An Undergraduate Internship Project Report on Tutor Express

By

Samiul Alam

Student ID: **1611203**

Summer, 2021

Supervisor:

Md Asif Bin Khaled

Lecturer

Department of Computer Science & Engineering

Independent University, Bangladesh

September 8, 2021

Dissertation submitted in partial fulfillment for the degree of Bachelor of Science in Computer Science

Department of Computer Science & Engineering

Independent University, Bangladesh

Attestation

This is to certify that to the best of my knowledge, the content of this report is my own work. I certify that the intellectual content of this report is the product of my own work and that all the assistance received in preparing this report and sources have been acknowledged.

Samiul Alam	02/09/2021
Signature	Date

Write Your Name Here

Name: Samiul Alam

Acknowledgement

First and foremost, I want to convey my heartfelt gratitude to Almighty Allah; I have come this far because of His mercy and blessing. Working as an Intern for Techdojo has been a fantastic experience. I've gotten a lot of help and encouragement from Techdojo members who have years of experience in software development. I would like to express my gratitude to the members of Techdojo for devoting their time and expertise which was essential to the production of this report.

Throughout my internship and the production of this report, I am grateful to my internal supervisor, Md Asif Bin Khaled, Lecturer, Department of Computer Science and Engineering, Independent University, Bangladesh (IUB), for his invaluable instructions, ongoing advice, support, and inspiration.

Similarly, I want to convey my gratitude to Ms. Shama Hoque, my external supervisor, for allowing me to be a part of this company. This initiative was propelled forward by her enthusiasm and skill to lead.

My thanks also goes out to all of the other Techdojo employees who assisted me in my skill development and made me feel at ease in the workplace. Many thanks to my mentors, Abdullah Omar, Sabah Ashraf, Munjerin Hossain, Prapti Shraboni and Anan Sarah who have guided me and given their time, effort and expert skills.

Finally, I am grateful for my family's, relatives', and friends' significant sacrifices, good wishes, moral support, useful advice, inspirations, and encouragements.

Samiul Alam
September 2021

Letter of Transmittal

2nd September 2021
Md Asif Bin Khaled
Lecturer
Department Of Computer Science and Engineering
Independent University, Bangladesh

Subject: Letter of Transmittal for Internship Report, Summer 2021

Dear Sir,
This is to inform that with due respect and honor, I, Samiul Alam (ID: 1611203) from Internship Program Course of Summer 2021 semester, section: 10, would like to transmit my internship report. This report is based on my internship and the project I worked on during that time. My internship at Techdojo Limited lasted from May 1st to September 1st, 2021, and it was finished successfully.

This report is based on my internship experience and the work I completed at Techdojo Limited. My main goal for the internship was to obtain experience working in the software engineering business and to become familiar with all of the company's technology-related fields, such as documentation, software development, and software development processes.

During my internship at Techdojo Limited, I had to learn and adapt to new technologies that were being used in various scenarios and requirements, as well as apply them to real-world projects. I hope you will find the following report adequate and worthy of your approval.

Sincerely

Samiul Alam, ID: 1611203
Email: 1611203@iub.edu.bd

Evaluation Committee

.....

Signature

.....

Name

.....

Supervisor

.....

Signature

.....

Name

.....

Internal Examiner

.....

Signature

.....

Name

.....

External Examiner

.....

Signature

.....

Name

.....

Convener

Abstract

Amongst the major jobs, there is a certain portion of jobs that is tutoring. There is minimum way to reach someone who is offering tuition job. With that, it also becomes difficult to find the most suitable tutor as per the requirement of a student. In order to overcome this situation, a certain group of developers at Techdojo, along with myself, have determined to work on a web application that makes a direct link between tutor and someone who is offering a tuition job. This web application was decided to be called "Tutor Express". Using MERN technology, the web application offers the user to post a tuition job on their own. Other users can apply to these tuition job posts as their suitability. The application can be viewed by the tuition job post owner. In this way, time of communication with a direct linkage is created between them. "Tutor Express" is able to offer the users to add, view, and apply to post successfully and tends to develop more with time in future.

Keywords— job, tuition, tutor, web

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background of the Work

Young generation, especially students from college and university, are always looking to have an extra amount of money for them. They all have there own reason for it. Tuition jobs has always been a traditional job amongst us. It has been going on for ages. We have our parents who were once tutor at their age, and we still have many relative that has tutoring as profession.

In this newer generation, being a tutor has caught interest of many youths and youngsters. They have identified that an extra hustle and hard work could gain them some benefits. Some think about the expenditures of themselves and do tuition jobs, but there are also many who needs and takes these tuition jobs for a living. There are many university students who pay their tuition fees and other expanses from the money coming from tuition jobs.

With COVID-19 present amongst us for few years, students and youngsters can find difficulties to find tuition jobs due to safety precautions all around and other circumstances. To reach the tuition jobs to the potential tutors, developers at Techdojo along with me, have decided to work on a web application. The web application can offer parents to post a tuition job for their child in their requirements and these will be listed for any user to apply.

This web app is a platform for all the university, college or any individual seeking tuition job. The users register and login. They can create individual tuition posts for other users/tutors to apply. These users will fill a form provided to them according to their needs and what they might look for in a tutor and post the job/tutor request. The other users have the view of all the tuition job posts that are available in this web app. The users/tutors can then apply to any suitable tuition job post. The user/tutor applies from the listing found in the web application. The user/tutor applies and puts on their information for the job selected.

1.2 Objectives

- Make a direct way of posting tuition jobs: Users/Parents can post tuition jobs directly by themselves without in need of third party.
- View Tuition jobs: Tutors can view all the tuition jobs available.
- Apply: Tutors can apply to any tuition jobs they wish.
- View Tutor application: Parents can view who as a tutor applied to their tuition job.

1.3 Scopes

Features available to the users after the development of this web application:

1. Add new tuition job
2. View tuition jobs on Home page
3. View tuition job details
4. Apply to tuition job
5. View tutor job application

Chapter 2

Literature Review

2.1 Relationship with Undergraduate Studies

The development of the "Tutor Express" initiative was aided by knowledge and skills earned in undergraduate studies. It would have been more challenging if these courses had not been completed prior to beginning work on this project.

One of the courses was 'CSE 203 Data Structure', which covered how to handle and operate complicated arrays, objects, classes, arrays of objects, arrays of objects, nested arrays, nested objects, and so on. Because "Tutor Express" has numerous data structures, the expertise gained from it has aided in the modeling of this project's data. The course 'CSE 303 Database Management' provided the initial introduction to project design and planning. System Development Life Cycle (SDLC), Rich Picture, Requirement Analysis, Entity Relationship Diagram (ERD), Business Process Model and Notation Diagram, and others were among the subjects and approaches covered. These strategies aided the project's development planning and strategy. The 'CSE 307 System Analysis' course taught me how to think as if I were a user. When considering the User Interface, I always considered the convenience of use, the colors to be seen, and so on. This course covered a variety of strategies that have prompted me to consider the importance of user interfaces. This project is based on the 'CSE 309 Web Applications and Internet' course. It covered the most essential technologies in the IT sector and contributed the most to this project. HTML, CSS, JavaScript, jQuery, Node.js, Express, MongoDB, and others are among them. These technologies are important because the project is a full stack web application with both a front end and a back end.

2.2 Related works

'caretutors.com' is a web app that offers tuition jobs for people. The app involves human involvement from both side, that is parents request and the request is later on added by the company, and tutors apply and is later on contacted by the company [1]. 'tutorsheba.com' is another web app that offers tuition job similarly like 'caretutors.com' but here they have kept an option of requesting for tutor without signing up [2].

"Tutor Express" aims to keep it all between the users, parents post up the tuition job directly and tutors who apply will be contacted by the parents directly. "Tutor Express" is trying to eliminate that and giving the option of putting up the exact request to the web application. Users sign in and fill out a form of important details to apply.

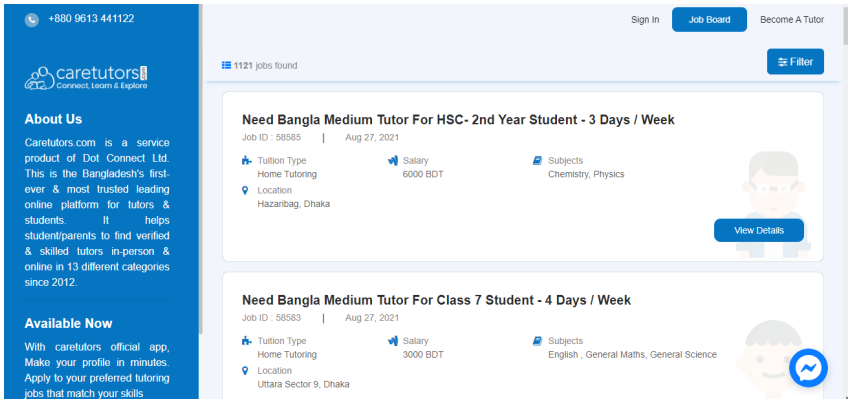


Figure 2.1: UI of caretutors.com

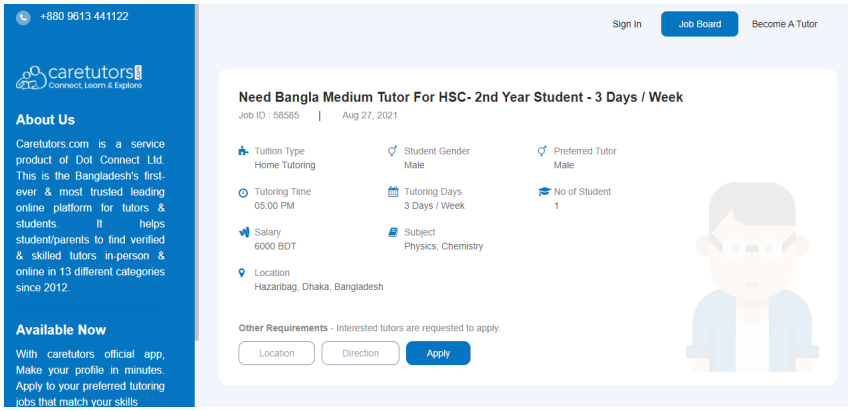


Figure 2.2: UI of caretutors.com

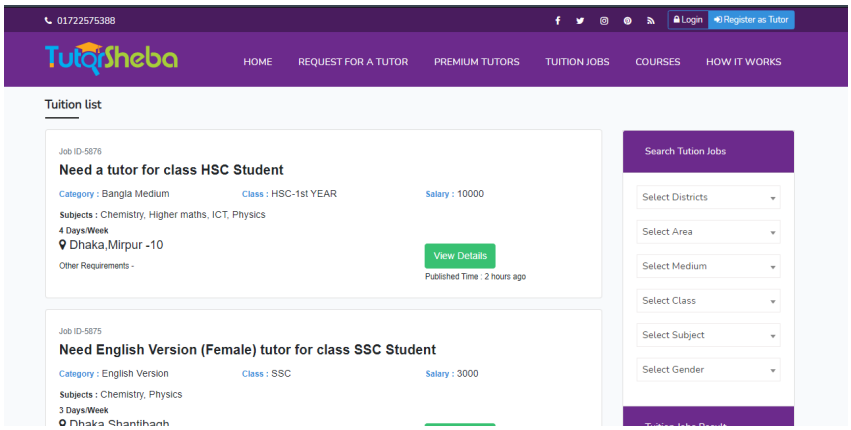


Figure 2.3: UI of tutorsheba.com

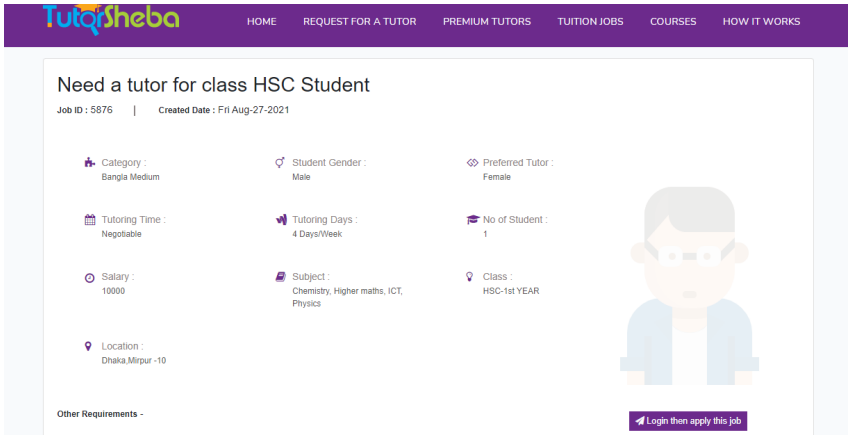


Figure 2.4: UI of tutorsheba.com

Chapter 3

Project Management & Financing

3.1 Work Breakdown Structure

A work breakdown structure (WBS) is a deconstruction of a project that is visible, hierarchical, and focused on deliverables. It is a useful diagram for project managers because it helps them to work backwards from a project's end product to identify all of the actions required to complete the project successfully.

Every one of the means of a venture are illustrated in the hierarchical outline of a work breakdown structure, which makes it a fundamental undertaking the executives instrument for arranging and booking. The last deliverable lays on top of the outline, and the levels beneath partition the venture extension to demonstrate the stages, deliverable and undertakings that are expected to finish the task.

Separating into piece works gains the task headway quicker and simpler as work is split between the group and equal work is conceivable. The following is the WBS for "Tutor Express". It shows the different errands in each stages in a breakdown way. These works were done by people and in bunches at helpful time.

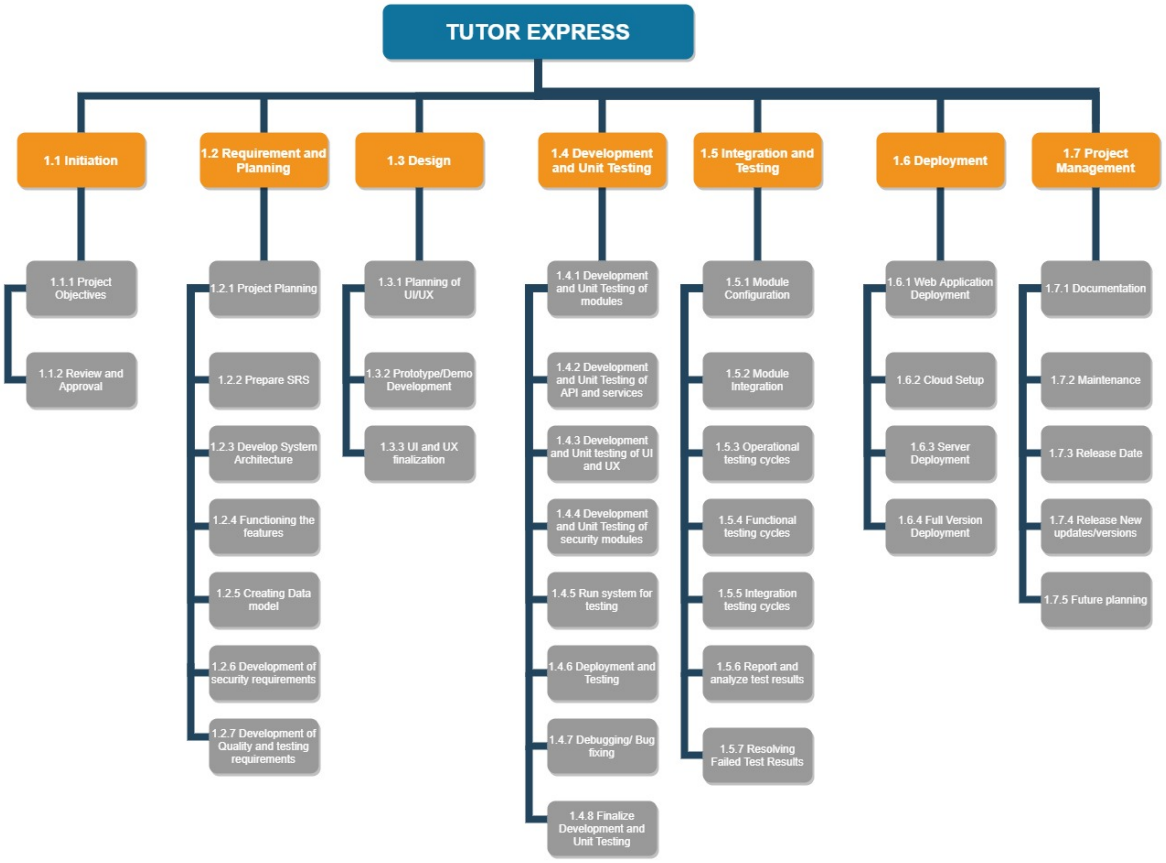


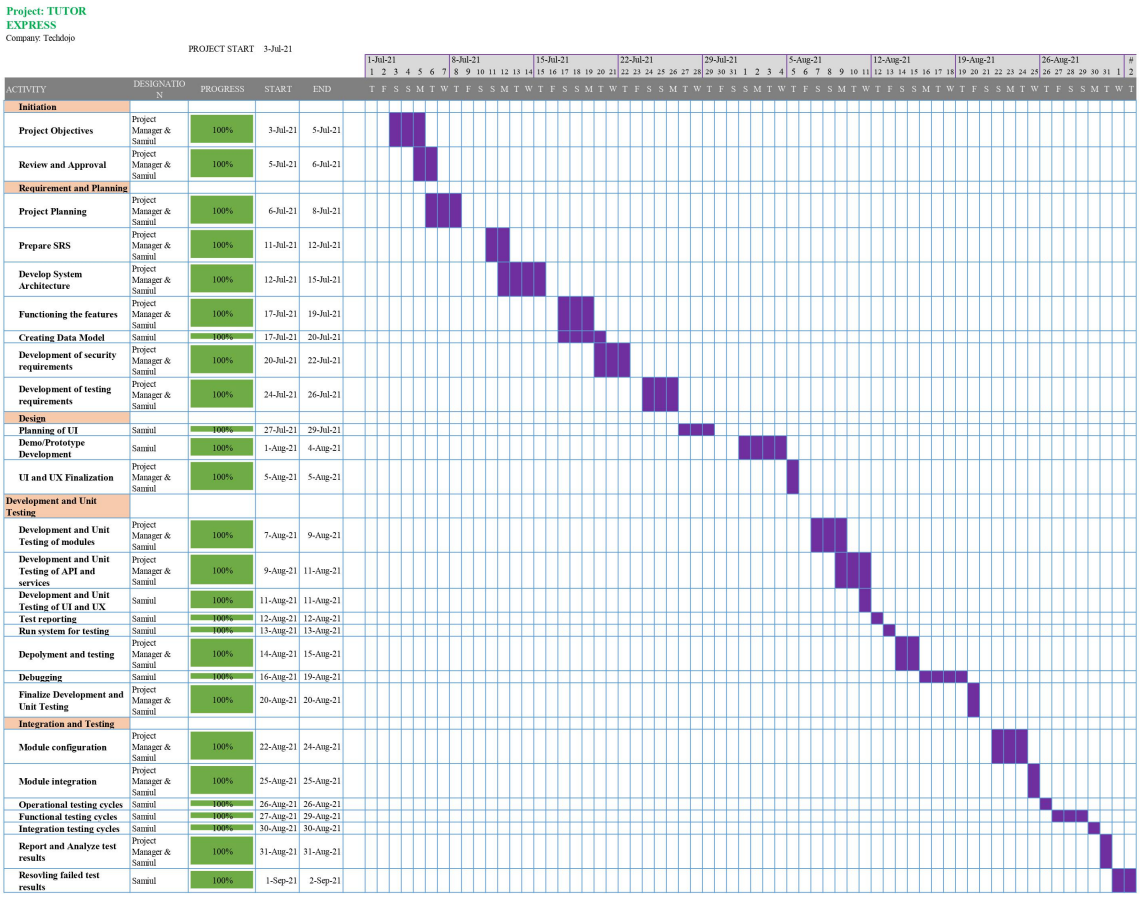
Figure 3.1: Work Breakdown Structure of Tutor Express

3.2 Gantt Chart

To plan and schedule project task, a Gantt chart has been created. It helped create a timeline of the project and list the order in which tasks should be completed. The Gantt chart also helped to keep track of the progress from the beginning. This helped to set goals and targeted time for each task and also monitor the delay in certain task if occurs.

Below is the Gantt Chart for "Tutor Express".

The Gantt Chart shows the distribution, timeline and the progress of the project currently. I have been involved during the the initiation phase where we discussed and finalized the project idea, features and the way to proceed the targets. In requirement and planning phase, I carried out creating the models for the project. As a team, we also discussed the feasibility of the features that is to be functioned. The designing phase was carried out by myself that included panning and prototyping the UI for the project, followed by finalizing the UI as a team. Development and unit testing was carried out step by step. There are time when we had to wait for one module to be completed, for another module to function. Testing phase had times when UI and APIs were working or not had to be checked. The Gantt Chart here shows all those in details.



ACTIVITY	DESIGNATION	PROGRESS	START	END
Initiation				
Project Objectives	Project Manager & Samiul	100%	3-Jul-21	5-Jul-21
Review and Approval	Project Manager & Samiul	100%	5-Jul-21	6-Jul-21
Requirement and Planning				
Project Planning	Project Manager & Samiul	100%	6-Jul-21	8-Jul-21
Prepare SRS	Project Manager & Samiul	100%	11-Jul-21	12-Jul-21
Develop System Architecture	Project Manager & Samiul	100%	12-Jul-21	15-Jul-21
Functioning the features	Project Manager & Samiul	100%	17-Jul-21	19-Jul-21
Creating Data Model	Samiul	100%	17-Jul-21	20-Jul-21
Development of security requirements	Project Manager & Samiul	100%	20-Jul-21	22-Jul-21
Development of testing requirements	Project Manager & Samiul	100%	24-Jul-21	26-Jul-21
Design				
Planning of UI	Samiul	100%	27-Jul-21	29-Jul-21
Demo/Prototype Development	Samiul	100%	1-Aug-21	4-Aug-21
UI and UX Finalization	Project Manager & Samiul	100%	5-Aug-21	5-Aug-21
Development and Unit Testing				
Development and Unit Testing of modules	Project Manager & Samiul	100%	7-Aug-21	9-Aug-21
Development and Unit Testing of API and services	Project Manager & Samiul	100%	9-Aug-21	11-Aug-21
Development and Unit Testing of UI and UX	Samiul	100%	11-Aug-21	11-Aug-21
Test reporting	Samiul	100%	12-Aug-21	12-Aug-21
Run system for testing	Samiul	100%	13-Aug-21	13-Aug-21
Depolyment and testing	Project Manager & Samiul	100%	14-Aug-21	15-Aug-21
Debugging	Samiul	100%	16-Aug-21	19-Aug-21
Finalize Development and Unit Testing	Project Manager & Samiul	100%	20-Aug-21	20-Aug-21
Integration and Testing				
Module configuration	Project Manager & Samiul	100%	22-Aug-21	24-Aug-21
Module integration	Project Manager & Samiul	100%	25-Aug-21	25-Aug-21
Operational testing cycles	Samiul	100%	26-Aug-21	26-Aug-21
Functional testing cycles	Samiul	100%	27-Aug-21	29-Aug-21
Integration testing cycles	Samiul	100%	30-Aug-21	30-Aug-21
Report and Analyze test results	Project Manager & Samiul	100%	31-Aug-21	31-Aug-21
Resovling failed test results	Samiul	100%	1-Sep-21	2-Sep-21

Figure 3.2: Gantt Chart for Tutor Express project

3.3 Estimated Costing

No.	Description	Unit	Quantity	Cost/unit	Total Cost(BDT)
1	Laptop	PC	2	70000	140000
2	Internet	Package	6	1500	4500
3	Electricity	Package	6	1000	3000
4	Developer	Month	1	10000	30000
5	Cloud Server	Package	1	2500	2500
					= 180000 BDT

Table 3.1: Estimated Costing for the project

The project is a skeleton project for the company. It is worked for future purpose where client can use it as a base. However, an estimated budgeting is prepared. We are allocated with laptop for programming and development purpose, and to run these, internet and electricity cost have been estimated. Developer cost is the salary during the project timeline. Cloud server cost is thought for a single month as it is not planned to be deployed.

Chapter 4

Methodology

4.1 Software Development Methodology

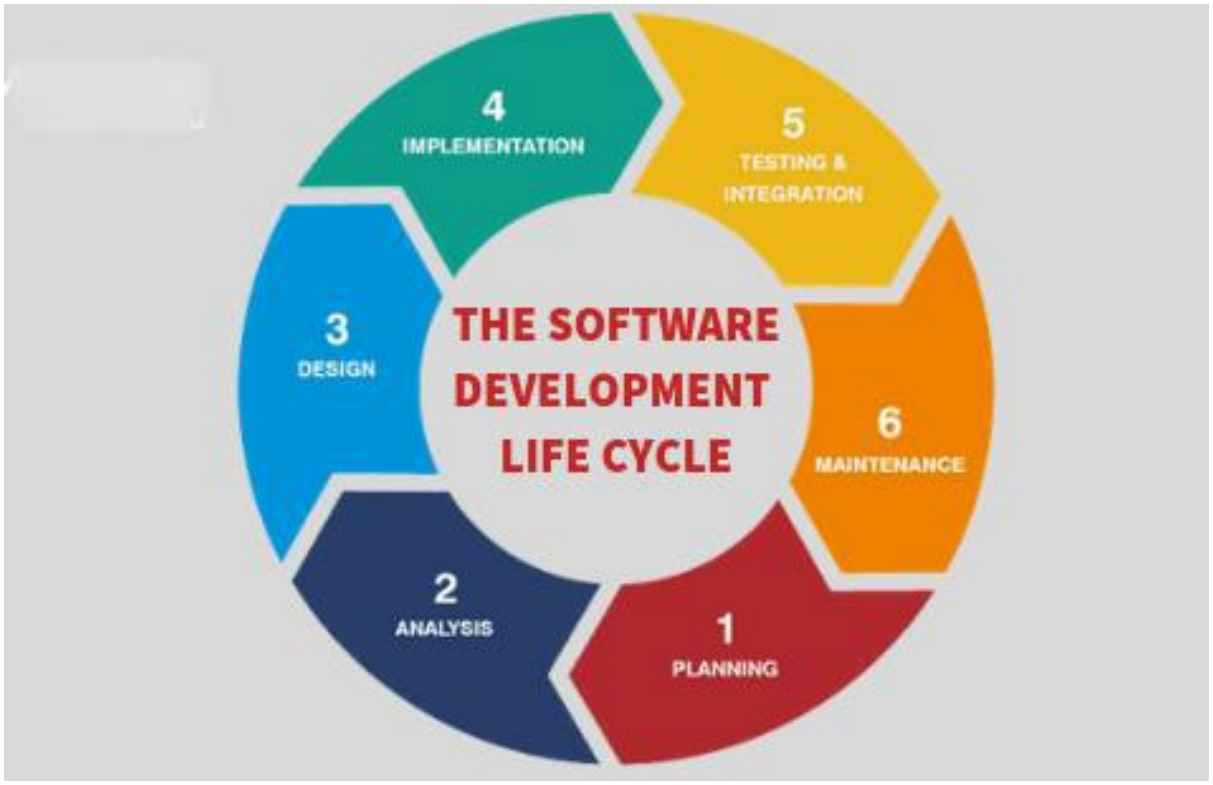


Figure 4.1: SDLC

Software development life cycle (SDLC) is a process maintained in a software project by the software companies [3] [4]. It involves stages that focuses on the plan, development, maintenance and enhancement of the software. The SDLC proposes methods for improvement of the quality of the software and overall growth process.

- Planning
- Analysis
- Design
- Implementation
- Testing and Integration
- Maintenance

Based on client requirements, an SDLC seeks to create high-quality systems that meet or exceed customer expectations by delivering systems that go through each clearly defined step within scheduled time frames and cost estimates [5].

There are several software development methodologies or models that are used in development among them we have used the Agile method to develop my project.

Agile is a collection of methods in software development that aims to increase the effectiveness of software development professionals, teams, and organizations. It features self-organizing and cross-functional teams working together with their customers/end users to identify requirements and build solutions(s). It promotes flexible solutions to changes in needs, resource availability, and understanding of the challenges to be solved, as well as adaptive planning, evolutionary development, early delivery, and continuous improvement. The most widely used Agile methodologies are:

1. Agile scrum methodology
2. Lean Software Development
3. Kanban
4. Extreme Programming (XP)
5. Crystal
6. Dynamic Systems Development Method (DSDM)
7. Feature Driven Development (FDD)

Because Agile Methodology is a form of project management technique, primarily used for software development, where needs and solutions grow through the collaborative effort of self-organizing and cross-functional teams and their clients, I chose it to build the project. The Agile SDLC approach divides the product into cycles and produces a working product in a short amount of time. Agile takes a more rapid development approach and accelerates the development process in a way that is both effective and efficient. We developed this software using Extreme Programming (XP) for this project.

4.1.1 Extreme Programming

Extreme Programming(XP) encourages frequent "releases" in short development cycles, with the goal of increasing productivity and introducing checkpoints where new client needs can be implemented [6]. Extreme programming also involves working in pairs or conducting extensive code reviews, unit testing all code, postponing programming features until they are actually required, a flat management structure, code simplicity and clarity, anticipating changes in the customer's requirements as time passes and the problem becomes better understood, and frequent communication with the customer and among programmers. The entire procedure is transparent and responsible. Developers make commitments about what they will accomplish and track their progress. XP has 5 simple values. The five values of XP are Communication, Simplicity, Feedback, Courage and Respect. They are described in more detail below:

- Communication: Inherently, software development is a team sport that relies on communication to communicate information from one team member to the rest of the team. XP emphasizes the importance of the right form of communication — face-to-face discussion using a white board or other drawing tool.
- Simplicity: Simplicity is thought to be "what is the simplest thing that will work?" The goal is to reduce waste and accomplish just what is absolutely necessary, such as keeping the system architecture as simple as possible to make it easier to maintain, support, and change.
- Feedback: Teams can find areas for development and update their procedures by receiving continual feedback on their previous efforts. Simple design is also supported by feedback.
- Respect: To communicate with one another, provide and accept feedback, and work together to develop simple concepts and solutions, team members must respect one another.
- Courage: To bring up organizational challenges that are limiting a team's effectiveness, courage is required. Stop doing something that isn't working and try something new, as well as accept and act on feedback, even if it is difficult to accept.

During our project, we could not manage a face-to-face interaction at all times due to safety circumstances, but always held a meeting when needed, to discuss solutions, ideas and progress of the project. I have only thought about keeping the scopes of the project with necessity factors only such as adding and viewing feature to be friendly and simple. The framework is kept neat and clean by all of us, so that it is easier to work for everyone in future. I, along the team produced something, collects feedback on its design and implementation, and then tweaks it in the future. Communication was maintained at everyone's time flexibility and there is always a sense of respect for each other in the team, despite being senior or junior. Courage was always lifted amongst the team member by one or another. There were times when a piece work is not performing at all or not well. But team members acted on the feedback and showed courage to follow new ways in making the project work.

4.1.2 Development Tools

In the development of this project, “Tutor Express”, several modern web application tools were used.

React:

Figure 4.2: React Logo

ReactJS is a front-end JavaScript library for creating user interfaces or UI components[7] that is free and open-source [8] [9] . React can be used to create single-page or mobile applications as a foundation. React, on the other hand, is solely concerned with state management and rendering that information to the DOM, so constructing React apps frequently necessitates the usage of extra frameworks for routing and client-side functionality.

I have used React to do all the front end rendering for this project. It has been used to create home page, menu bar, forms and view of the tuition list.

Node.js and Express.js:

Figure 4.3: NodeJS Logo

Node.js is a cross-platform, open-source back-end JavaScript run time environment that uses the V8 engine to execute JavaScript code outside of a web browser [10]. Node.js allows developers to utilize JavaScript to create command-line tools and server-side scripting, which involves running scripts on the server before sending the page to the user's browser [11].

Node.js is designed to run on a dedicated HTTP server and to run one process at a time in a single thread. Applications written in Node.js are event-driven and execute asynchronously. Incoming requests are processed in a continuous event stack, and little requests are sent one after the other without waiting for responses.

NodeJS has been used to create back end functions, install dependencies, create modules for this project. We have used it to route back end servers and work the callback functions of creating tuition post, viewing them, editing and applying to them.

Figure 4.4: ExpressJS Logo

Express.js, or simply Express, is a back-end web application framework for Node.js that was distributed under the MIT License as free and open-source software [12] [13].

API request and server has been built using express and can handle request as required for this project.

Figure 4.5: Git Logo

Git: Git is a piece of software that allows you to track changes in any set of files [14] [15]. Git has been used to command the push of the code to the GitHub.

4.1.3 Other Non-Development Tools

The non-development tools used for "Tutor Express" which helped to make the communication easy, keep track of the workflow and repository are:

Figure 4.6: Discord Logo

Discord: Discord is a community-building platform that combines VoIP, instant messaging, and digital distribution [16] [17]. We use it in Techdojo company for quick communication with other developers. The senior developer would assign task and provide relevant resources related to that using Discord. Any queries from me would be addressed here and I was able to ask for help from the senior developers as well.

Figure 4.7: GitHub Logo

GitHub.com: It includes Git's distributed version control and source code management (SCM) features, as well as its own [18] [19]. The project was pushed after having final versions when needed so that it can be viewed and feedback can be given.

Figure 4.8: Google Docs Logo

Google Docs: Google Docs is an online word processor [20]. Google docs is used to keep a weekly progress timeline, plan for the week, update on the progress of the goal of the week and also addresses any issues to be fixed in the code.

Chapter 5

Body of the Project

5.1 Work Description

The web application features adding new tuition job post, viewing them, displaying them in the home page and applying to these tuition job posts. I have created a form component page where all the tuition requirement fields are submitted by the user. As it is submitted, create function works and routes through posting the submission in the back end and to MongoDB. These are then fetched back from the database and displayed to the homepage in tuition component. Individual tuition details can be viewed and routes to different page. There is another form found at the bottom that I have created, for whoever wishes to apply to that tuition job. If the fields are satisfied then the data is posted to the database in reference to that tuition job. The form submission data is shown to the user who is the owner of that particular tuition job post. This patches and connects these two types of user and makes the app as it was targeted to be.

5.2 System Analysis

"The process of evaluating a procedure or business to define its aim and purposes and build systems and procedures that will efficiently achieve them," according to the definition of systems analysis [21]. System Analysis is the process of gathering and interpreting data, finding faults, and breaking down a system into its constituent parts [22]. A system analysis is carried out to investigate a system in order to determine its goals. It is a problem-solving strategy that enhances the system and guarantees that all of the system's components work together to achieve their goals. The chapter include parts of system analysis methods to understand the project better.

5.2.1 Six Element Analysis

System Roles						
Process	Human	Non Computing Hardware	Computing Hardware	Software	Database	Comm. and Network
Create tuition job post	User, Parents	Student Information	Desktop, Laptop	Web browser	MongoDB	Internet, WiFi, Cellular Data
View particular tuition job post	User, Parents, Tutors	N/A	Desktop, Laptop	Web browser	MongoDB	Internet, WiFi, Cellular Data
View all tuition job list	User, Parents, Tutors	N/A	Desktop, Laptop	Web browser	MongoDB	Internet, WiFi, Cellular Data
Apply to tuition job post	User, Tutors	Tutor Information	Desktop, Laptop	Web browser	MongoDB	Internet, WiFi, Cellular Data
View applied tutor info to particular tuition job post	User, Parents	N/A	Desktop, Laptop	Web browser	MongoDB	Internet, WiFi, Cellular Data

Table 5.1: Six Element Analysis

5.2.2 Feasibility Analysis

A feasibility study determines if a proposed project or system is feasible. A feasibility study tries to objectively and logically identify the strengths and weaknesses of a current business or new enterprise, as well as the possibilities and risks that exist in the natural environment, the resources needed to carry out the project, and the likelihood of success [23] [24] [25]. In its most basic form, the two factors for determining feasibility are the required cost and the value to be obtained.

- Technical Feasibility: This evaluation is based on an outline design of system requirements to see if the company has the technical skills to complete the project. It is centered on getting a grasp of the organization's current technological resources and their application to the anticipated needs of the proposed system. It's an assessment of the hardware and software to see how well it meets the requirements of the planned system. "Tutor Express" is built using React, Express, MongoDB and NodeJS being a full stack MERN application. Every developer working in this project, including myself has at least one skill to adapt and work in the project. Thus making the project technically feasible.

- Operational Feasibility: Operational feasibility is a metric for how well a proposed system solves problems and exploits possibilities discovered during scope definition, as well as how well it meets the criteria identified during the requirements analysis phase of system development. In terms of development timetable, delivery date, corporate culture, and existing business procedures, the operational feasibility study looks at how well the proposed development project fits into the existing business environment and objectives. Desired operational results must be communicated during design and development to ensure success. "Tutor Express" is a skeleton app and is thought made to be simple and usable by all. It has straight logic that are understandable and to enlighten that, the buttons instruct accurately to the feature. Also each page is kept for one purpose so that users are never confused. Thus making the web app operational feasible.

- Economic Feasibility: Economic feasibility is a type of cost-benefit analysis of the project under consideration that determines if it can be implemented [26]. This phrase refers to the evaluation and study of a project's ability to aid decision-making by identifying its strengths, flaws, opportunities and risks, resources required to implement the project, and chances of success. "Tutor Express" is not currently being deployed, but if considering that it only needs yearly cloud server subscription after it is completely done and deployed. This makes it affordable in maintenance and also economically feasible.

5.2.3 Problem Solution Analysis

Problems: During our undergraduate years, we all seek to have a way of income to ease our way of life. Tuition job is popular in this case, as well as tough to find the most suitable one. Youngsters usually look for tuition jobs using social media, or in-person connection. This leaves some with a good luck opportunity but many with no luck. Many of them look for such jobs in order to pay their own tuition/semester fees. Many of the young interested people do not get beneficial from the source found in social media. Especially, during the pandemic it has become more difficult. Therefore, dedicated web platforms are needed to address such problems.

- Tough to search: Hunting down a tuition job becomes stressful for many and may take months.
- Loss of time: In case of not finding one suitable tuition job, a person losses valuable time.
- Redundant finds: Sometimes people are reached with offers that is not suitable for them.
- Lack of options: Offers coming from person or through any media is just a handful that leaves anyone with minimum option to choose from.
- No direct path for two end user: Usually there is a third media in between any tuition job, whether in-person or through social media ads.

Solutions: The solutions provided by "Tutor Express" is analysed and mentioned below.

- Solution for 'Tough to search': The web app offers tuition jobs coming from anyone and everyone throughout the city, and reducing the tutor users search stress.
- Solution for 'Redundant Finds': Unsuitable offers reaching them can be ignored, as the web app offers them a platform to look for better tuition jobs.
- Solution to 'Lack of options': The users are offered here with a huge tuition job list, as tuition jobs are always posted by the users of the web app directly.
- Solution for 'No direct path for two end user': The tutors and parents will be offered with trust as they both know that they are posting the tuition job post and it is being applied by other users also. This ensures that both party is trusting the information from both sides and being as genuine as possible.

5.3 System Design

The process of defining the architecture, product design, modules, interfaces, and data for a system in order to meet specific criteria is known as systems design [27]. The application of systems theory to product development is known as systems design. The disciplines of systems analysis, systems architecture, and systems engineering have some overlap. The process of identifying and developing systems to meet the user's specific requirements is known as systems design. Software architecture is an abstract depiction of a program's general structure. This section includes various design level illustrations to help you understand the system and data flow better.

5.3.1 Rich Picture

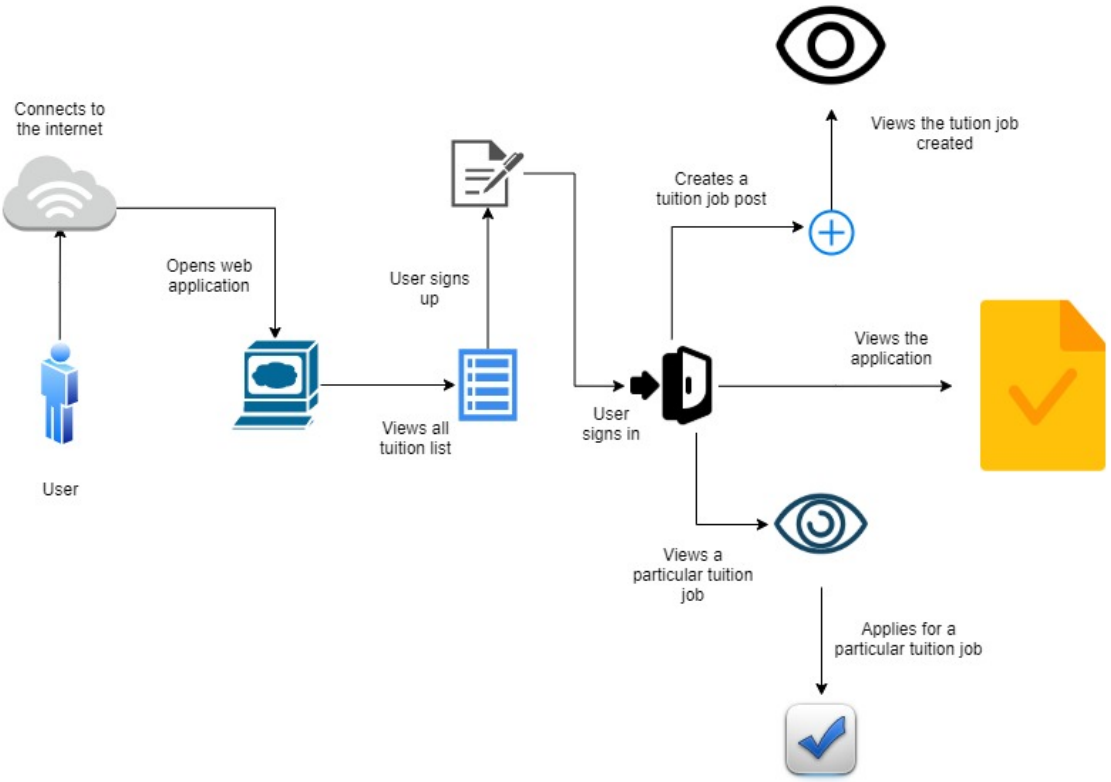


Figure 5.1: Rich Picture for "Tutor Express"

The rich picture show how the web application will function. Firstly, user gets a view of the list of the tuition jobs already posted. User signs up and signs in to proceed forward. The user can either post a tuition job or apply to a tuition job by switching profile. If a user creates a tuition job, they have to fill a form with required fields. As the form is submitted, the user views the details and then posts it to be open to other users to be viewed and apply. If the user wants to apply to a tuition job, they can switch back to regular profile and browse through all and any tuition. When selected, the user then needs to fill a form with required information to apply for the tuition job. If any user is an owner of a tuition job post, they can also view who applied to the job, along with the information submitted. The user can use these features again or sign out.

5.3.2 UML Diagrams

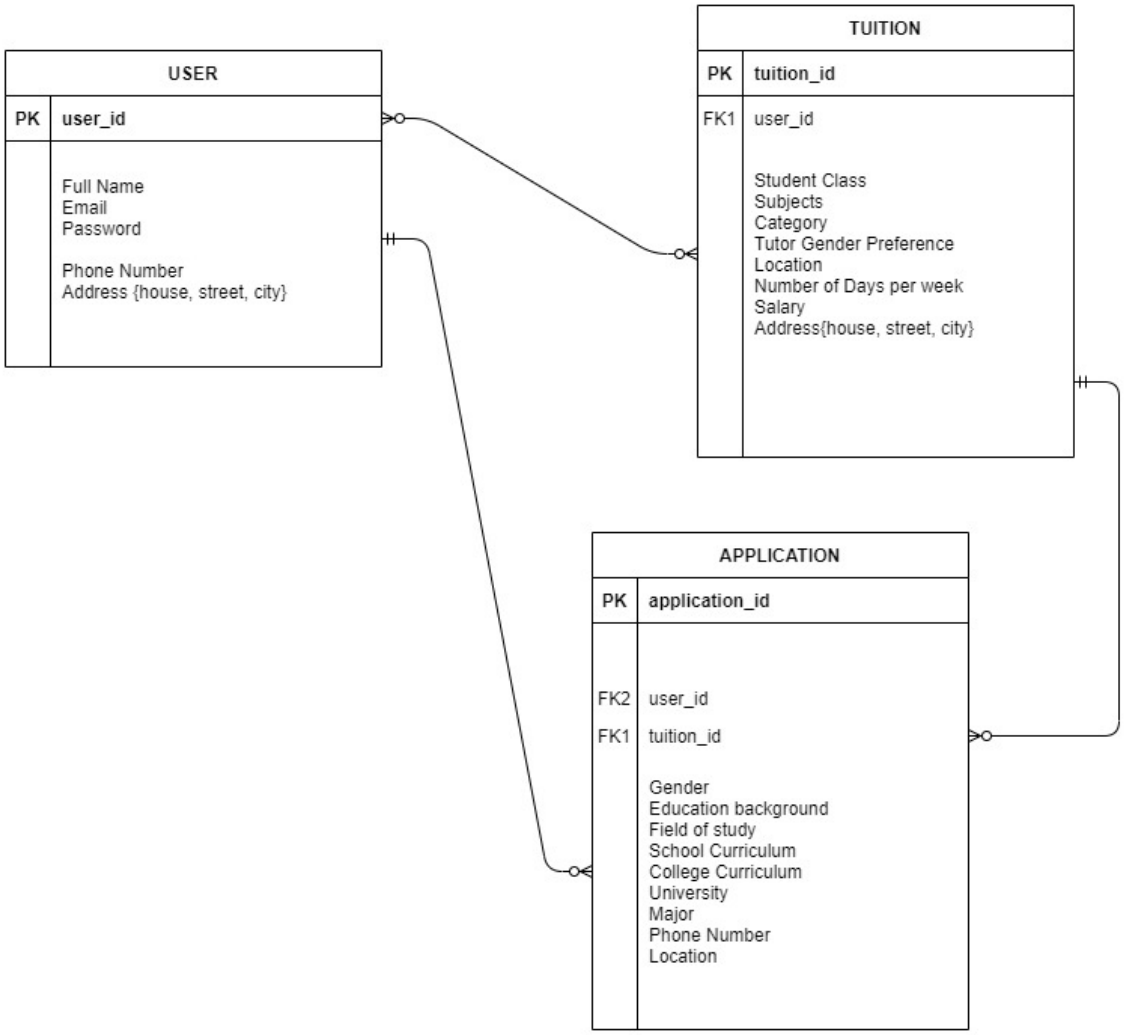


Figure 5.2: Entity Relationship Diagram for "Tutor Express"

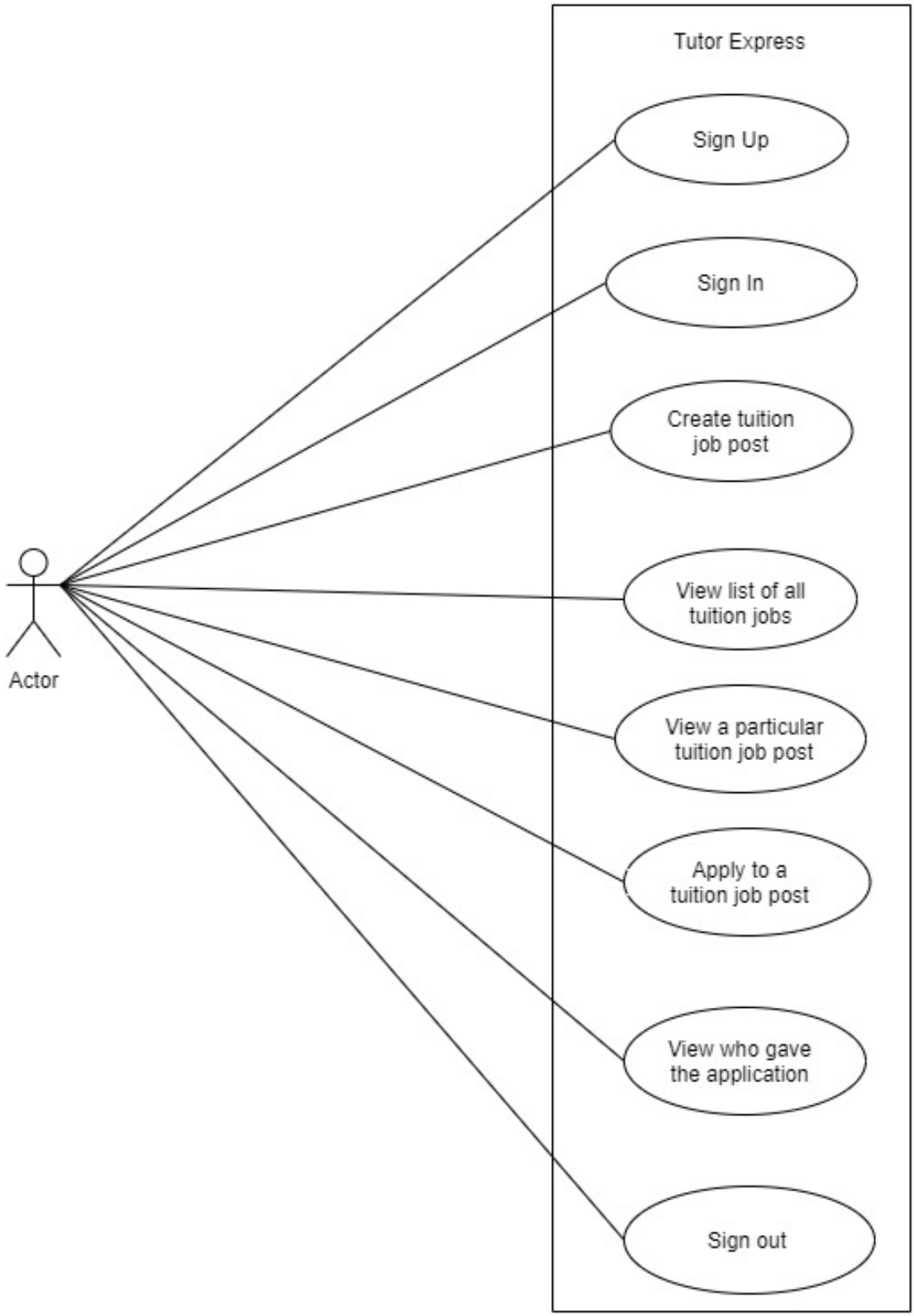


Figure 5.3: Use Case Diagram for "Tutor Express"

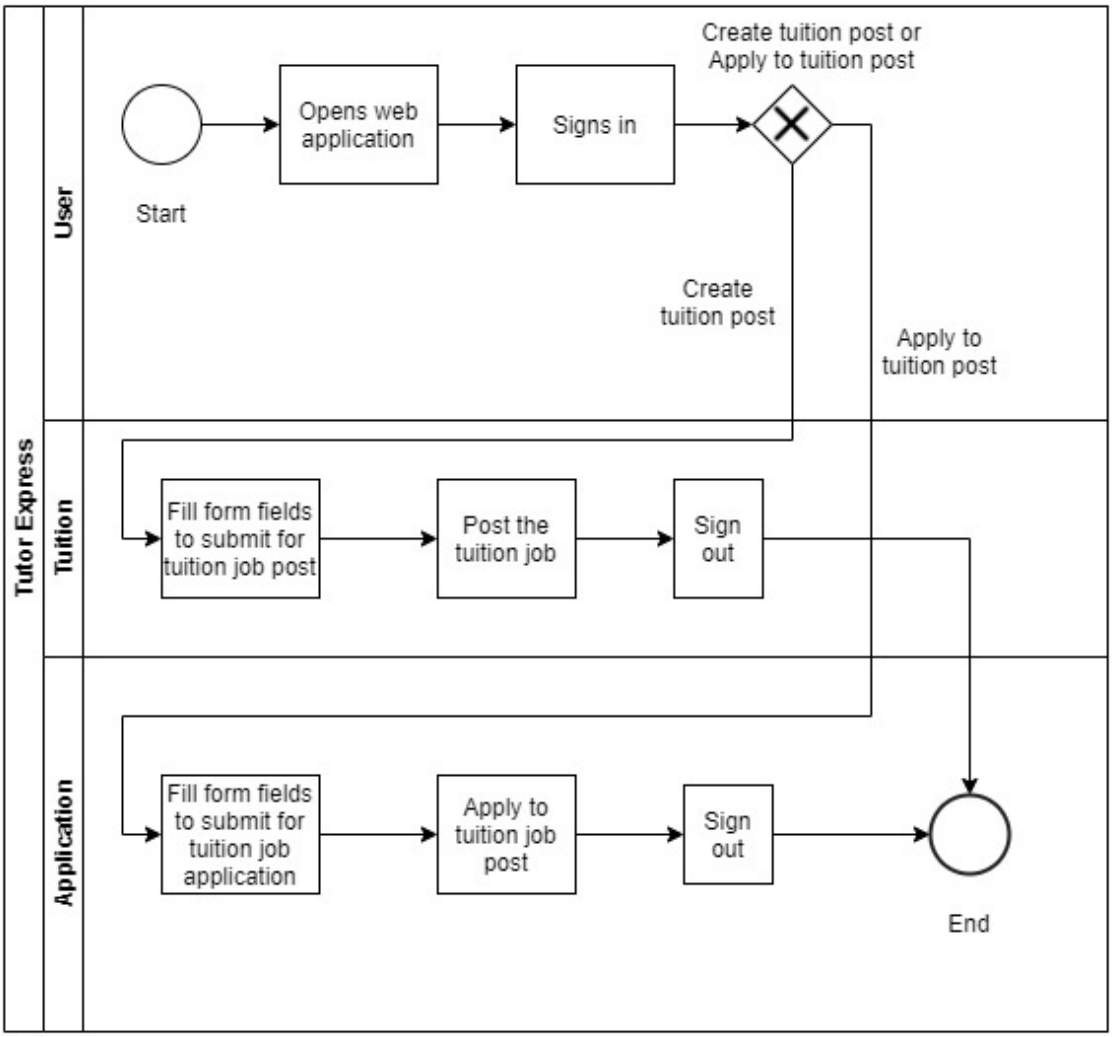


Figure 5.4: Business Process Model and Notation (BPMN 2.0) for "Tutor Express"

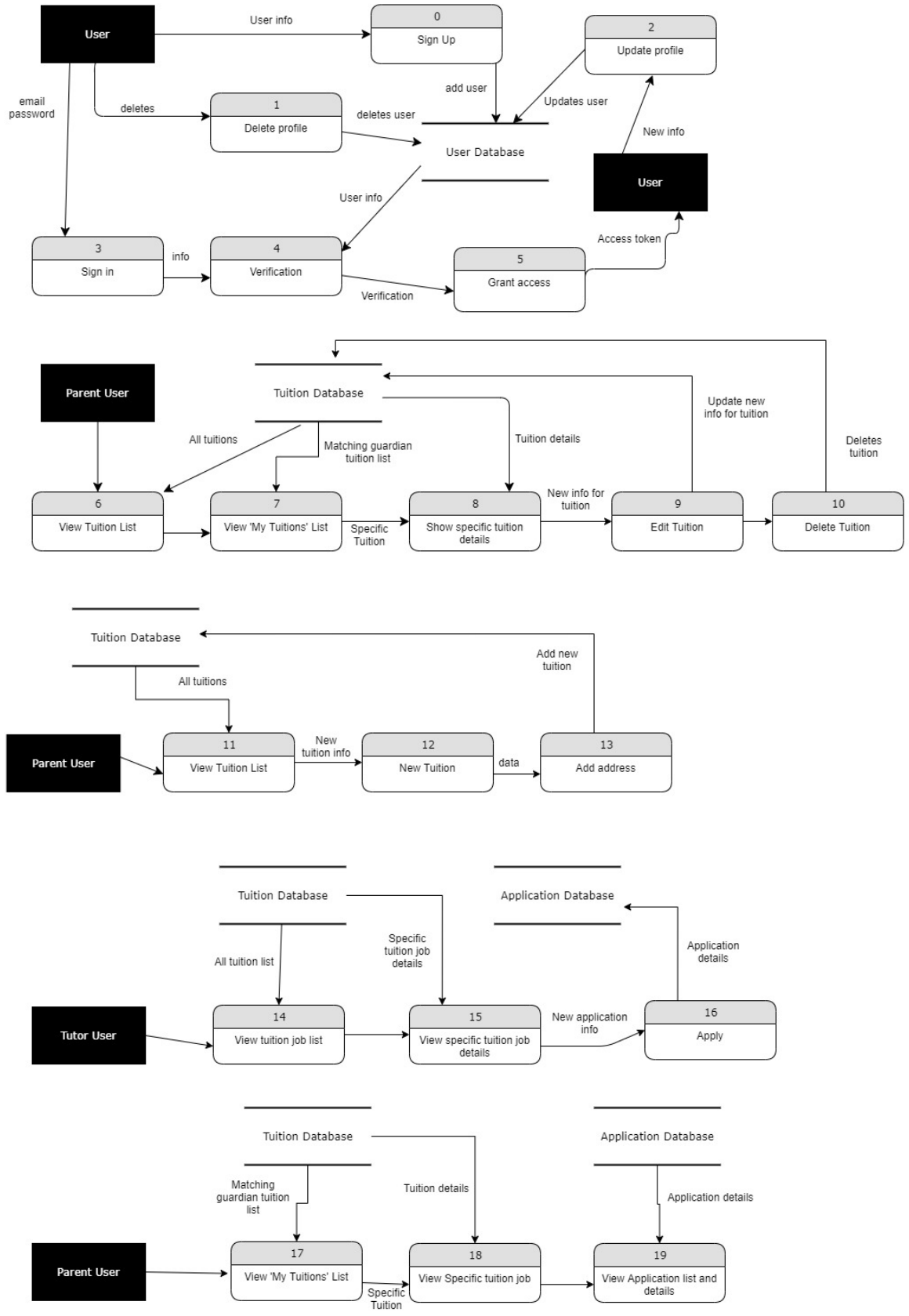


Figure 5.5: Data Flow Diagram for “Tutor Express”

5.3.3 Functional and Non-Functional Requirements

Software Requirements: The software requirements are a list of the target system's features and functions [28]. Users' expectations of the software product are expressed in requirements. From the client's perspective, the requirements can be evident or buried, known or unknown, expected or unexpected.

Functional Requirements: This category includes requirements that are relevant to the functionality of software. They specify the software system's internal and external functions and functionality. The following tables show functional requirements that were decided requirements earlier. It also includes input and output for the process.

Function: Compatible with all browser		
Input	Process	Output
N/A	Web application must be programmed in a common environment	Application is accessible from all web browser
Precondition: User must have a working computer with internet connection		
Post condition: All users are able to access the application		

Table 5.2: Functional Requirement 1: Compatibility

Function: Navigate from one page to another		
Input	Process	Output
Select page to navigate to	Set up browser navigation	Navigate to desired page
Precondition: The web application is browsed in or opened		
Post condition: Page navigated to is displayed and further navigation is open		

Table 5.3: Functional Requirement 2: Navigation

Function: Create an account		
Input	Process	Output
Fill form with required fields	Call API to server to create a new account from provided data	Prompt to navigate to sign in page
Precondition: User must be in the sign up page and connected to internet		
Post condition: User is prompt with a success message and will be navigated to sign in page		

Table 5.4: Functional Requirement 3: Sign Up

Function: Sign in		
Input	Process	Output
Fill form with required fields	Call API to server to authorize from provided data	Navigate to home page
Precondition: User must be in the sign in page and connected to internet		
Post condition: User is navigated to home page		

Table 5.5: Functional Requirement 4: Sign In

Function: Submit data for a tuition job post		
Input	Process	Output
Fill form with required fields	Call API to server to submit values from provided data	Navigate to tuition overview page
Precondition: User must be signed in and connected to internet		
Post condition: User is navigated to tuition overview page		

Table 5.6: Functional Requirement 5: Submit data for tuition

Function: Create a tuition job post		
Input	Process	Output
Fill required address form and press button	Call API to server to create post from provided data	Navigate to home page
Precondition: User must be signed in, tuition information form must be filled		
Post condition: User is navigated to home page with successful post		

Table 5.7: Functional Requirement 6: Create a tuition job post

Function: View tuition job posts		
Input	Process	Output
Navigate to home page	Call API to server to fetch all post from database	View list of all tuition posts
Precondition: User must be signed in and connected to internet		
Post condition: User is navigated to home page		

Table 5.8: Functional Requirement 7: View all tuition job posts

Function: View a tuition job post		
Input	Process	Output
Navigate to home page and select any tuition job post	Call API to server to fetch the selected post from database	View data of that tuition post
Precondition: User must be signed in and connected to internet		
Post condition: User is navigated to tuition overview		

Table 5.9: Functional Requirement 8: View a tuition job post

Function: Switch user		
Input	Process	Output
Navigate to my profile page and click edit to switch	Call API to server to update the user	Navigate back to my profile page
Precondition: User must be signed in and connected to internet		
Post condition: User is switched to normal user(tutor) and navigated to my profile page		

Table 5.10: Functional Requirement 9: Switch User

Function: Apply to a tuition job		
Input	Process	Output
Navigate to tuition job post, submit required fields and click apply	Call API to server to create application for that job post	Navigate back to home page
Precondition: User must be signed in, switched to normal profile and connected to internet		
Post condition: User is navigated to home page with successful application		

Table 5.11: Functional Requirement 10: Apply to a tuition job post

Function: View data of the application		
Input	Process	Output
Navigate to particular tuition job post	Call API to server to fetch data for that job post	Navigate back to that tuition job post and view the application data
Precondition: User must be signed in, switched to parents profile and connected to internet		
Post condition: User is navigated to course overview		

Table 5.12: Functional Requirement 11: View application to a tuition job post

Function: Sign out		
Input	Process	Output
Navigate to any page	Call API to server to get data and sign out	Navigate to home page
Precondition: User must be signed in and connected to internet		
Post condition: User is signed out navigated to home page		

Table 5.13: Functional Requirement 12: Sign out

Non-Functional Requirements: This category includes requirements that aren't connected to the software's functionality. They are software's implicit or assumed qualities, which users take for granted [29].

- Performance: Under a given workload, performance refers to how quickly a software system or a specific portion of it responds to specific user actions. Given the current number of users, this measure often explains how long a user must wait before the intended operation (the page renders, a transaction is performed, etc.) occurs. "Tutor Express" will offer a smooth experience as there is no input lag and the styling is kept light to adapt to most of the hardware specifications.

- Security and Control: This non-functional requirement ensures that all data stored within the system or its components is safe from malware assaults and unauthorized access. On both the server and client sides, all information is secure. Only the program administrators and developers have access to the application's core code, which allows them to directly change any type of data. Back end technology has been utilized in this project, with node.js and express.js providing several layers of protection to ensure that the system's security requirements are met. Control requirements define the system's operating environment, as well as the type and level of security that must be supplied. Access to the system or information must be restricted in accordance with privacy regulations.

- Efficiency: The ability of a system to produce outputs with low waste is referred to as efficiency. We attempted to minimize redundant steps in the processes and make efficient use of resources. We accomplished efficiency by keeping our code non-repetitive by utilizing reusable code and components.

- Maintainability: Maintainability is the amount of time it takes to fix, change, or adapt a solution or one of its components to improve performance or other attributes, or to adapt to a changing environment. For the style and feel of the application, there is just one standard User Interface. Without affecting existing modules, the application can be expanded to handle a large number of additional modules. The application is designed in such a way that both the server and client sides are simple to maintain.

5.4 Product Features

5.4.1 Input

Function: Compatible with all browser	
Input	Process
N/A	Web application must be programmed in a common environment
Precondition: User must have a working computer with internet connection	

Table 5.14: Input 1: Compatibility

Function: Navigate from one page to another	
Input	Process
Select page to navigate to	Set up browser navigation
Precondition: The web application is browsed in or opened	

Table 5.15: Input 2: Navigation

Function: Create an account	
Input	Process
Fill form with required fields	Call API to server to create a new account from provided data
Precondition: User must be in the sign up page and connected to internet	

Table 5.16: Input 3: Sign Up

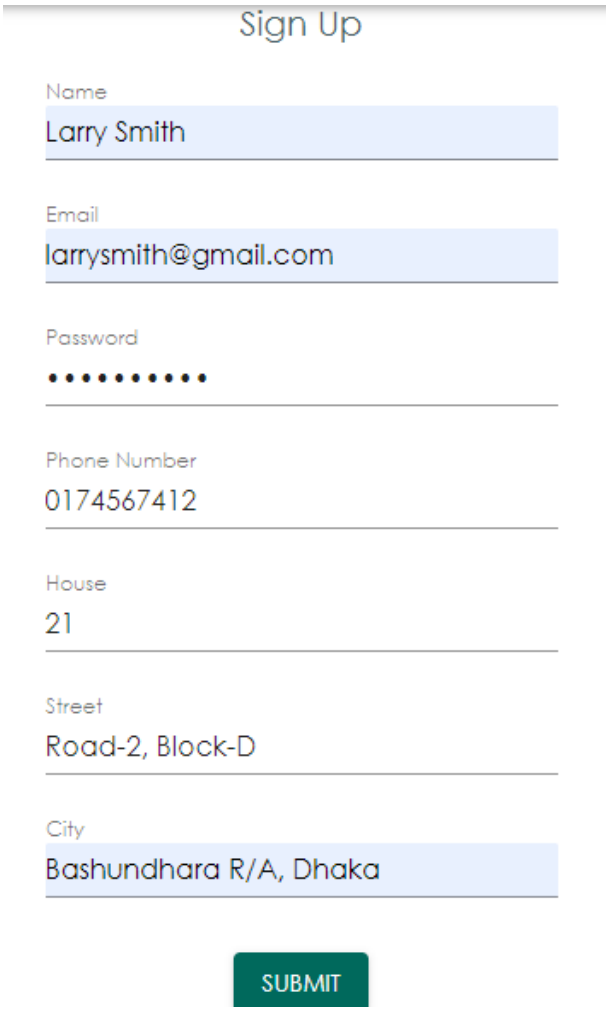


Figure 5.6: Sign up Input

Function: Sign in	
Input	Process
Fill form with required fields	Call API to server to authorize from provided data
Precondition: User must be in the sign in page and connected to internet	

Table 5.17: Input 4: Sign In

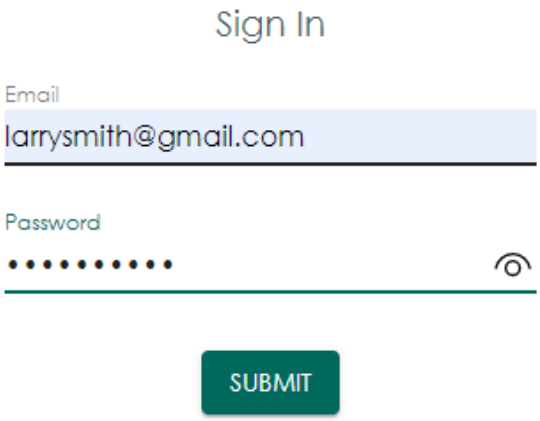


Figure 5.7: Sign in Input

Function: Submit data for a tuition job post	
Input	Process
Fill form with required fields	Call API to server to submit values from provided data
Precondition: User must be signed in and connected to internet	

Table 5.18: Input 5: Submit data for tuition

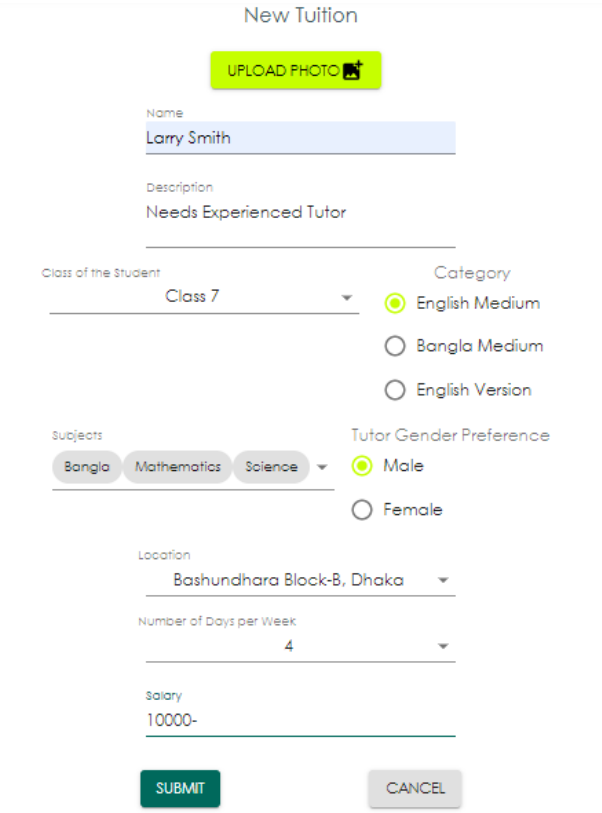


Figure 5.8: Tuition job post Input

Function: Create a tuition job post	
Input	Process
Fill required address form and press button	Call API to server to create post from provided data
Precondition: User must be signed in, tuition information form must be filled	

Table 5.19: Input 6: Create a tuition job post

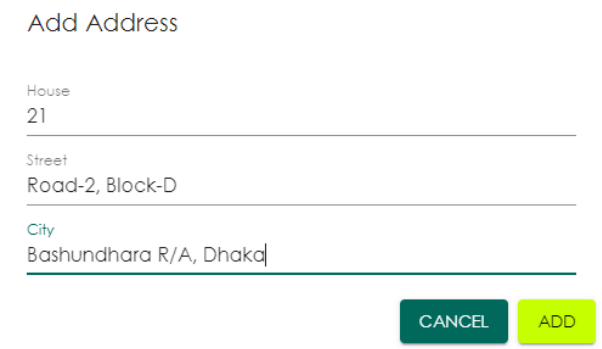


Figure 5.9: Address Input

Function: View tuition job posts	
Input	Process
Navigate to home page	Call API to server to fetch all post from database
Precondition: User must be signed in and connected to internet	

Table 5.20: Input 7: View all tuition job posts

Function: View a tuition job post	
Input	Process
Navigate to home page and select any tuition job post	Call API to server to fetch the selected post from database
Precondition: User must be signed in and connected to internet	

Table 5.21: Input 8: View a tuition job post

Function: Switch user	
Input	Process
Navigate to my profile page and click edit to switch	Call API to server to update the user
Precondition: User must be signed in and connected to internet	

Table 5.22: Input 9: Switch User

Function: Apply to a tuition job	
Input	Process
Navigate to tuition job post, submit required fields and click apply	Call API to server to create application for that job post
Precondition: User must be signed in, switched to normal profile and connected to internet	

Table 5.23: Input 10: Apply to a tuition job post

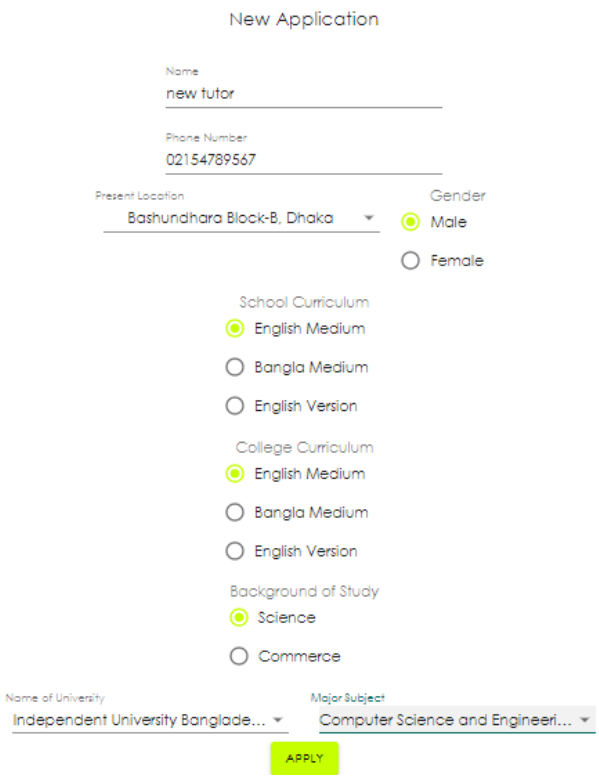


Figure 5.10: Application Input

Function: View data of the application	
Input	Process
Navigate to particular tuition job post	Call API to server to fetch data for that job post
Precondition: User must be signed in, switched to parents profile and connected to internet	

Table 5.24: Input 11: View application to a tuition job post

Function: Sign out	
Input	Process
Navigate to any page	Call API to server to get data and sign out
Precondition: User must be signed in and connected to internet	

Table 5.25: Input 12: Sign out

5.4.2 Output

Function: Compatible with all browser	
Process	Output
Web application must be programmed in a common environment	Application is accessible from all web browser
Post condition: All users are able to access the application	

Table 5.26: Output 1: Compatibility

Function: Navigate from one page to another	
Process	Output
Set up browser navigation	Navigate to desired page
Post condition: Page navigated to is displayed and further navigation is open	

Table 5.27: Output 2: Navigation

Function: Create an account	
Process	Output
Call API to server to create a new account from provided data	Prompt to navigate to sign in page
Post condition: User is prompt with a success message and will be navigated to sign in page	

Table 5.28: Output 3: Sign Up

Function: Sign in	
Process	Output
Call API to server to authorize from provided data	Navigate to home page
Precondition: User must be in the sign in page and connected to internet	

Table 5.29: Output 4: Sign In

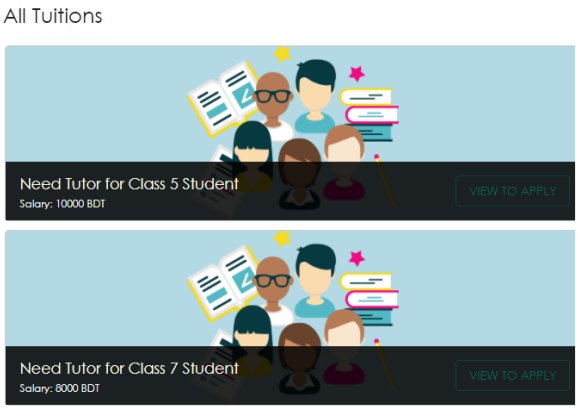


Figure 5.11: Sign in Output

Function: Submit data for a tuition job post	
Process	Output
Call API to server to submit values from provided data	Navigate to tuition overview page
Post condition: User is navigated to tuition overview page	

Table 5.30: Output 5: Submit data for tuition

Function: Create a tuition job post	
Process	Output
Call API to server to create post from provided data	Navigate to home page
Post condition: User is navigated to home page with successful post	

Table 5.31: Output 6: Create a tuition job post

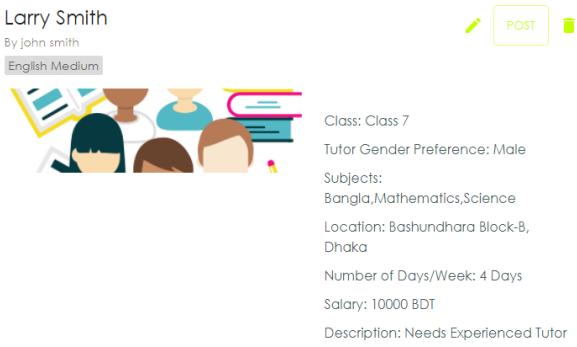


Figure 5.12: Tuition form Output

Function: View tuition job posts	
Process	Output
Call API to server to fetch all post from database	View list of all tuition posts
Post condition: User is navigated to home page	

Table 5.32: Output 7: View all tuition job posts

Function: View a tuition job post	
Process	Output
Call API to server to fetch the selected post from database	View data of that tuition post
Post condition: User is navigated to tuition overview	

Table 5.33: Output 8: View a tuition job post

Function: Switch user	
Process	Output
Call API to server to update the user	Navigate back to my profile page
Post condition: User is switched to normal user(tutor) and navigated to my profile page	

Table 5.34: Output 9: Switch User

Function: Apply to a tuition job	
Process	Output
Call API to server to create application for that job post	Navigate back to home page
Post condition: User is navigated to home page with successful application	

Table 5.35: Output 10: Apply to a tuition job post

Function: View data of the application	
Process	Output
Call API to server to fetch data for that job post	Navigate back to that tuition job post and view the application data
Post condition: User is navigated to course overview	

Table 5.36: Output 11: View application to a tuition job post

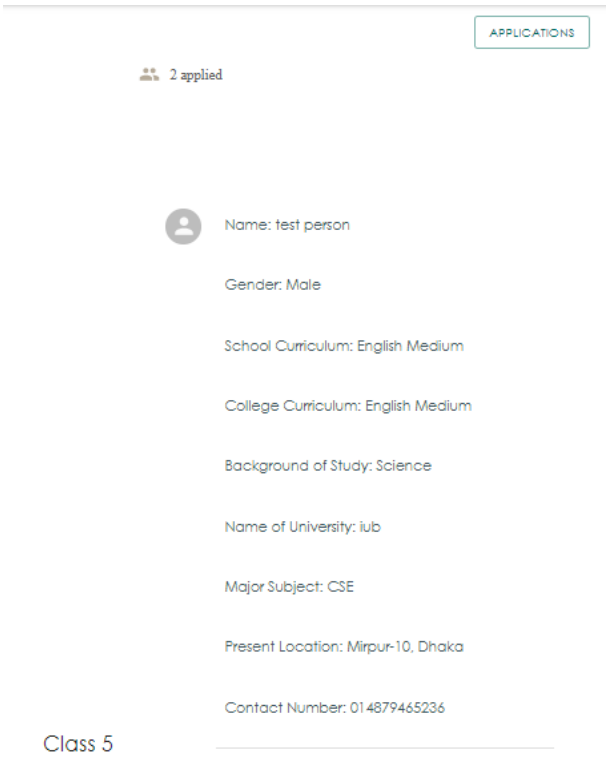


Figure 5.13: Application Output

Function: Sign out	
Process	Output
Call API to server to get data and sign out	Navigate to home page
Post condition: User is signed out navigated to home page	

Table 5.37: Output 12: Sign out

5.4.3 Architecture

The underlying structures of a software system, as well as the discipline of building such structures and systems, are referred to as software architecture. Each structure is made up of software elements, their relationships, and the qualities of both the elements and the relationships. Specific structural options from software design possibilities are included in software architecture choices. For "Tutor Express", a three-tier client server architecture is used.

A client-server architecture is a computer network architecture in which a centralized server serves multiple clients. Client computers provide an interface that enables a computer user to request server services using the application. The results returned by the server is viewed again through the application. Client queries are sent to database servers, which wait for them to arrive before responding.

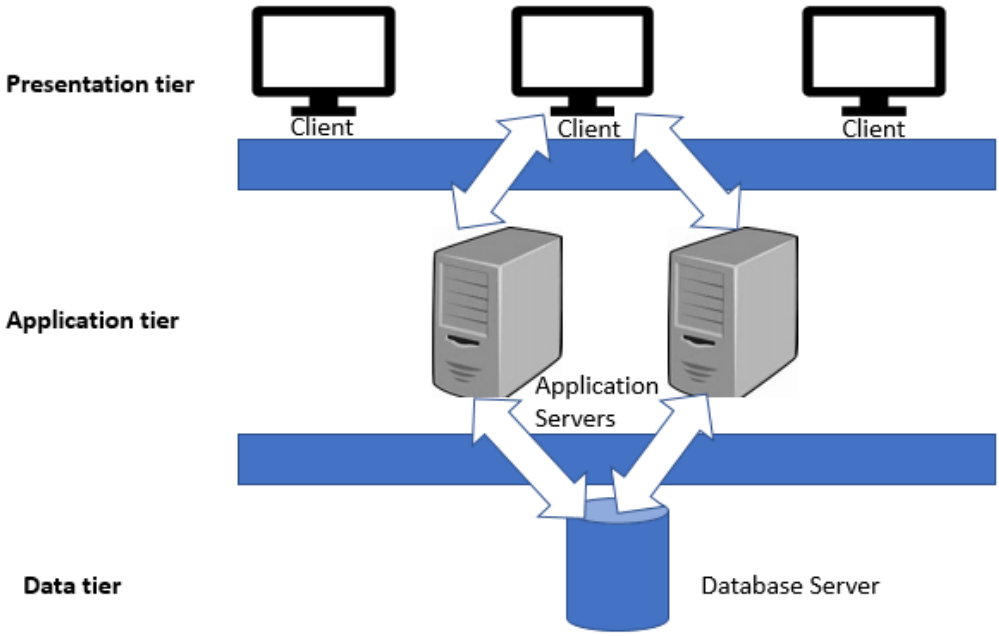


Figure 5.14: Client Server Architecture

In "Tutor Express", every authorized user will have access to the application server. When a user creates a tuition job post, it sends a request to the application server and saves it in database server. When the users view all or any tuition job post, it sends a request to the server to display them by fetching it from database. After an application is made by a use, the application server receives a request and stores the data given in the database server. Having a centralized control benefits this, and also this architecture allows server to be flexible when user capacity increases.

Chapter 6

Results & Analysis

The front end of the web application was built with React, while the back end was built with Node.js and Express.js. Because it is speedier and takes up less space, React is used. Because Node.js and Express.js are lightweight, they offer routing, sessions, and caching, and, most crucially, they interact with the MongoDB database.

This chapter includes screenshots of the web application so that we can see how it appears in action.

- Home Page: The home page contains the list of all the tuition jobs with minimum details. The menu bar is present in all pages. Here, the user is presented with sign up and sign in. There is also a home button that leads to this home page.

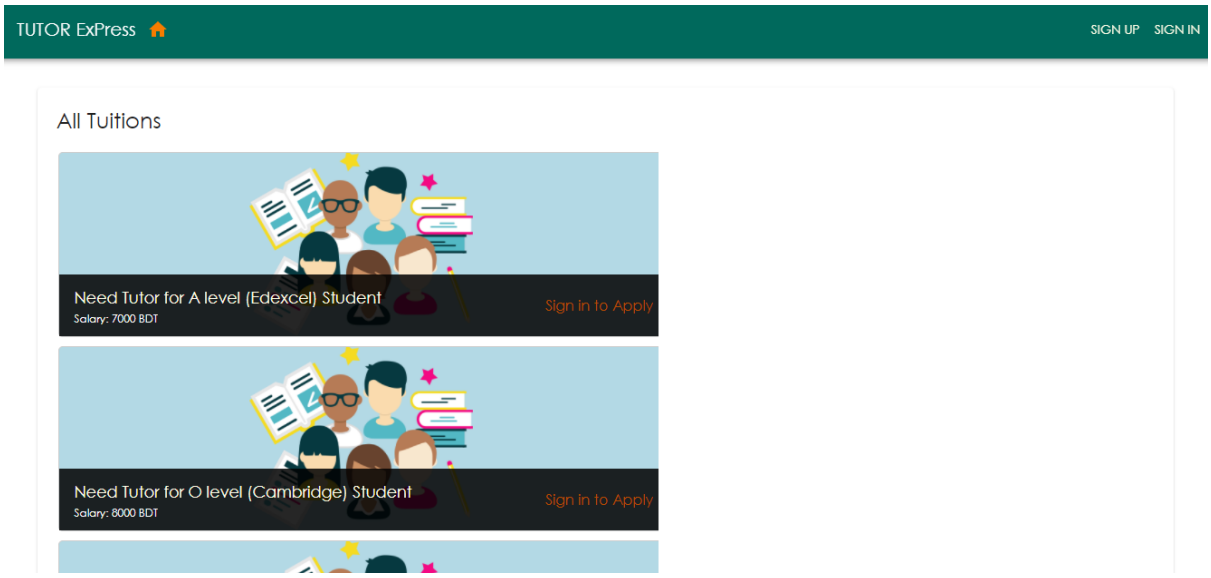


Figure 6.1: Screen view of Home Page

- Sign Up Page: In this page, the user will be displayed a form with fields that are required to fill in order to sign up. After successful sign up, the user can go to sign in page.

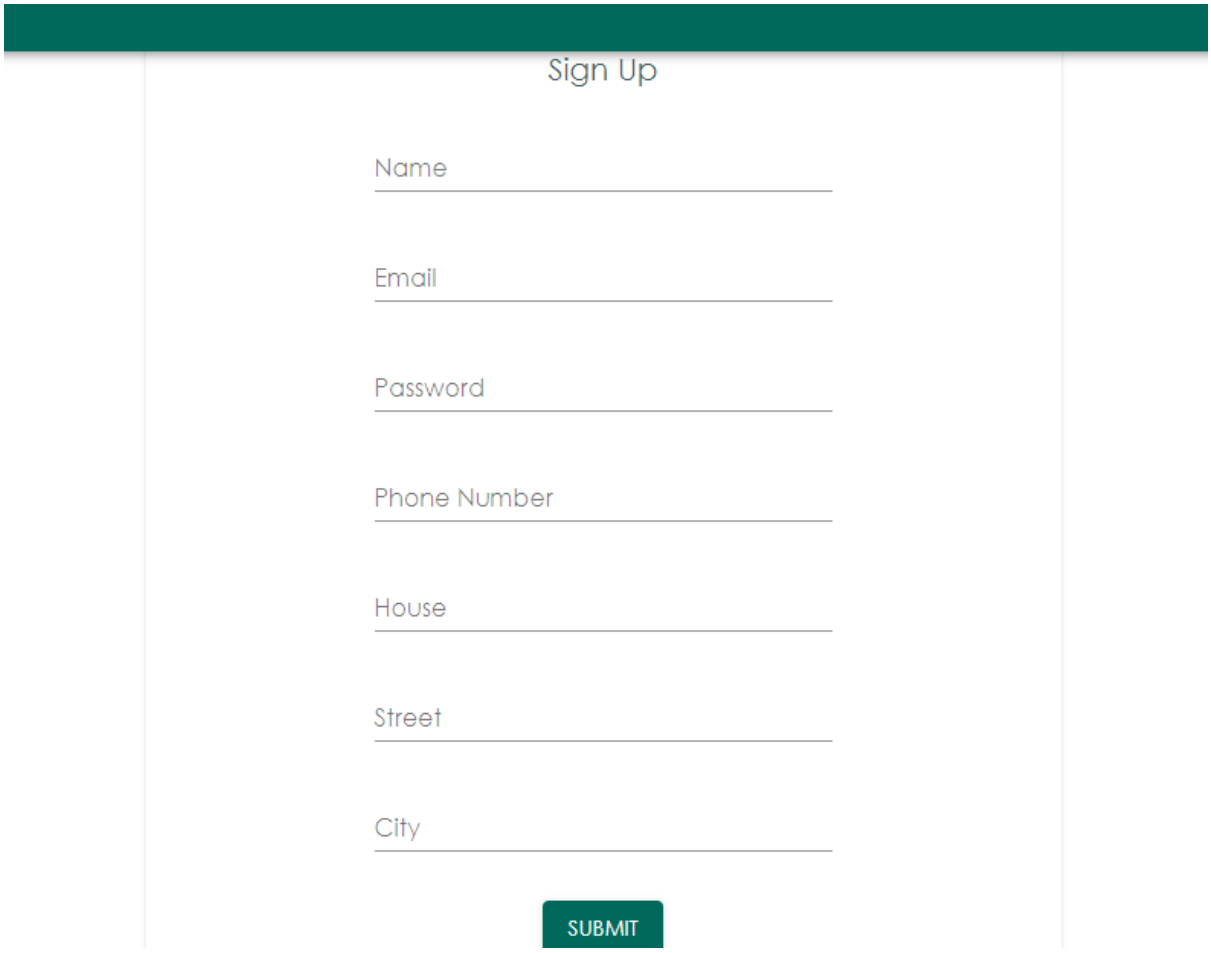


Figure 6.2: Screen view of Sign Up page

- Sign In Page: The user can sign in to use further features of the web application by filling in with the username and password created during sign up.

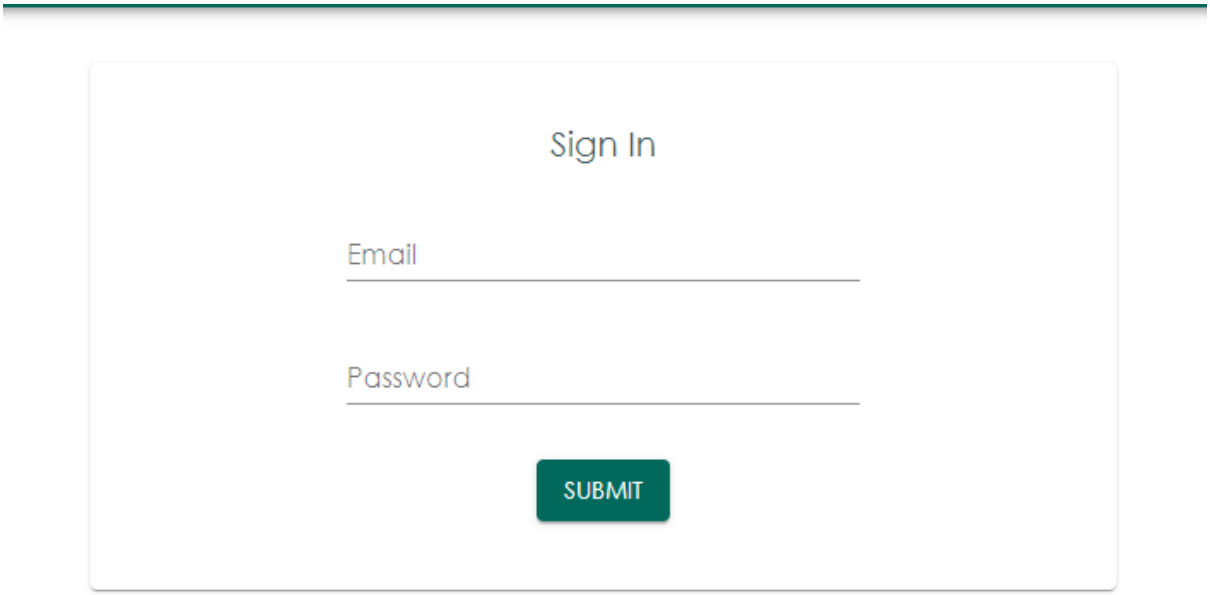


Figure 6.3: Screen view of Sign In page

- User profile page: After signing in, the user now has a 'My profile' section in the menu bar. This button links to the user profile page, where the user can wish to edit or delete their profile.

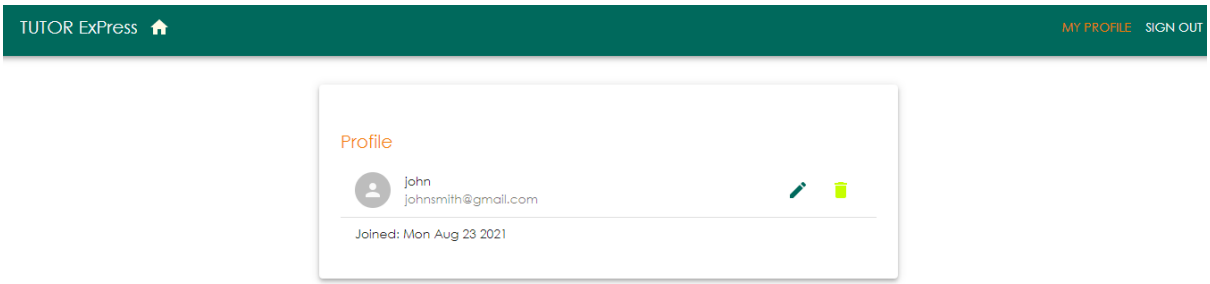


Figure 6.4: Screen view of My Profile page

- Edit profile page: The user can edit their profile or can switch to a parent profile if they wish to post a tuition job for others user. Only switching, in this edit page will enable the user to do so.

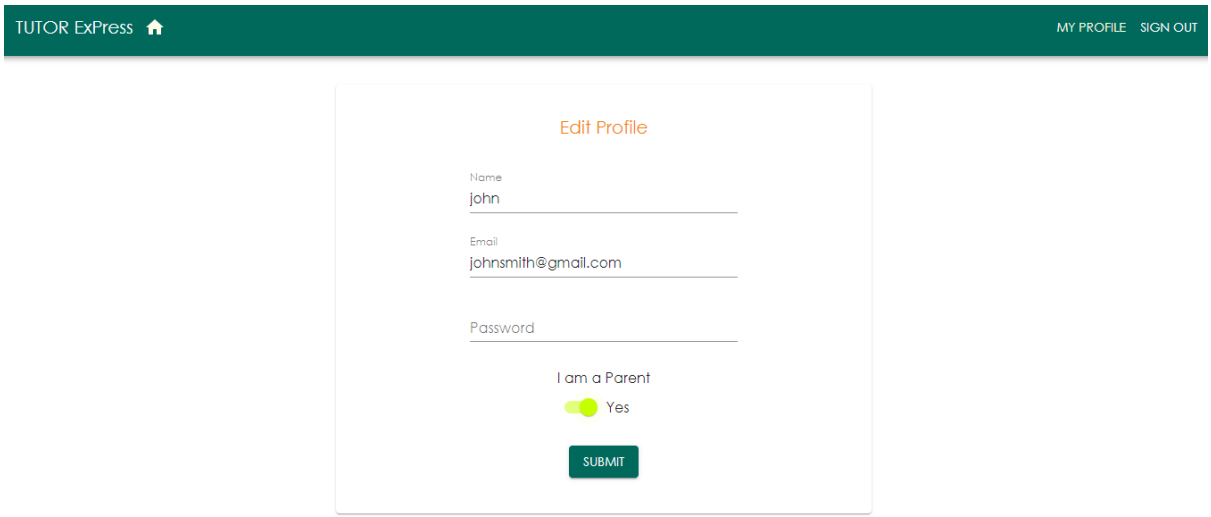


Figure 6.5: Screen view of Edit Profile page

- List of tuition page: When the user switches to parent profile, the menu bar has a new button called "Tutoring". This button links to "My Tuitions" page, where user can view all the tuition jobs they have created. They are also given a button to add more new tuition jobs if they wish to.

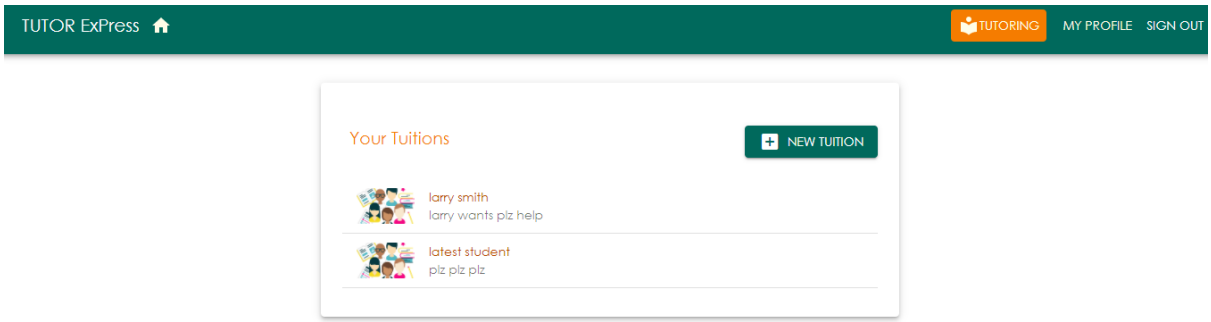


Figure 6.6: Screen view of My Tuitions page

- New Tuition page: The user is given a form to filled when they press the button to add new tuition. The form has various fields to be filled and then submitted.

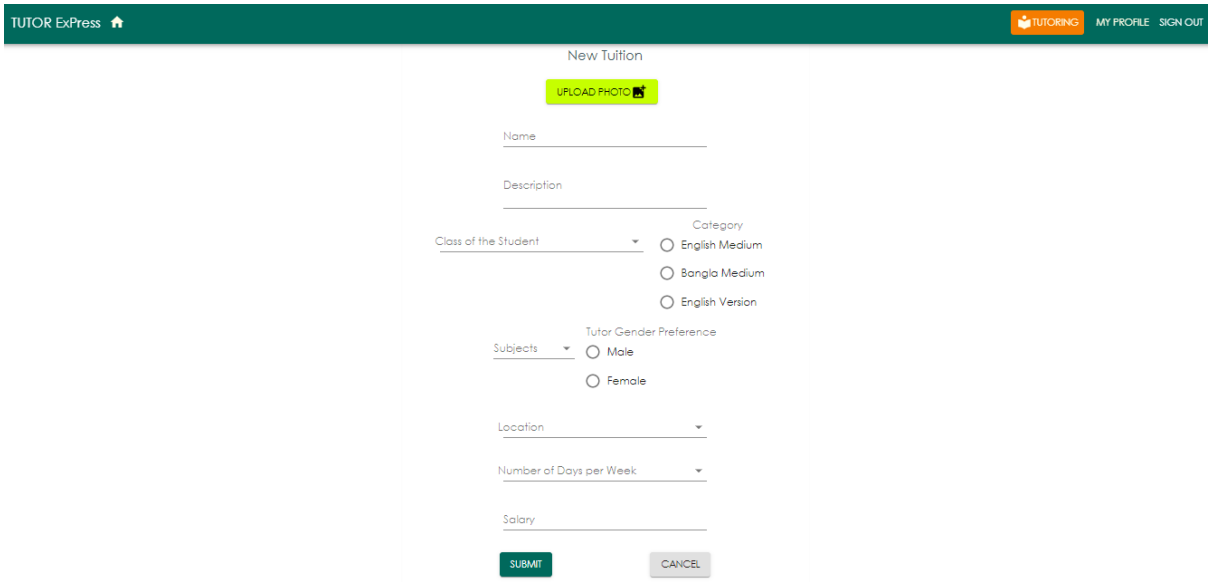


Figure 6.7: Screen view of New Tuition page

- Tuition details page: This page gives the user to overview what they have submitted for the new tuition job before finally posting. The user can edit if they want to change anything using edit button and also delete the tuition job should they wish to. In order to post successfully, the user must add address for the tuition job. This can be done from the button below this page.

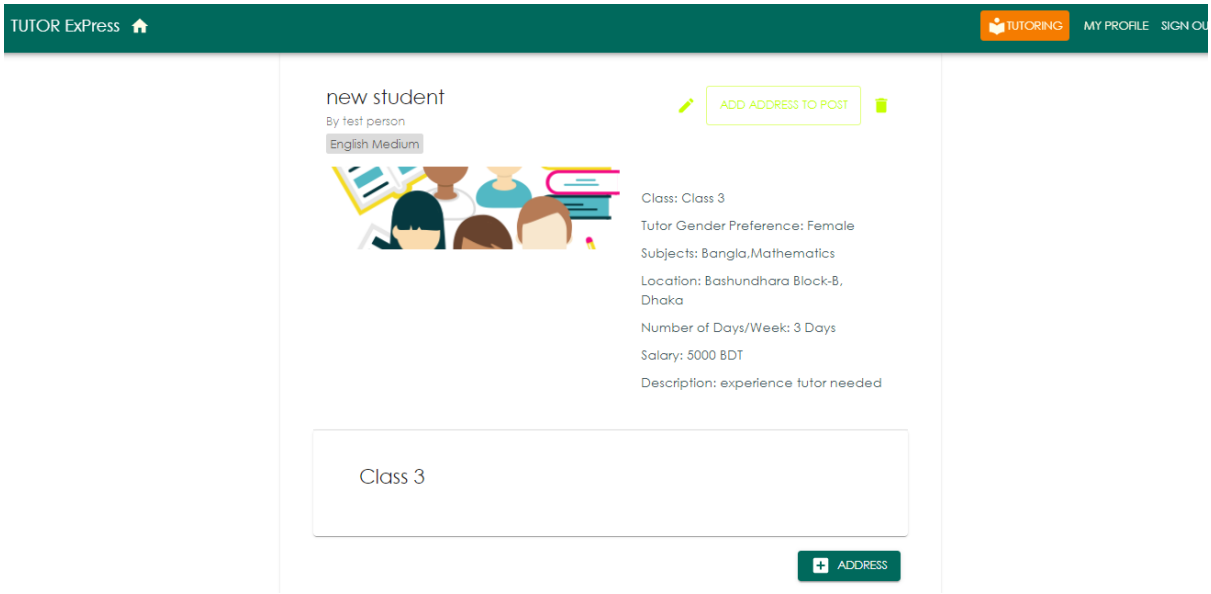


Figure 6.8: Screen view of Tuition details page

- Add Address dialog box: The user can see a dialog box when they click on 'Add Address' button to input the address for the tuition job. The address is never revealed to any user.

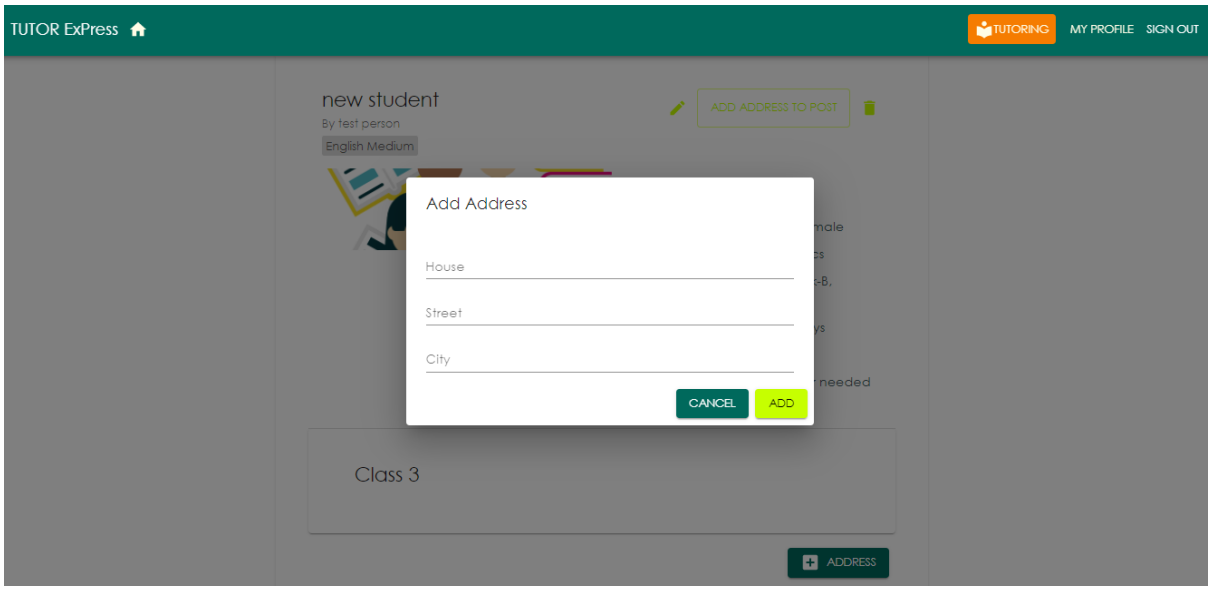


Figure 6.9: Screen view of Add Address dialog box

- Edit Tuition page: In this page, the user is given a filled form of previous input to edit any field if they wish to and save for the tuition job post.

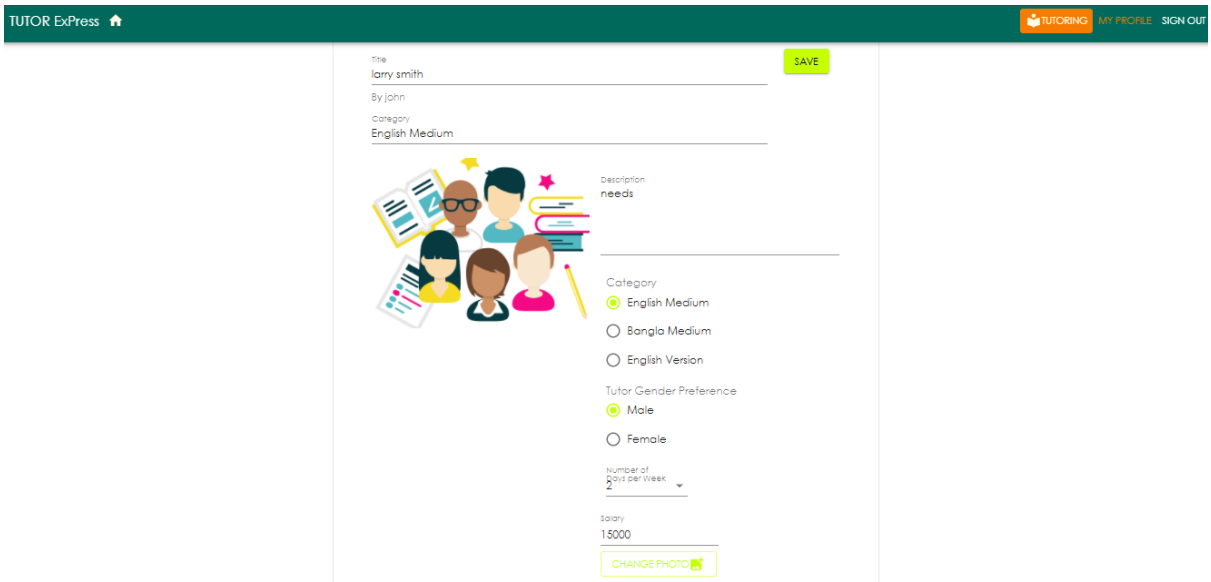


Figure 6.10: Screen view of Edit tuition page

- New Tutor Application form: When a user wants to apply to a tuition job post, they have to view the tuition post. In that page, the user is given a form to fill if they want to apply for that tuition job. After application, the user is taken back to home page.

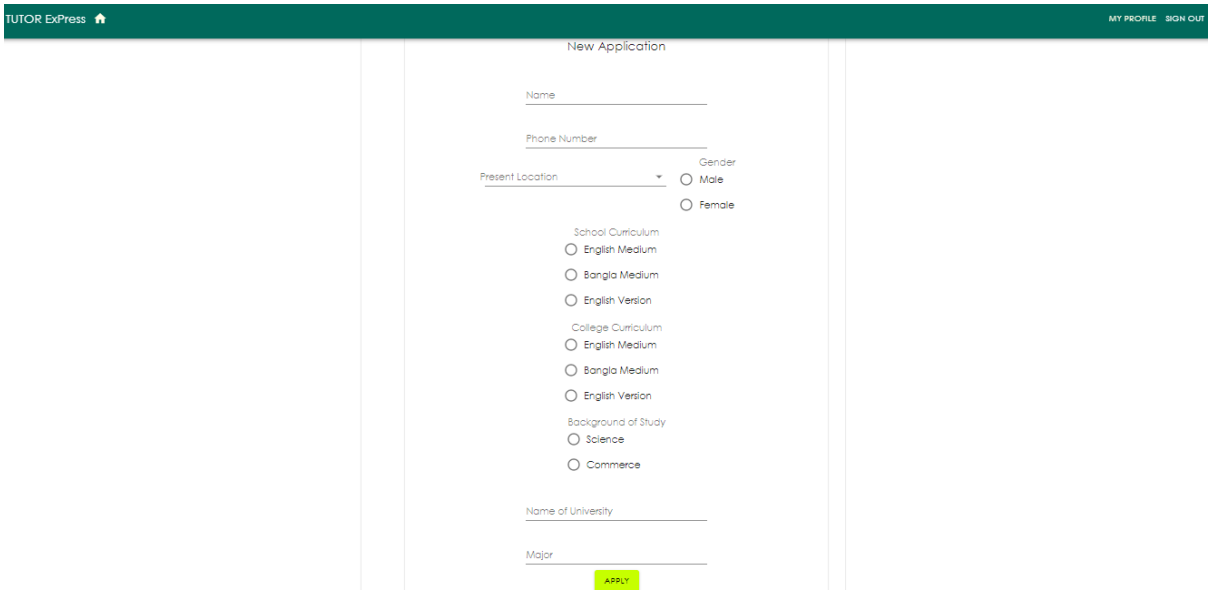


Figure 6.11: Screen view of New Tutor Application form

- Applications view page: When an application is submitted for a particular tuition job, the owner of that job will be able to view the application information. They will see this at the bottom, when they view that particular tuition job post.

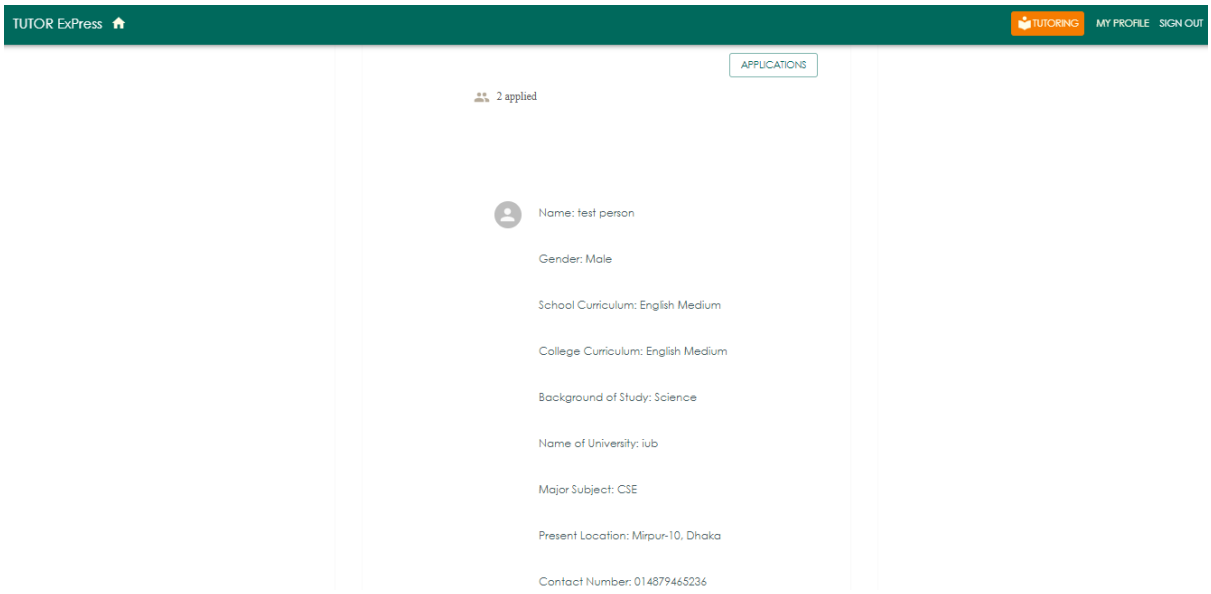


Figure 6.12: Screen view of Applications

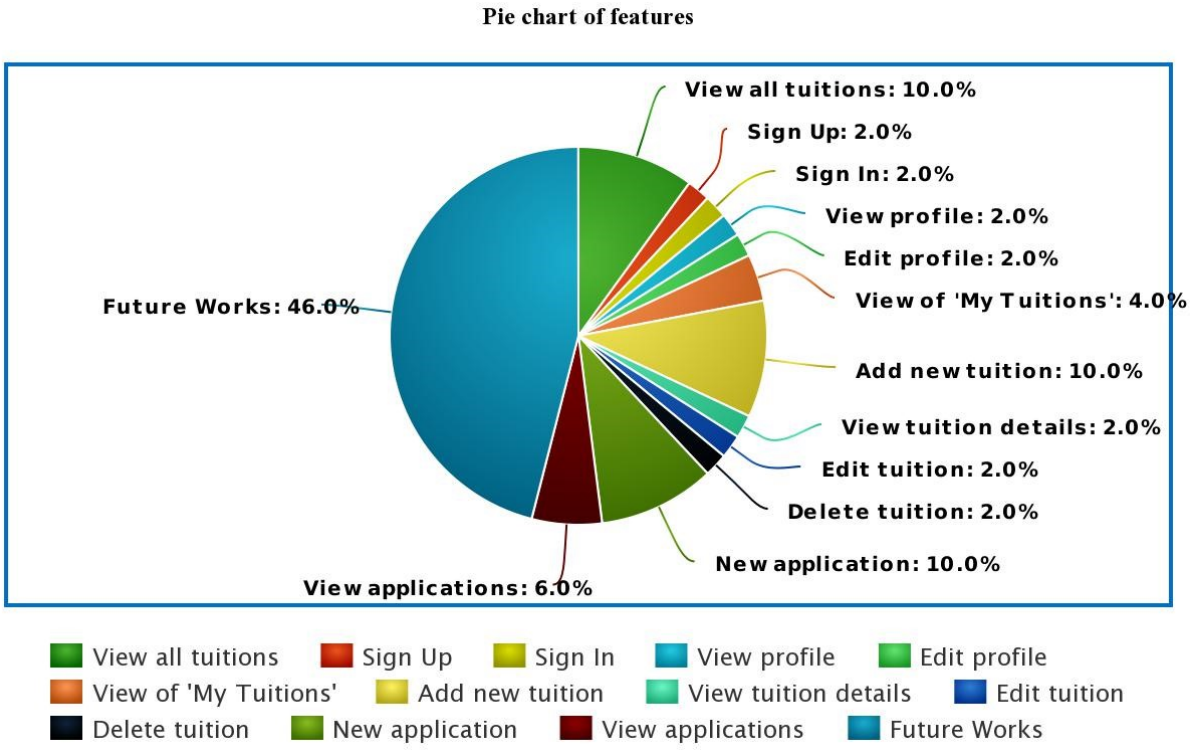


Figure 6.13: Pie chart of features

Features	Working %
View all tuition job post	100
New tuition form page	100
Tuition details page	100
Edit tuition page	100
New application form page	100
View application page	100

Table 6.1: Working features

Chapter 7

Project as Engineering Problem Analysis

7.1 Sustainability of the Project/Work

The responsibility for ensuring that project outputs, outcomes, and benefits are sustainable over their life cycles, as well as during their creation, usage, implementation, and disposal, is known as project sustainability [30].

Sustainability of a project depends on three pillars:

- Society Sustainability: It denotes the extent to which people will support the project and how actively they will do so. Support can take numerous forms, including using the app, subscribing to paid services, rating and providing comments, and so on. When "Tutor Express" is launched as a website, it is expected to attract a large number of users due to it's unique concept of connecting individuals and employment, as opposed to other platforms on the market. The web application can be considered to be society-sustainable as the user base grows throughout society.

- Economy Sustainability: This refers to how the application's operating costs will be maintained once it has been released, as well as if it will earn sufficient income to be profitable. Server costs, database storage costs, third-party API costs, and so on are all part of an application's running costs. When "Tutor Express" is a fully operational website, there will be advertisements put up in the pages to create revenue. The web app also tends to bring paid verification process for users. These types of revenue generation will cover up the maintenance and running charges for the web application. So the web application is sustainable in terms of economy.

- Environment Sustainability: This refers to how much the project will be able to control, maintain and operate if it is given to a different environment. If a new team or a different organization is given the project, will they be able to run the program. Usually the existing team works in further maintenance and any future updates and sometime work

with newer team for it. "Tutor Express" has many more features planned for itself to be executed in future. Since it is more focused on database, maintenance and smoother updates will be needed. The important features can also be used and worked on many other projects as well. This ensures that it is also sustainable for the environment.

7.2 Social and Environmental Effects and Analysis

People in this generation looks for jobs everywhere possible. Job market being such a hard place, people hunt for these in newspapers, social media, online platforms and in-person as well. Tuition jobs take up a small portion of this job market. When in need or in general, many look for them in their potential source. "Tutor Express" aims to be that potential source for everyone and create a path for them to be reached by the tutor seekers out there.

7.2.1 Social Effect

"Tutor Express" targets to create a direct connection between tuition job seekers and parents or people looking for a tutor for their younger ones. In case of potential tuition job seekers, they tend to face stress when they are not finding one tuition job. Stress for them also arises as they do not know if their application for the tuition jobs will reach to the other end. Similarly, parents are also giving time to look for potential tutors according to their needs. By creating this direct link, the web application ensures and saves time for the parent user. They can simply post for the tuition job they are offering and save time. On the other hand, tutors can be ensured that their application with their input information is being passed properly, reducing stress and delivering a positive impact.

7.2.2 Environmental Effect

"Tutor Express" saves time for the users, as they can be ensured that both ends are receiving appropriately. The app is also user friendly and easy to use which saves more time for them as well. This allows the users to give more time to themselves, family, and environment. Also, having that sense of positive impact on them can allow them to create a better environment, compared to the stress of having a hard time to earn and find tuition jobs.

7.3 Addressing Ethics and Ethical Issues

There are regulations and ethics that must be observed when working on producing and distributing an application in the world of the internet, where there is so much data harvesting, hacking, and crimes. Since they have all been taken into careful consideration, the creators of "Tutor Express" think that the program does not violate any code of behavior for application release and development. Some to be mentioned are:

- Collecting only relevant User data: The application collects only the data using forms that are relevant for a tuition job. The data are stored in the database. As the application aims to create a direct smooth link between two types of user, it may seem that a lot of data is taken as input. But each are only relevant for the purpose to be served.

- Not sharing or selling User data: Although some fields in the form input may seem a privacy concern for few users, the application does not reveal all to every user. The application also ensures that the data collected is not being shared or sold to any other service or third party API.

- Data Storage Security: Only the lead developer and owner of "Tutor Express" will have access to the database. As the database will be on a cloud server, only their login credentials will give them the access to it, in case of any development purpose.

- No use of profanity: There are no slang, swear words, or objectionable language in the project. The application's language and instructions are simple, straightforward, and to the point.

- No discrimination policy: Apart from gender and educational background, "Tutor Express" does not discriminate in any way on the basis of age, race, sexuality, religious beliefs, color, language, political or other viewpoint, national or social origin, property, birth, or any other status.

- Proper use of third party services and API: When "Tutor Express" takes any third part services or API in the future, it must make sure that no violation of the services take occurs during further development.

- Clean Promotion: "Tutor Express" only aims to promote the company that created it, the application itself and the service it provides.

- Clean Advertisement: The advertisements that will be running on "Tutor Express" will only be clean and positive and not disturbing to the users eyes. No advertisements with violence, nudity, blood, etc will be shown in the pages.

Chapter 8

Lesson Learned

8.1 Problems Faced During this Period

- Realizing the targets: Even though the requirements were known, there were often times when I would do other things that was not the actual requirement.
- Co operating with new technologies: React language was a first for me, along with working in a full-stack web development. Implementing React as a front end was also new.
- Feasibility of the project: There were times where questions aroused about the feasibility of certain features. Some results have shown troubleshoots where the feasibility had to be compromised.
- Staying with time: As the technologies were a first for me to work with, it was tough to meet the deadlines, thus slowed the pace of the overall development.
- Identifying bugs: There were many bugs found during development. Some were hard to identify, some were even harder to fix. There were bugs that would take few days to solve, especially bugs which was unfamiliar. Some involved styling bugs as well.

8.2 Solution of those Problems

- Planning: As soon as I started to understand that my beginning was not going as the requirements, I started to plan out the feasible requirements. I planned out the phases to be completed to move in order for next incoming phase and started developing to meet the requirements.
- Acquiring skill sets: I tried to learn from the resources available to me, in the from of documents and tutorials. But it was still a struggle to imply them in real life.
- Feasibility Analysis: The feasibility analysis for the project has kept me nudged that what is feasible for this project in the current environment. The analysis also help me to approach alternate ways when troubleshooting occurred.

- Reusable program: I have used reusable codes which can be used in many places for the development of this project in order to be time efficient and keep the pace moderate. On the other hand, I tried to be better in this skill set as well.
- Debugging: Debugging using console from the browser development tools was very essential. Some bugs required internet research to solve. I have also learned that styling should be kept as simple as possible. Some debugging required group meeting sessions with senior developers.

Chapter 9

Future Work & Conclusion

9.1 Future Works

The project, "Tutor Express" is still in its development phase and there are many more features that are thought of and yet to be developed in the future. Some future works to mention are:

- Create a chat feature between the applicant tutor and the tuition job owner
- Confirm button, if the user wishes to let the web application know which applicant was chosen for that particular tuition job
- Share tuition job post to other platforms or social media
- Integrate Google Maps for location
- Sign up or Sign in via Google or Facebook
- More animations and styled buttons
- Notifications feature for application received
- Responsive for different devices

9.2 Conclusion

I worked on a web application called "Tutor Express" during my internship period. In the web application, user can post a tuition job themselves and other interested users are open to apply to it. User can view who applied and contact with them on their own.

The internship period in Techdojo has been an outstanding experience. Gaining experience with the leading technologies like React, Node.js and Express.js is one of the major takeaways from the internship program. I learned about several types of program development approaches and how to create different types of applications. I also got a taste of what it's like to work as a developer. These experiences taught me more about what it takes to be a software engineer or

programmer, and they helped me prepare to be a more responsible and inventive developer in the future. Throughout my project work, I cooperated with my mentors and senior developers in the company to solve the roadblocks faced. In addition to skill sets, the project taught me to be more self-paced, disciplined, patient, take initiative, and have the bravery to address challenges. Furthermore, my communication skills have improved as a result of having to participate in weekly updates and address any programming concerns as well as correct them during programming sessions if necessary. Solving those difficulties enhanced my programming skills in the JavaScript language as a result of these problem-solving sessions. The React itself is a JavaScript library and Node.js and Express.js is also JavaScript for which I was able to be more involved in JavaScript. This internship has given me the opportunity to learn more about the development environment and market. I'd like to express my gratitude to everyone who has made my time as an intern so pleasant.

Bibliography

[1] Caretutors, "caretutors.com." [Online; accessed 01-September-2021].

[2] Tutorsheba, "tutorsheba.com." [Online; accessed 01-September-2021].

[3] "Selecting a development approach." [Re validated March 27, 2008].

[4] P. C. Pendharkar, J. A. Rodger, and G. H. Subramanian, "An empirical study of the cobb–douglas production function properties of software development effort," *Information and Software Technology*, vol. 50, no. 12, pp. 1181–1188, 2008.

[5] Tutorialspoint contributors, "Sdlc - overview." [Online; accessed 31-August-2021].

[6] "Human centred technology workshop 2006." PDF.

[7] React, "Reactjs." [Online; accessed 01-September-2021].

[8] Paul Krill, "React: Making faster, smoother uis for data-driven web apps," 2014. [INFOWORLD TECH WATCH].

[9] Chris Dawson, "Javascript's history and how it led to reactjs," 2014. [The New Stack].

[10] Wikipedia contributors, "Node.js — Wikipedia, the free encyclopedia," 2021. [Online; accessed 31-August-2021].

[11] Node, "Nodejs." [Online; accessed 01-September-2021].

[12] Wikipedia contributors, "Express.js — Wikipedia, the free encyclopedia," 2021. [Online; accessed 31-August-2021].

[13] Express, "Expressjs." [Online; accessed 01-September-2021].

[14] Wikipedia contributors, "Git — Wikipedia, the free encyclopedia," 2021. [Online; accessed 31-August-2021].

[15] Git, "Git." [Online; accessed 01-September-2021].

[16] Wikipedia contributors, "Discord (software) — Wikipedia, the free encyclopedia," 2021. [Online; accessed 31-August-2021].

[17] Discord, "Discord." [Online; accessed 01-September-2021].

[18] Wikipedia contributors, "Github — Wikipedia, the free encyclopedia," 2021. [Online; accessed 31-August-2021].

[19] Github, "Github." [Online; accessed 01-September-2021].

[20] Google, "Google docs." [Online; accessed 01-September-2021].

[21] Tutorialspoint contributors, "System analysis and design - overview." [Online; accessed 31-August-2021].

[22] Lonnie D. Bentley, "Systems analysis and design for the global enterprise," p. 160 7th Edition. [Book].

[23] Justis, R. T. and Kreigsmann, B., "The feasibility study as a tool for venture analysis," 1979. [Business Journal of Small Business Management].

[24] Georgakellos, D. A. and Marcis, A. M., "Application of the semantic learning approach in the feasibility studies preparation training process," 2009. [Information Systems Management].

[25] Young, G. I. M., "Feasibility studies," 1970. [Appraisal Journal].

[26] CEOpedia, "Economic feasibility." [Online; accessed 31-August-2021].

[27] Wikipedia contributors, "Systems design — Wikipedia, the free encyclopedia," 2021. [Online; accessed 31-August-2021].

[28] Tutorialspoint contributors, "Software requirements." [Online; accessed 31-August-2021].

[29] Altexsoft, "Non-functional requirements: Examples, types, how to approach," 21 Nov 2019. [Online; accessed 31-August-2021].

[30] youmatter, "Sustainability – what is it? definition, principles and examples," 18 June 2021. [Online; accessed 31-August-2021].