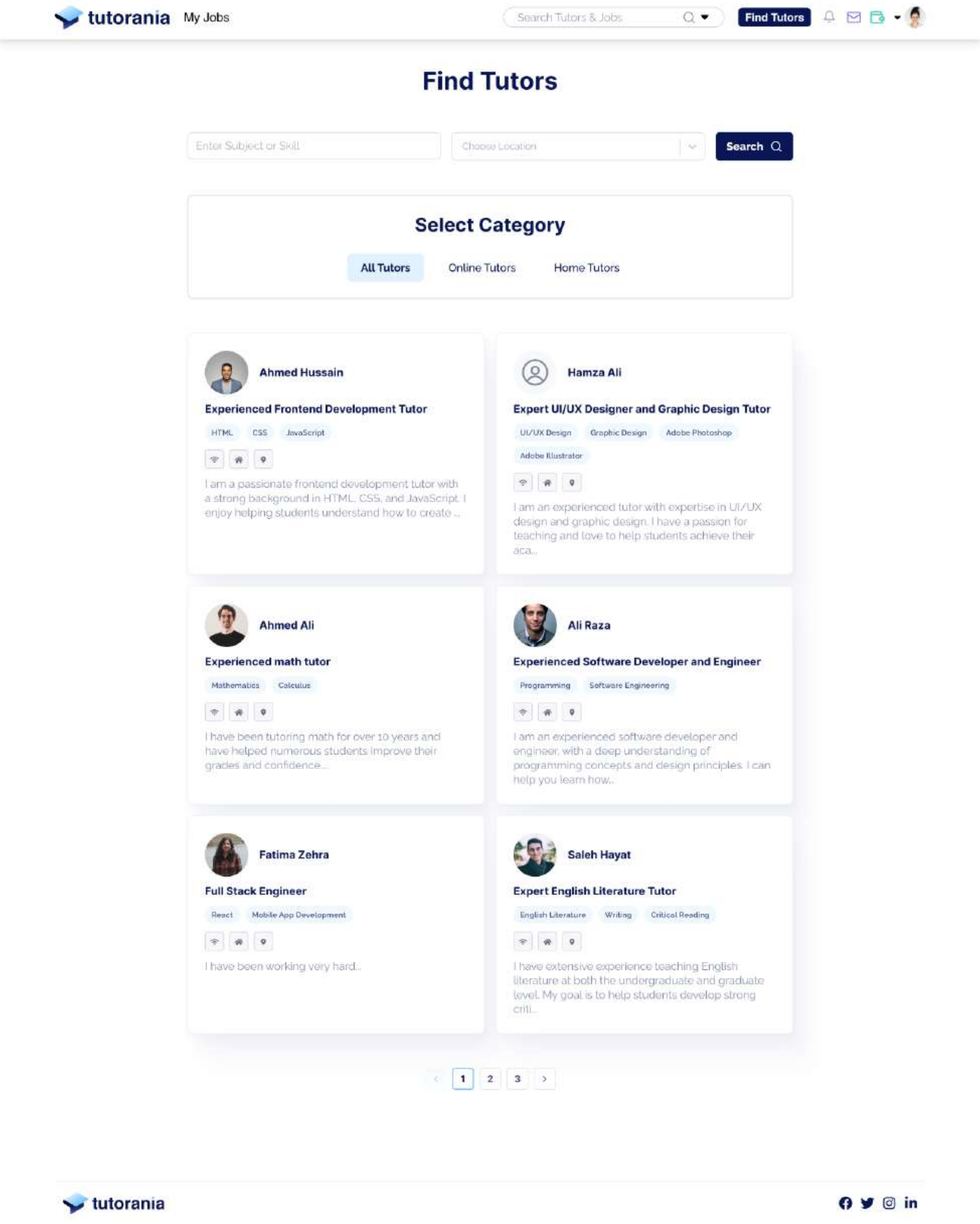
Calculus · Python

Testing description...

$ 10

1

tutorania



Find Tutors

Select Category

All Tutors · Online Tutors · Home Tutors

Ahmed Hussain

Experienced Frontend Development Tutor

HTML · CSS · JavaScript

I am a passionate frontend development tutor with a strong background in HTML, CSS, and JavaScript. I enjoy helping students understand how to create ...

Hamza Ali

Expert UI/UX Designer and Graphic Design Tutor

UI/UX Design · Graphic Design · Adobe Photoshop · Adobe Illustrator

I am an experienced tutor with expertise in UI/UX design and graphic design. I have a passion for teaching and love to help students achieve their aca...

Ahmed Ali

Experienced math tutor

Mathematics · Calculus

I have been tutoring math for over 10 years and have helped numerous students improve their grades and confidence...

Ali Raza

Experienced Software Developer and Engineer

Programming · Software Engineering

I am an experienced software developer and engineer, with a deep understanding of programming concepts and design principles. I can help you learn how...

Fatima Zehra

Full Stack Engineer

React · Mobile App Development

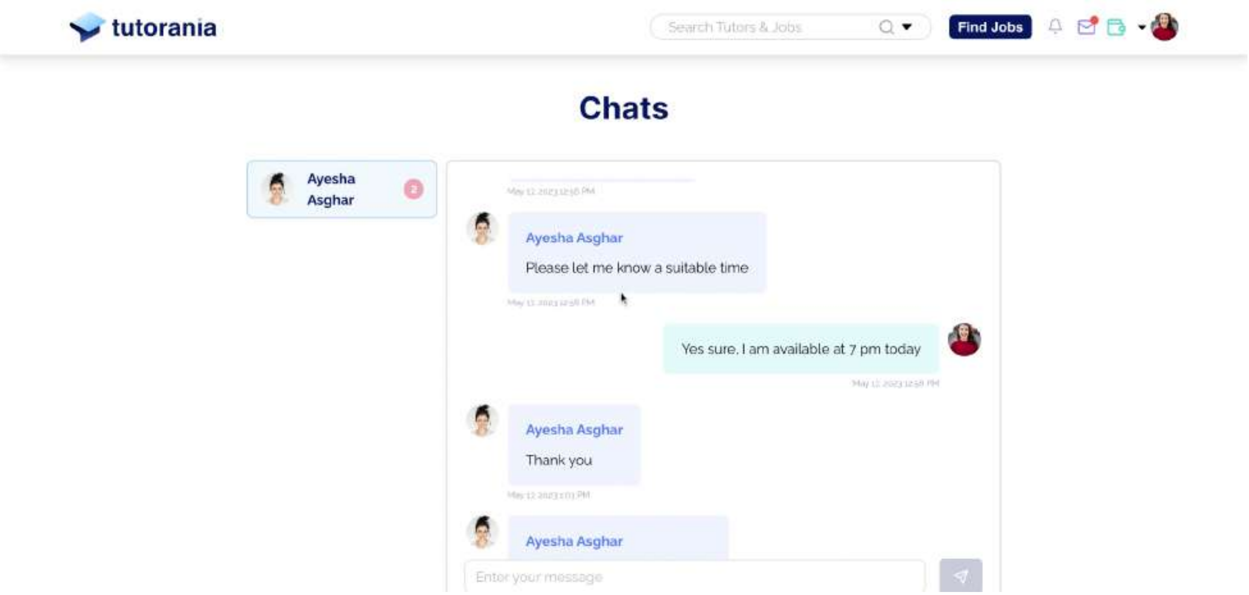
I have been working very hard...

Saleh Hayat

Expert English Literature Tutor

English Literature · Writing · Critical Reading

I have extensive experience teaching English literature at both the undergraduate and graduate level. My goal is to help students develop strong criti...



Chats

Ayesha Asghar 2

May 12 2023 12:56 PM

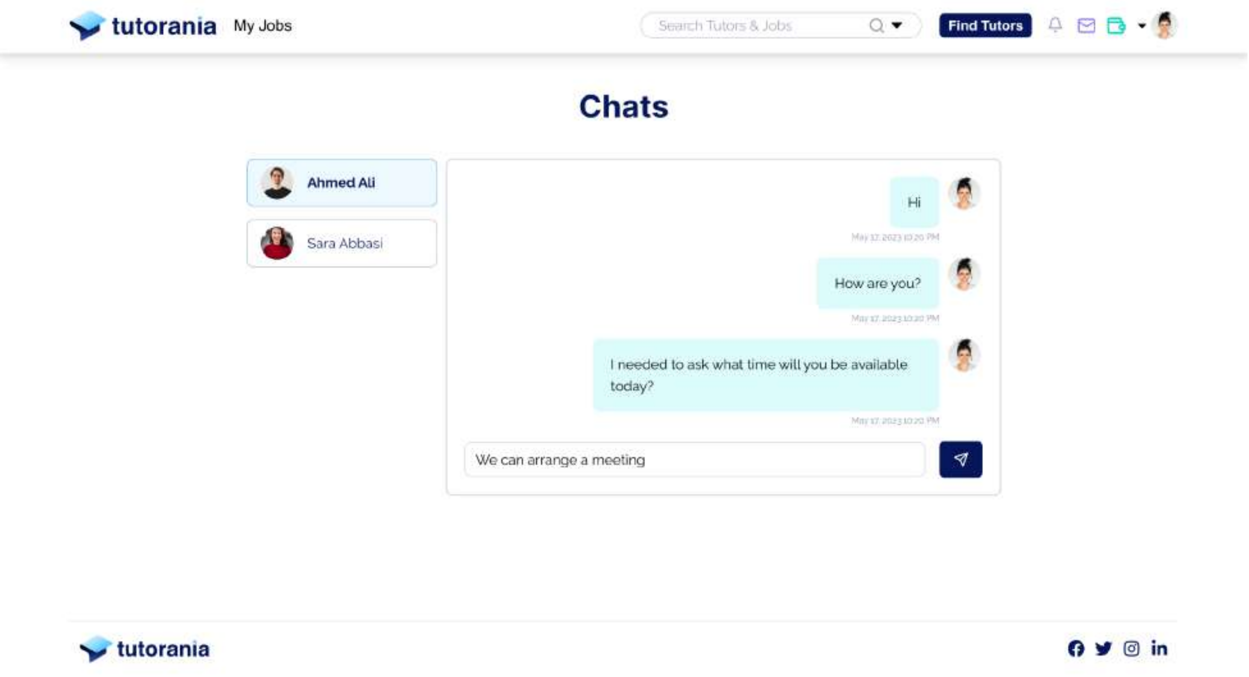
Ayesha Asghar
Please let me know a suitable time
May 12 2023 12:58 PM

Yes sure. I am available at 7 pm today
May 12 2023 12:58 PM

Ayesha Asghar
Thank you
May 12 2023 1:01 PM

Ayesha Asghar

Enter your message



Chats

Ahmed Ali

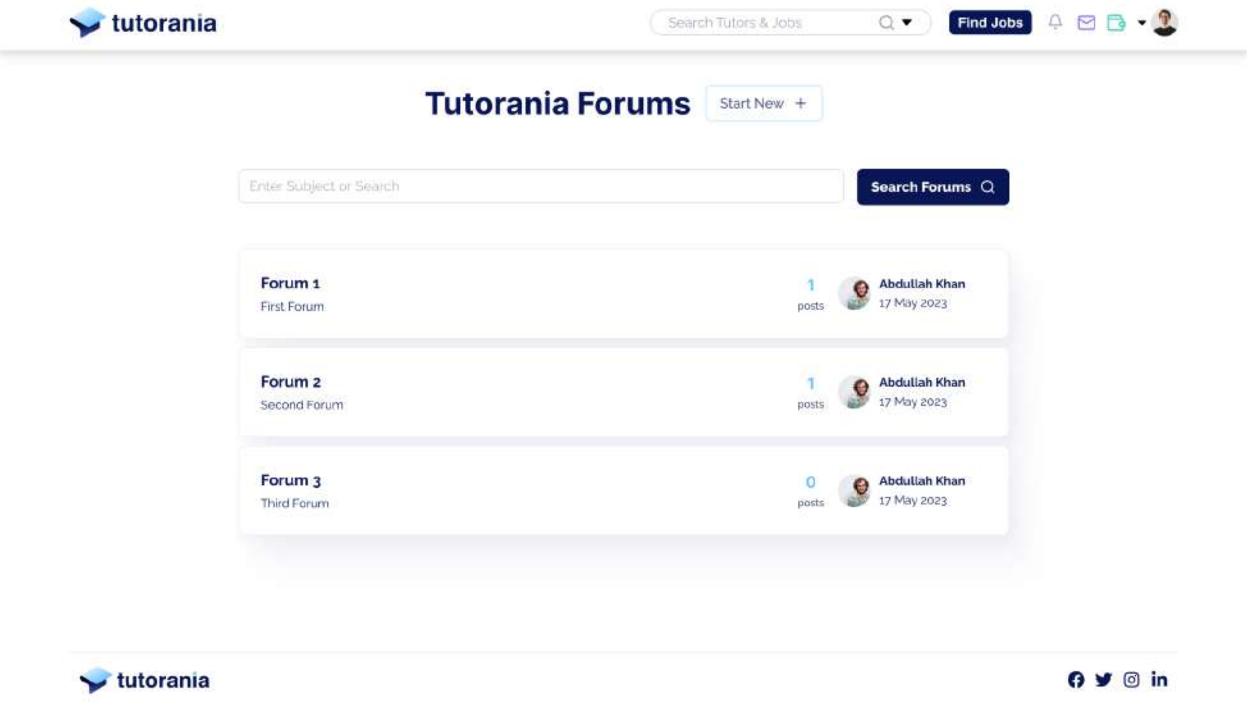
Sara Abbasi

Hi
May 17, 2023 10:20 PM

How are you?
May 17, 2023 10:20 PM

I needed to ask what time will you be available today?
May 17, 2023 10:20 PM

We can arrange a meeting

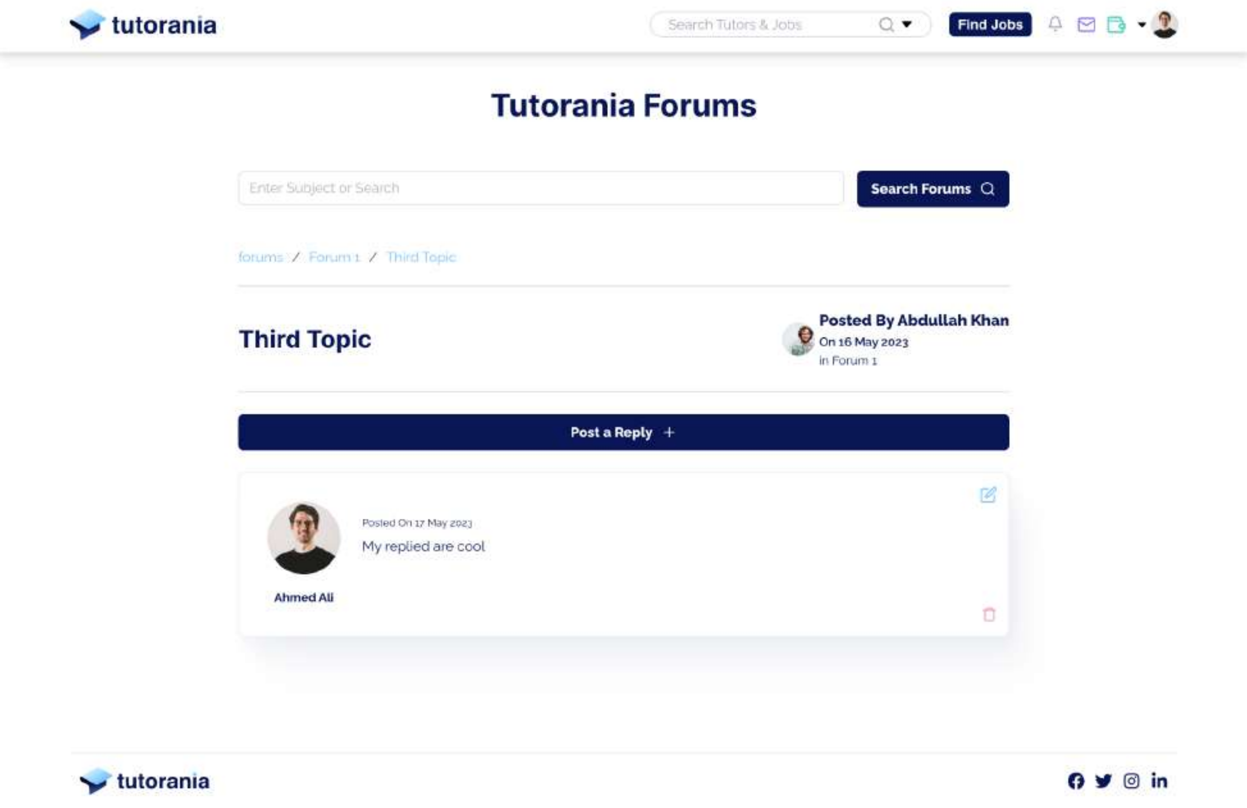


Tutorania Forums

Forum 1
First Forum
1 posts
Abdullah Khan
17 May 2023

Forum 2
Second Forum
1 posts
Abdullah Khan
17 May 2023

Forum 3
Third Forum
0 posts
Abdullah Khan
17 May 2023



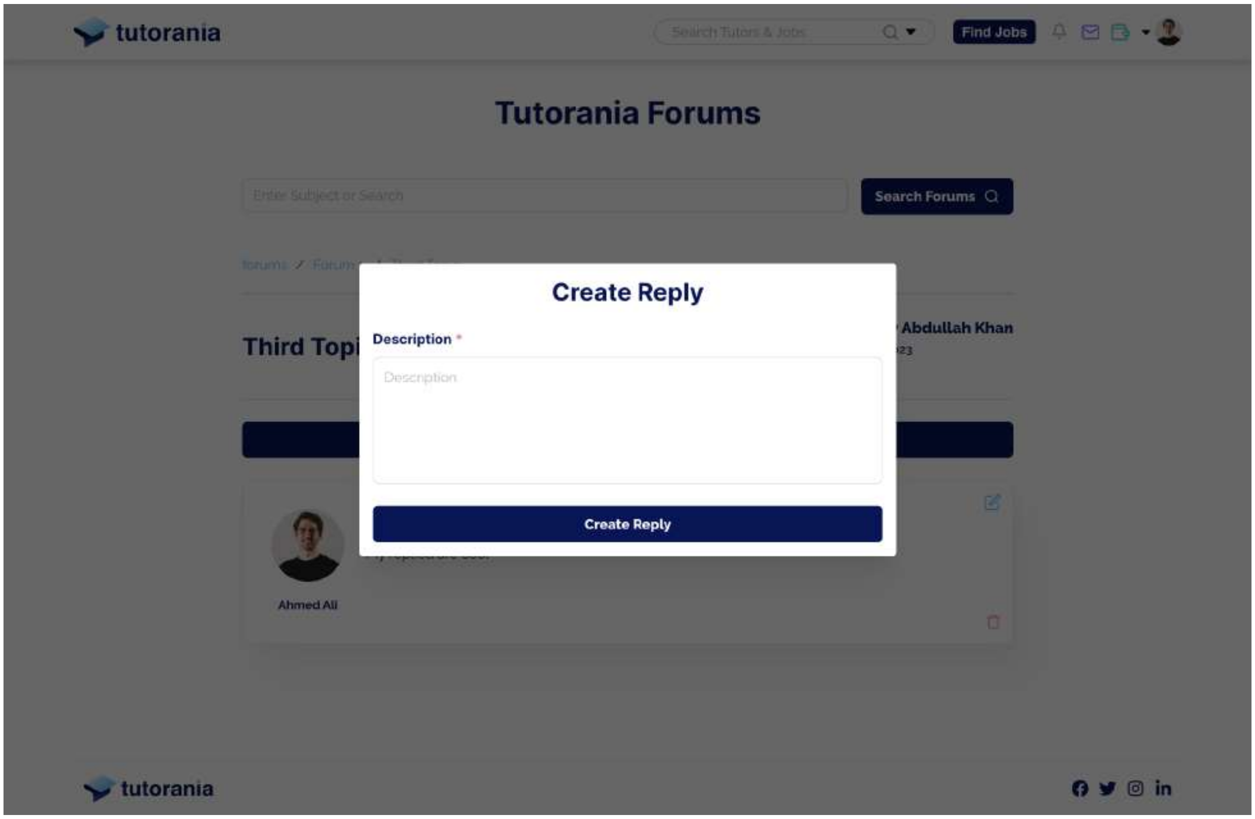
Tutorania Forums

Third Topic

Posted By Abdullah Khan
On 16 May 2023
in Forum 1

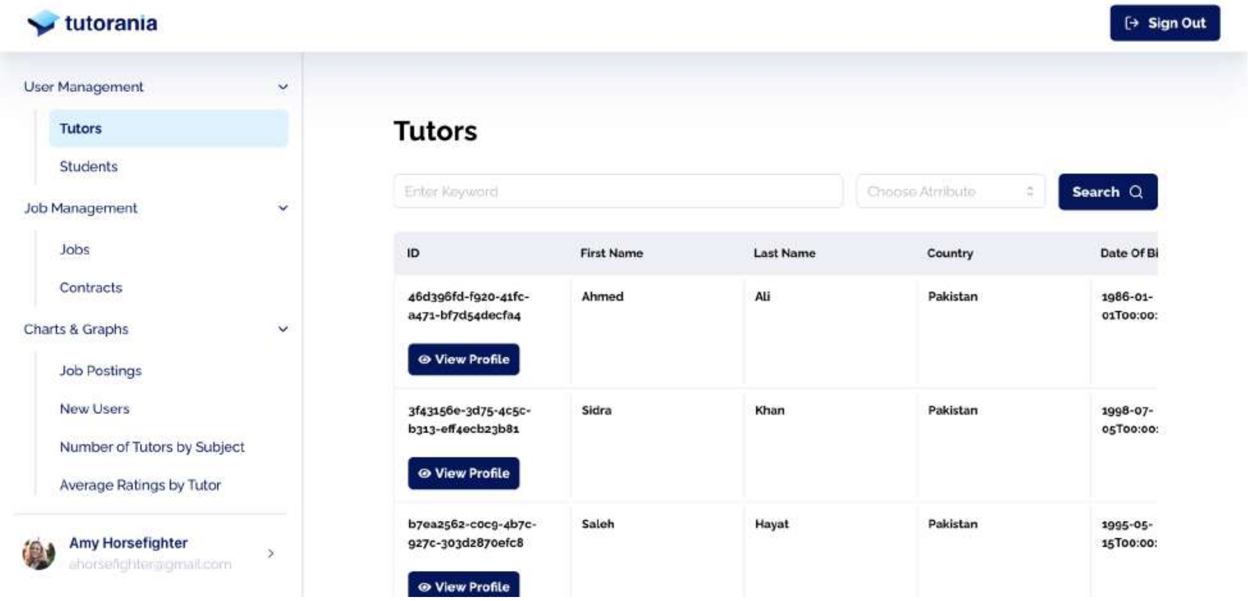
Posted On 17 May 2023
My replied are cool

Ahmed Ali



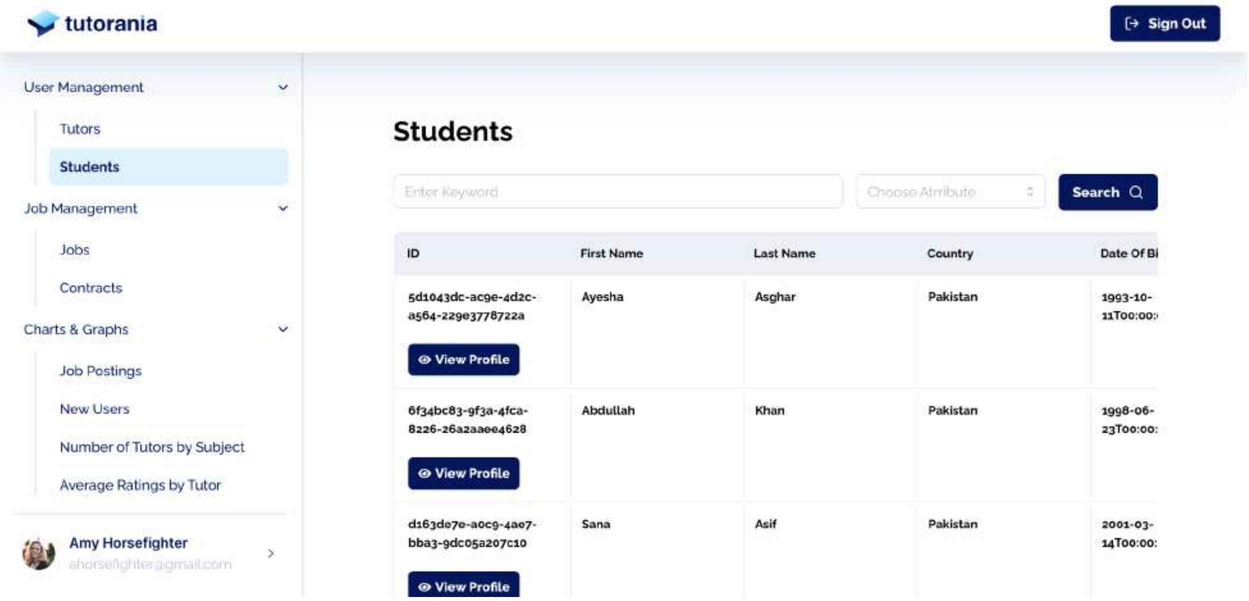
Create Reply

Description *



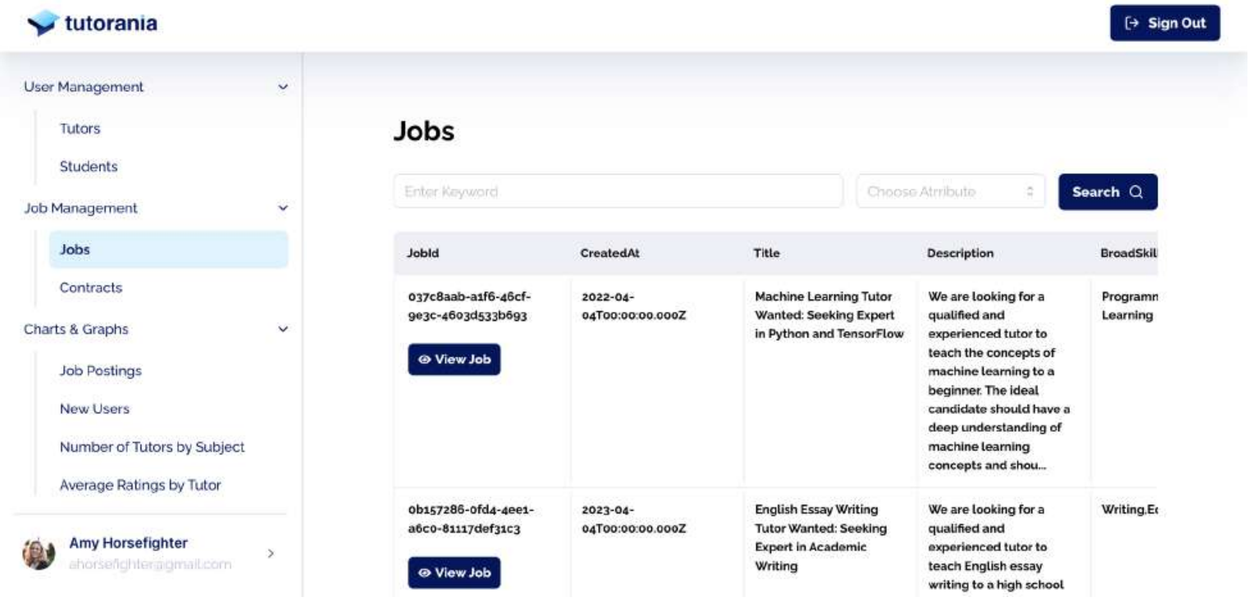
Tutors

ID	First Name	Last Name	Country	Date Of Bi
46d396fd-f920-41fc-a471-bf7d54decfa4 View Profile	Ahmed	Ali	Pakistan	1986-01-01T00:00:
3f43156e-3d75-4c5c-b313-eff4ecb23b81 View Profile	Sidra	Khan	Pakistan	1998-07-05T00:00:
b7ea2562-c0c9-4b7c-927c-303d2870efc8 View Profile	Saleh	Hayat	Pakistan	1995-05-15T00:00:



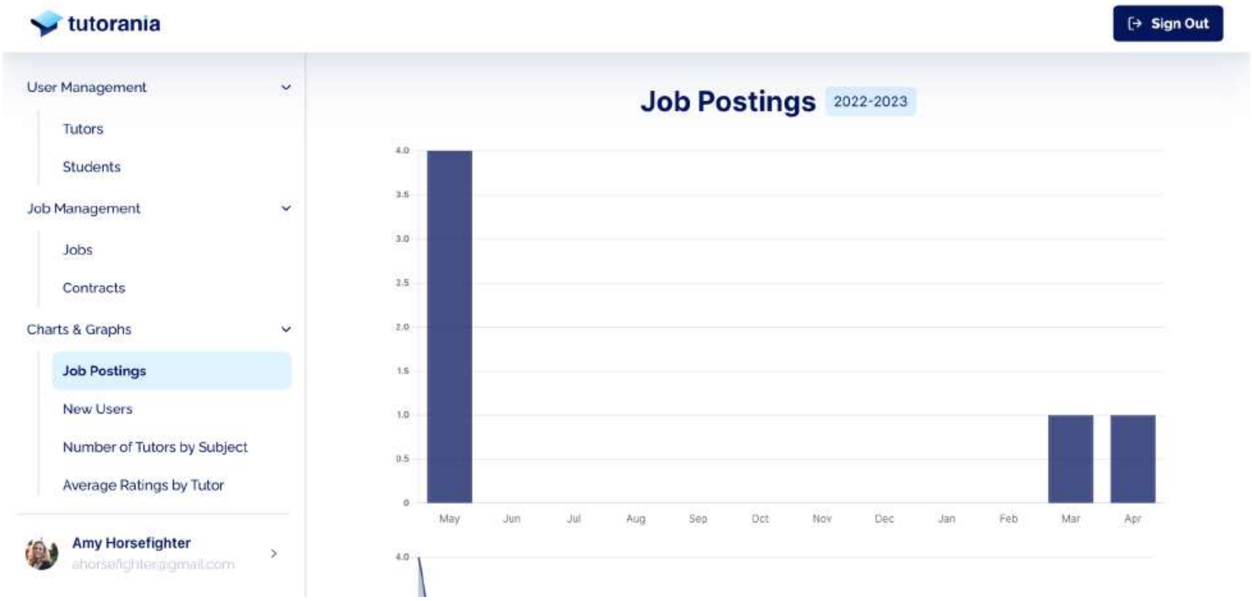
Students

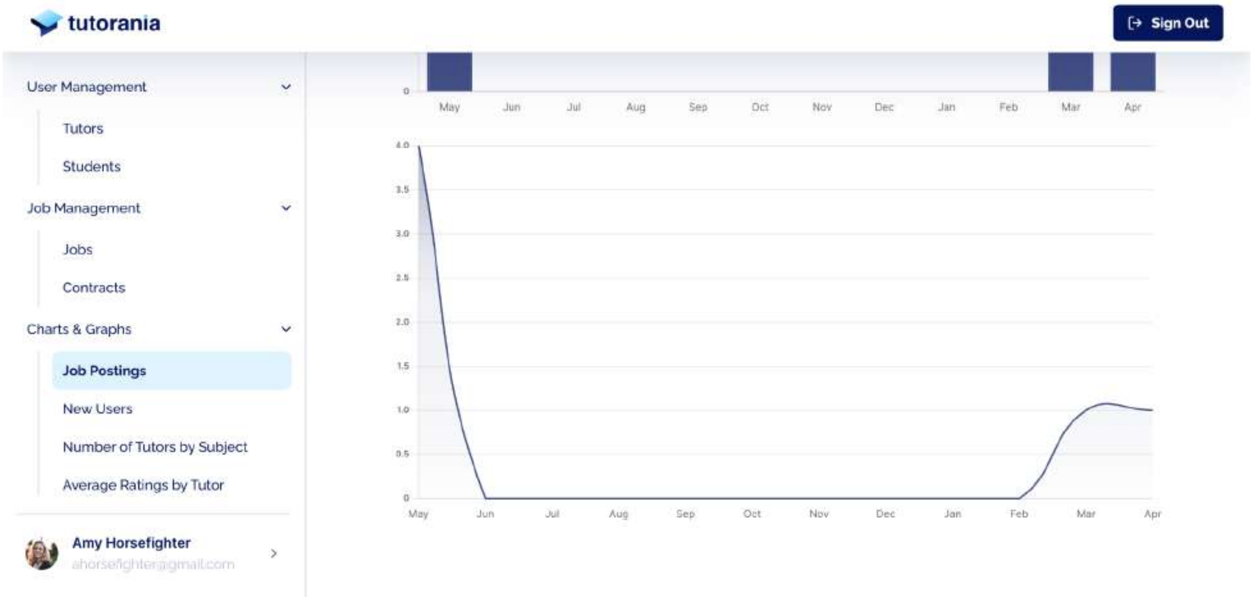
ID	First Name	Last Name	Country	Date Of Bi
5d1043dc-ac9e-4d2c-a564-229e3778722a View Profile	Ayesha	Asghar	Pakistan	1993-10-11T00:00:
6f34bc83-9f3a-4fca-8226-26a2aaee4628 View Profile	Abdullah	Khan	Pakistan	1998-06-23T00:00:
d163de7e-a0c9-4ae7-bba3-9dc05a207c10 View Profile	Sana	Asif	Pakistan	2001-03-14T00:00:

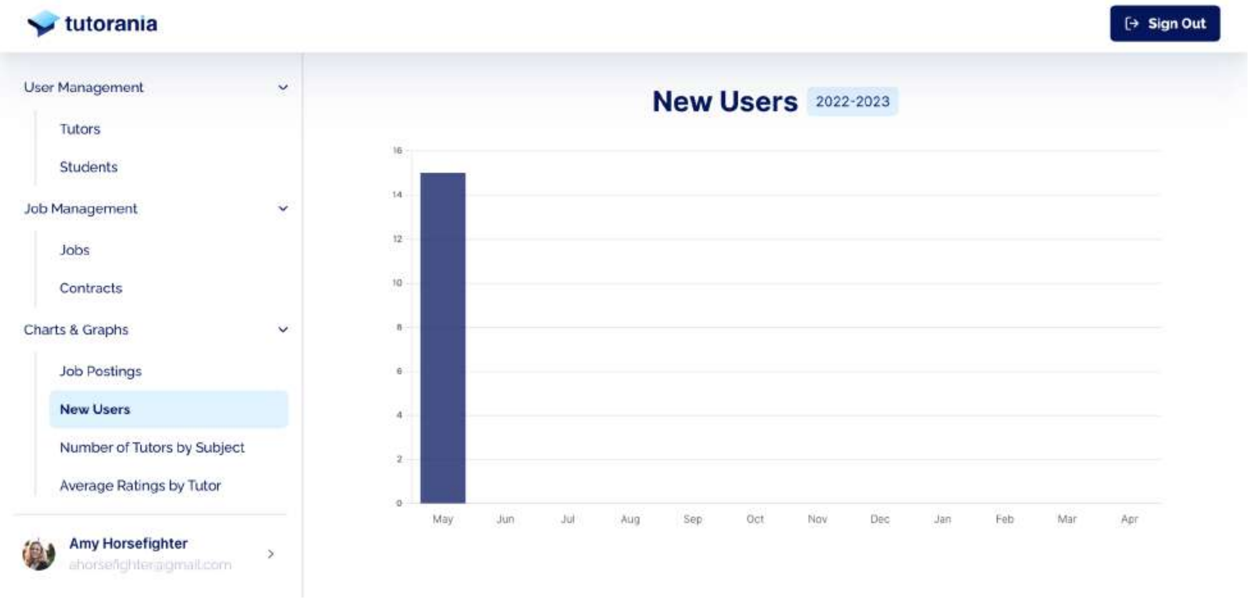


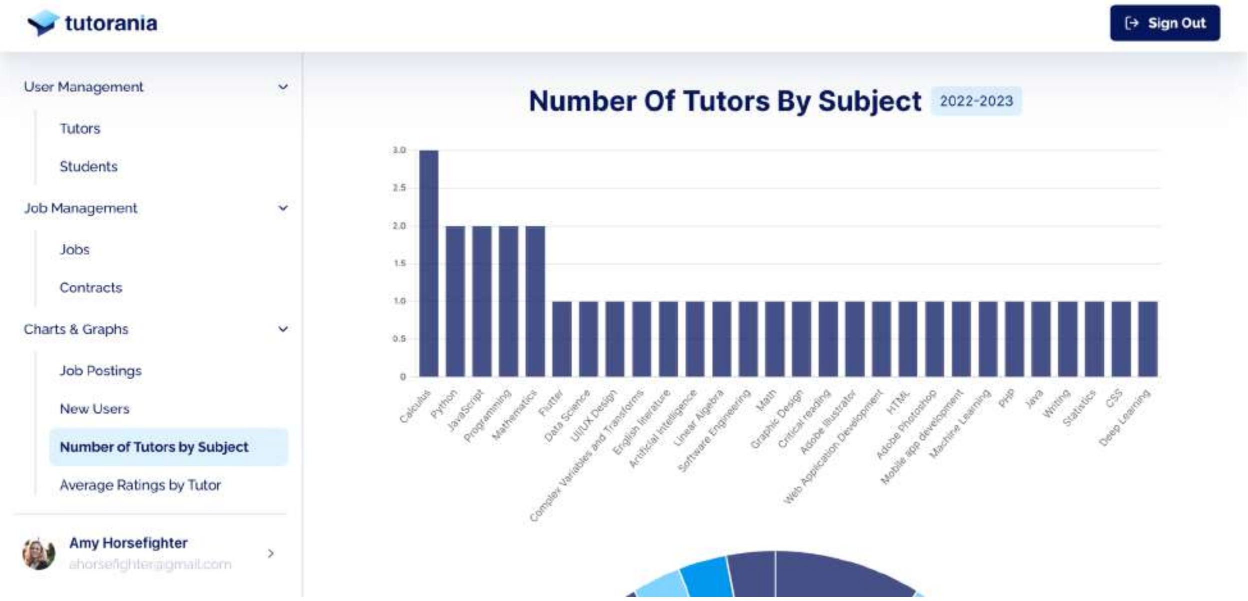
Jobs

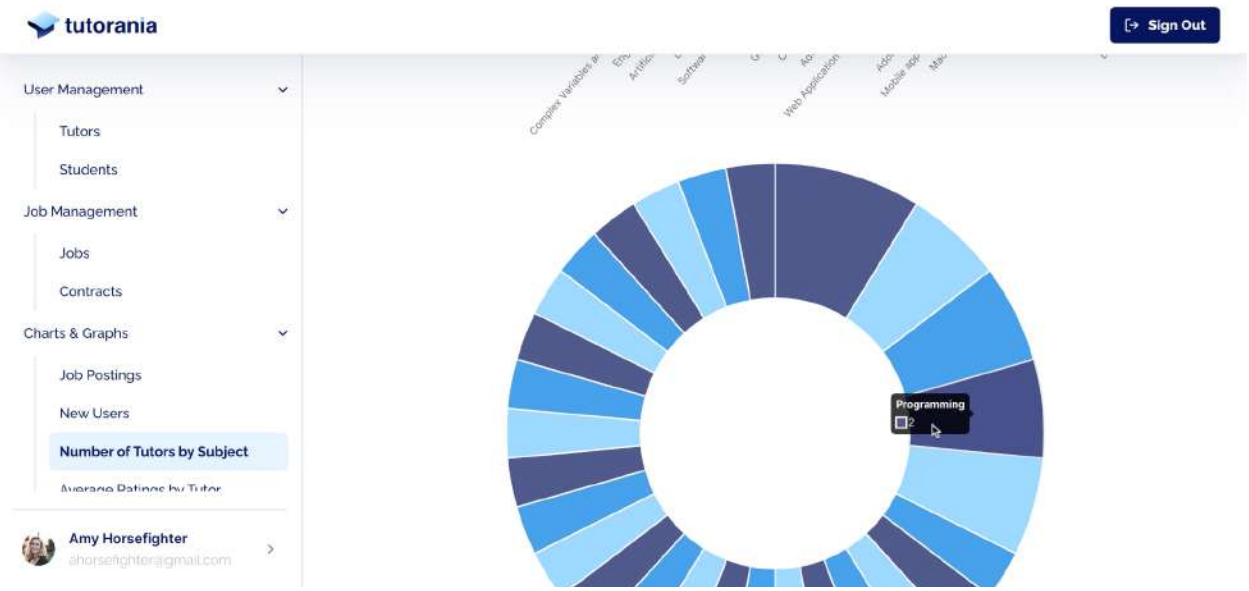
JobId	CreatedAt	Title	Description	BroadSkil
037c8aab-a1f6-46cf-9e3c-4603d533b693 View Job	2022-04-04T00:00:00.000Z	Machine Learning Tutor Wanted: Seeking Expert in Python and TensorFlow	We are looking for a qualified and experienced tutor to teach the concepts of machine learning to a beginner. The ideal candidate should have a deep understanding of machine learning concepts and shou...	Programn Learning
0b157286-0fd4-4ee1-a6c0-81117def31c3 View Job	2023-04-04T00:00:00.000Z	English Essay Writing Tutor Wanted: Seeking Expert in Academic Writing	We are looking for a qualified and experienced tutor to teach English essay writing to a high school	Writing,E

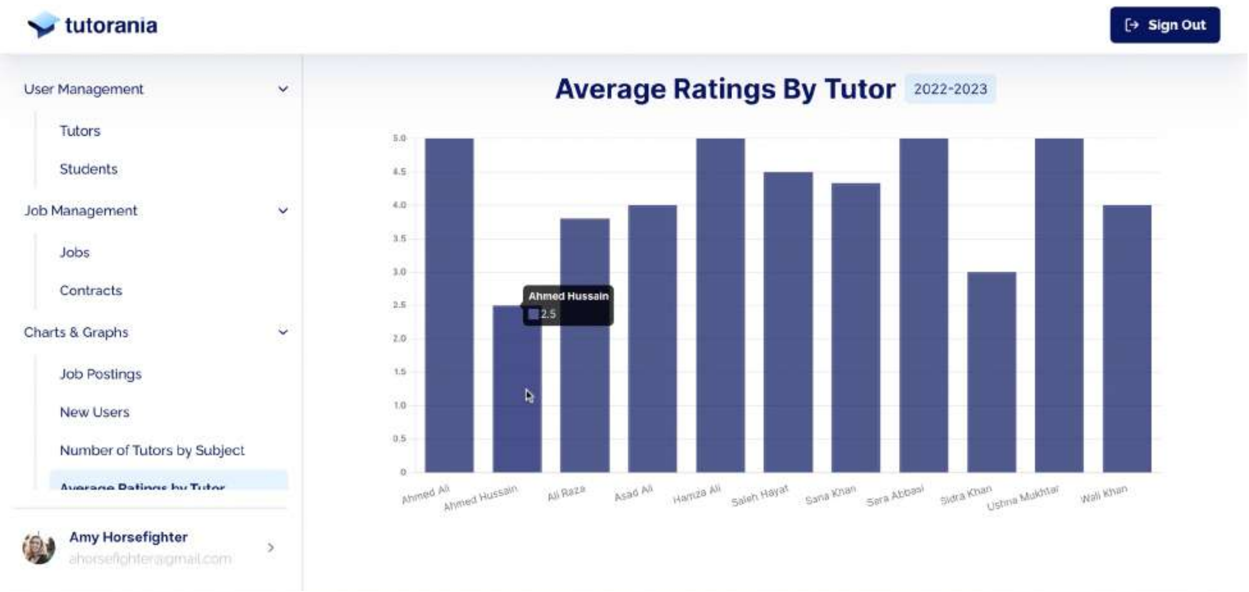

tutorania
Sign Out
User Management
Tutors
Students
Job Management
Jobs
Contracts
Charts & Graphs
Job Postings
New Users
Number of Tutors by Subject
Average Ratings by Tutor
Amy Horsefighter
ahorsefighter@gmail.com
Job Postings
2022-2023
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Jan
Feb
Mar
Apr
4.0

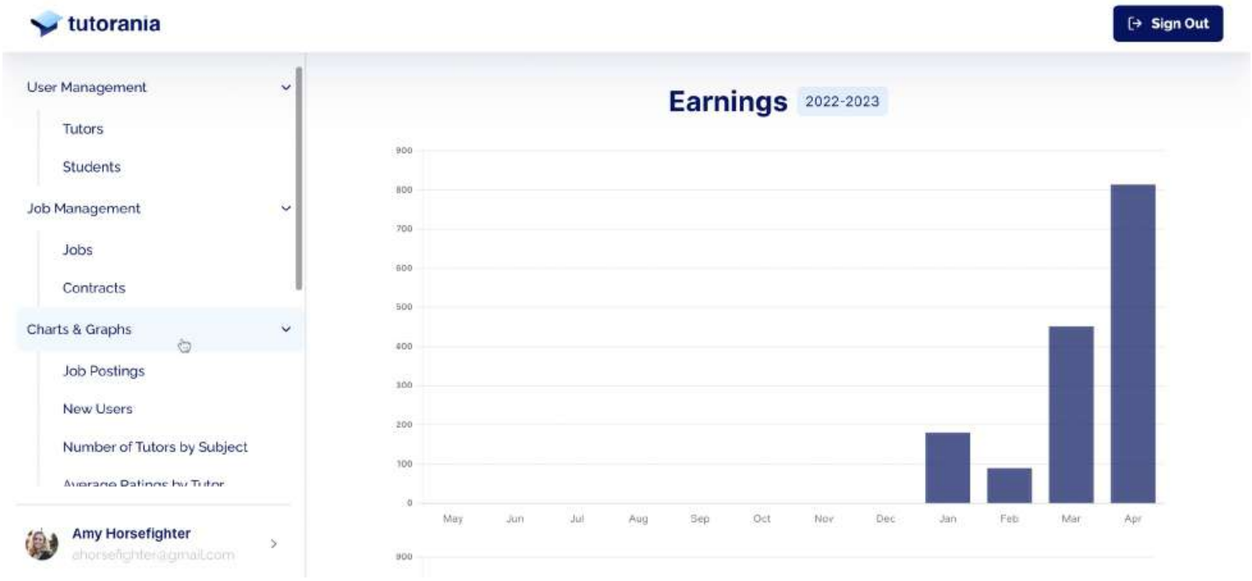

tutorania
Sign Out
User Management
Tutors
Students
Job Management
Jobs
Contracts
Charts & Graphs
Job Postings
New Users
Number of Tutors by Subject
Average Ratings by Tutor
Amy Horsefighter
ahorsefighter@gmail.com
0
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Jan
Feb
Mar
Apr
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Jan
Feb
Mar
Apr

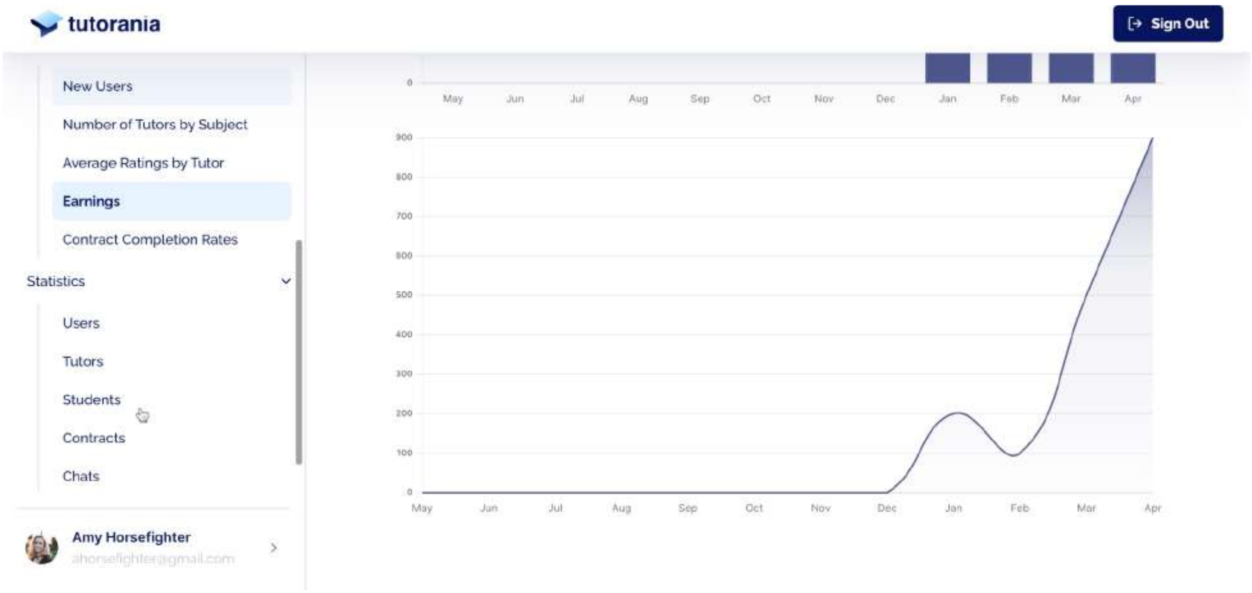

tutorania
Sign Out
User Management
Tutors
Students
Job Management
Jobs
Contracts
Charts & Graphs
Job Postings
New Users
Number of Tutors by Subject
Average Ratings by Tutor
Amy Horsefighter
ahorsefighter@gmail.com
New Users 2022-2023
16
14
12
10
8
6
4
2
0
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Jan
Feb
Mar
Apr

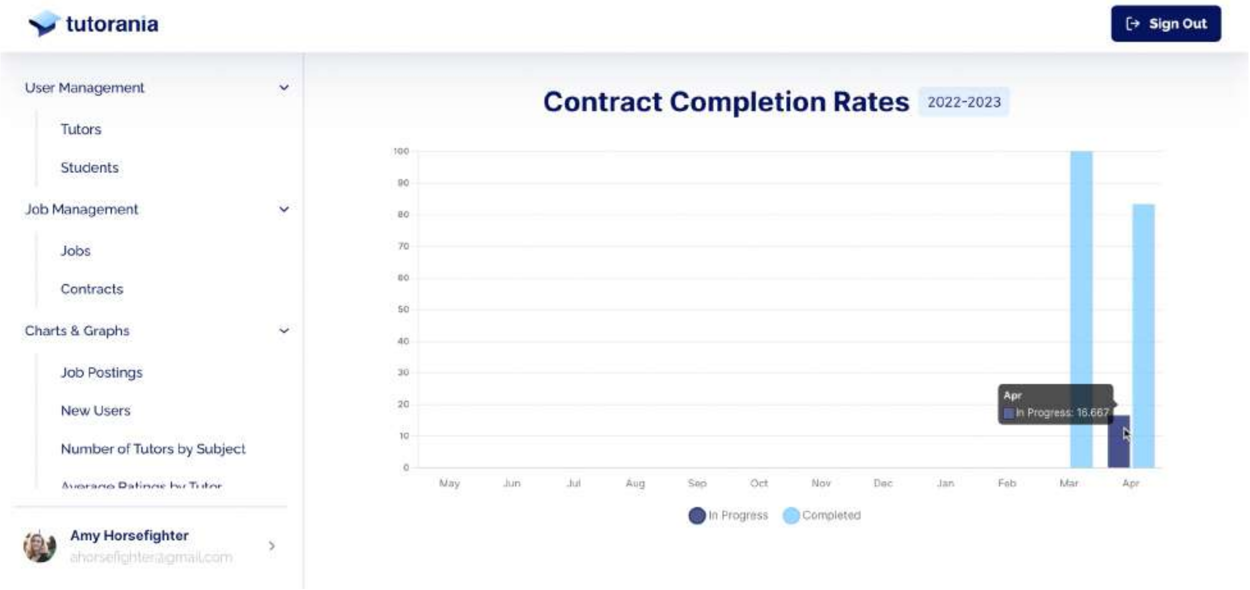

tutorania
Sign Out
User Management
Tutors
Students
Job Management
Jobs
Contracts
Charts & Graphs
Job Postings
New Users
Number of Tutors by Subject
Average Ratings by Tutor
Amy Horsefighter
ahorsefighter@gmail.com
Number Of Tutors By Subject 2022-2023
3.0
2.5
2.0
1.5
1.0
0.5
0
Calculus
Python
JavaScript
Programming
Mathematics
Flutter
Data Science
UI/UX Design
Complex Variables and Transforms
English literature
Artificial intelligence
Linear Algebra
Software Engineering
Math
Graphic Design
Critical reading
Adobe Illustrator
Web Application Development
HTML
Adobe Photoshop
Mobile app development
Machine Learning
PHP
Java
Writing
Statistics
CSS
Deep Learning

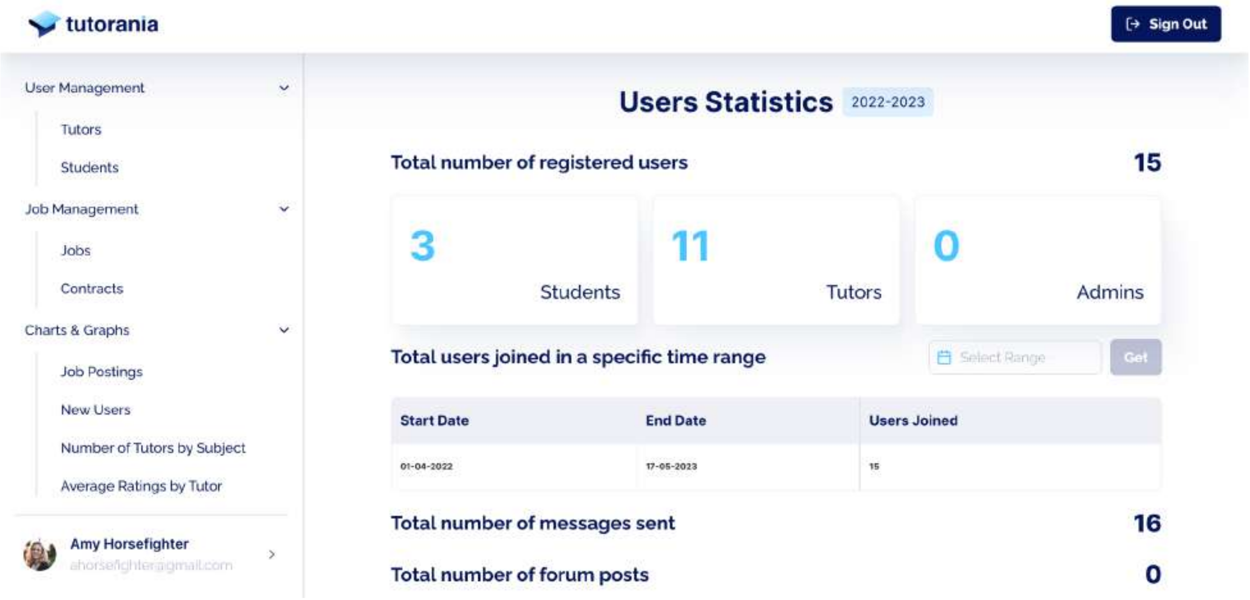

tutorania
Sign Out
User Management
Tutors
Students
Job Management
Jobs
Contracts
Charts & Graphs
Job Postings
New Users
Number of Tutors by Subject
Average Ratings by Tutor
Amy Horsefighter
ahorsefighter@gmail.com
Complex Variables a
En
Artifici
Softwa
G
C
Ac
Web Application
Adc
Mobile app
Mac
Programming
2


tutorania
Sign Out
User Management
Tutors
Students
Job Management
Jobs
Contracts
Charts & Graphs
Job Postings
New Users
Number of Tutors by Subject
Average Ratings by Tutor
Amy Horsefighter
ahorsefighter@gmail.com
Average Ratings By Tutor
2022-2023
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0
Ahmed Hussain
2.5
Ahmed Ali
Ahmed Hussain
Ali Raza
Asad Ali
Hamza Ali
Saleh Hayat
Sana Khan
Sara Abbasi
Sidra Khan
Ushna Mukhtar
Wali Khan


tutorania
Sign Out
User Management
Tutors
Students
Job Management
Jobs
Contracts
Charts & Graphs
Job Postings
New Users
Number of Tutors by Subject
Average Ratings by Tutor
Amy Horsefighter
ahorsefighter@gmail.com
Earnings 2022-2023
900
800
700
600
500
400
300
200
100
0
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Jan
Feb
Mar
Apr


tutorania
Sign Out
New Users
Number of Tutors by Subject
Average Ratings by Tutor
Earnings
Contract Completion Rates
Statistics
Users
Tutors
Students
Contracts
Chats
Amy Horsefighter
ahorsefighter@gmail.com
0
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Jan
Feb
Mar
Apr
900
800
700
600
500
400
300
200
100
0
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Jan
Feb
Mar
Apr





Users Statistics 2022-2023

Total number of registered users 15

3 Students

11 Tutors

0 Admins

Total users joined in a specific time range

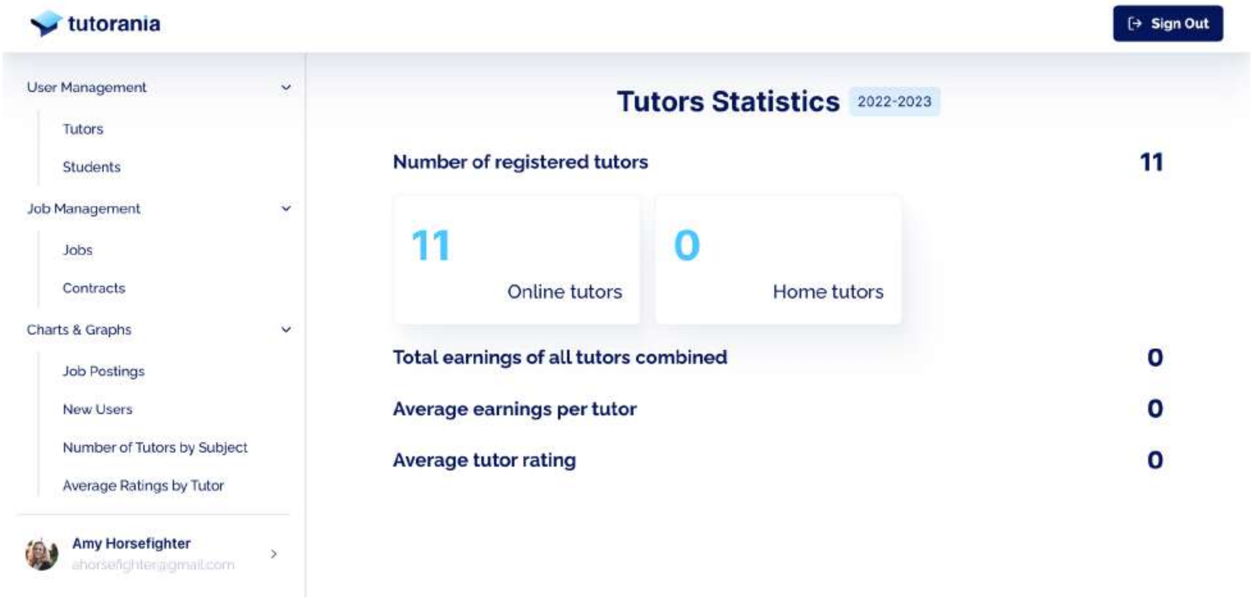
Select Range

Get

Start Date	End Date	Users Joined
01-04-2022	17-05-2023	15

Total number of messages sent 16

Total number of forum posts 0



Tutors Statistics 2022-2023

Number of registered tutors 11

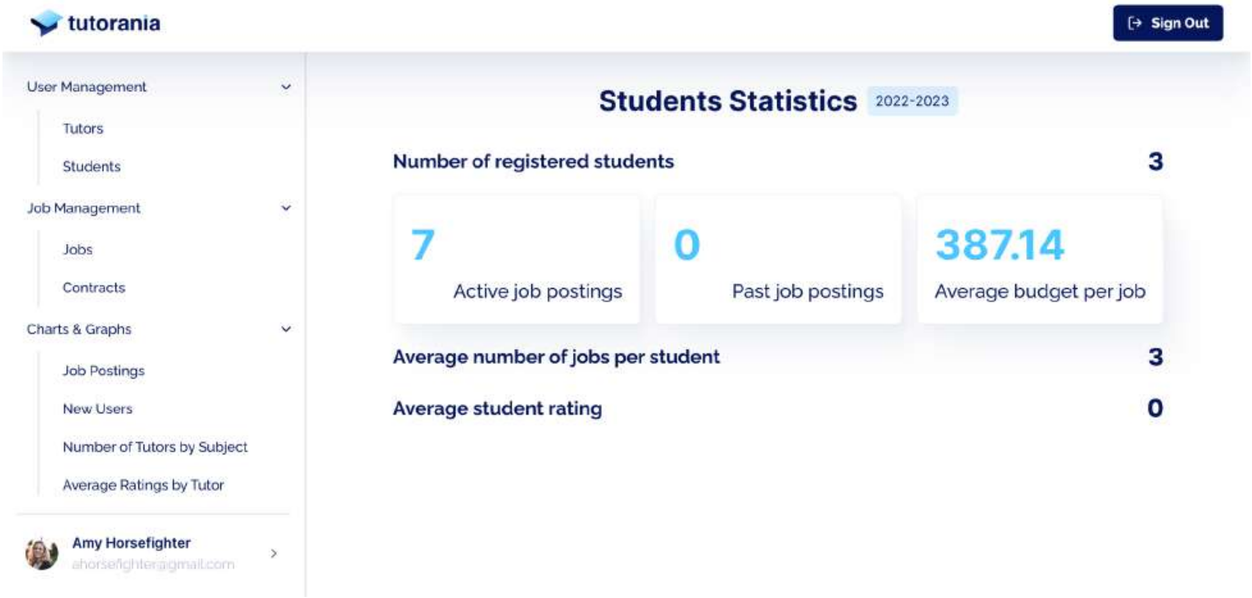
11 Online tutors

0 Home tutors

Total earnings of all tutors combined 0

Average earnings per tutor 0

Average tutor rating 0



Students Statistics 2022-2023

Number of registered students 3

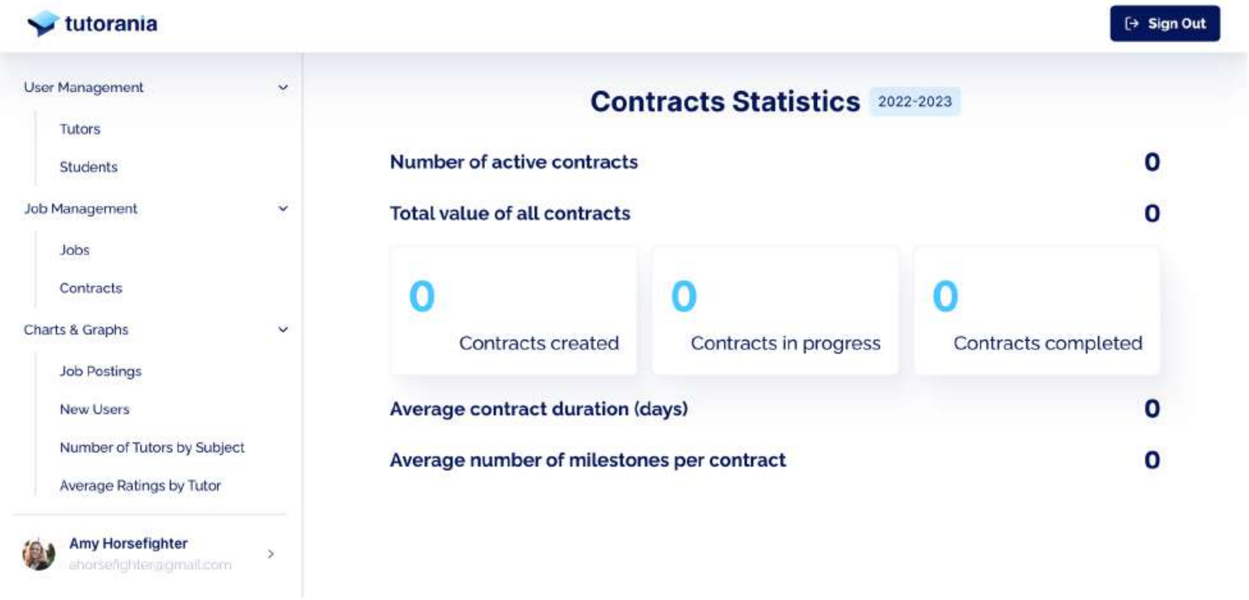
7 Active job postings

0 Past job postings

387.14 Average budget per job

Average number of jobs per student 3

Average student rating 0



Contracts Statistics 2022-2023

Number of active contracts 0

Total value of all contracts 0

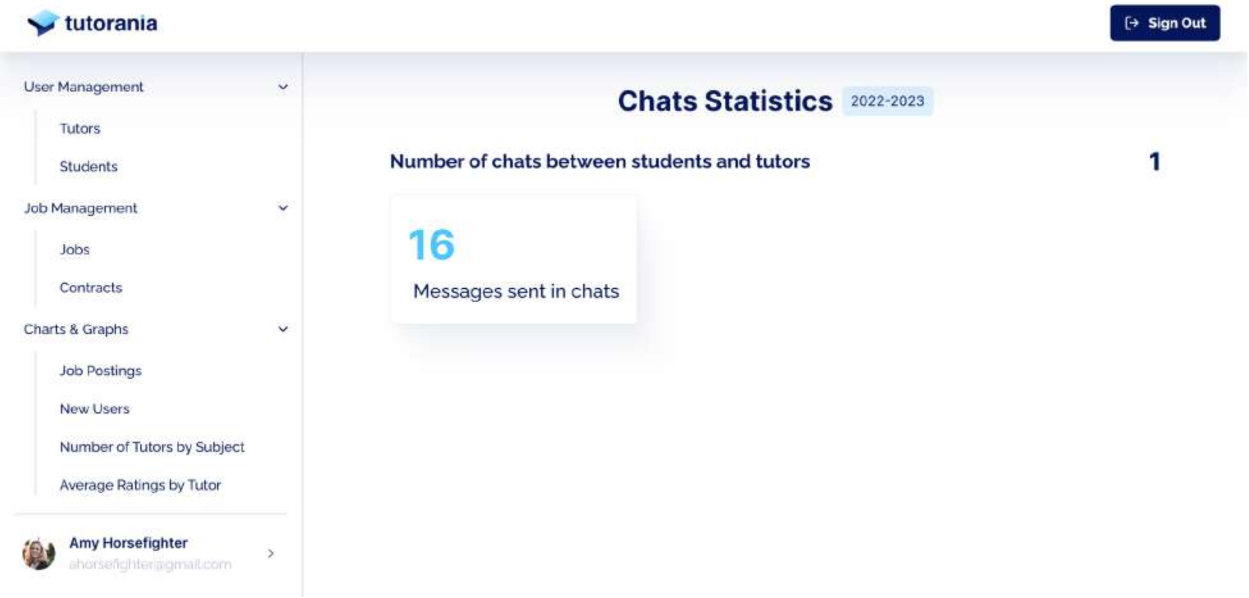
0 Contracts created

0 Contracts in progress

0 Contracts completed

Average contract duration (days) 0

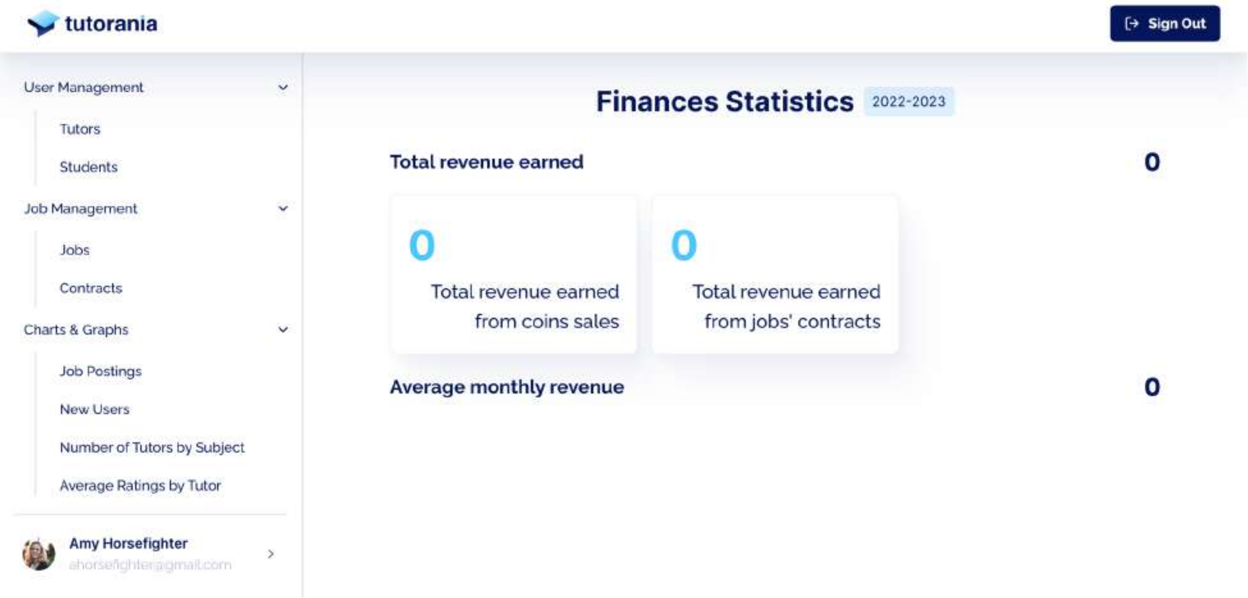
Average number of milestones per contract 0



Chats Statistics 2022-2023

Number of chats between students and tutors 1

16 Messages sent in chats



Finances Statistics 2022-2023

Total revenue earned 0

0 Total revenue earned from coins sales

0 Total revenue earned from jobs' contracts

Average monthly revenue 0