



islington college
(इस्लिङ्टन कॉलेज)

Module Code & Module Title

CS6P05NI Final Year Project

Assessment Weightage & Type

25% FYP Interim Report

1st Semester

2023 Autumn

PROJECT TITLE: MyTutor Nepal

Student Name: Prasanna Bahadur Chand

London Met ID: 22015727

College ID: np01cp4s220183

Internal Supervisor: Prithivi Maharjan

External Supervisor: Prajil Shrestha

Assignment Due Date: 3rd January 2023

Assignment Submission Date: 3rd January 2023

Word Count (Where Required): 3788

I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded.

Abstract

This report dives into the progress of my project which is a web application named MyTutor Nepal. MyTutor Nepal aims to solve the underlying problems in Nepal's education by providing a platform where students can book tutors to provide mentorship and guidance. This report includes a background study which discusses the understanding of the project, tools being used, methodology which will be used to develop them system and research conducted on similar systems. Development up to date, showcasing the progress to this date is also included with evidence. The diagrams made in this report are developed according to the Gantt chart. The document also consists of a detailed analysis of progress and a review for each progress. The final part of the report contains future works which showcases how the remaining tasks will be carried out with research included.

+

Table of Contents

1	<i>Introduction</i>	9
1.1	Project Introduction.....	9
1.2	Current scenario.....	9
1.3	Problem Scenario.....	10
1.4	Project as a solution.....	10
1.5	Aims and Objectives.....	11
1.5.1	Aim	11
1.5.2	Objectives	11
1.6	Structure of the Report	11
1.6.1	Introduction.....	11
1.6.2	Background.....	11
1.6.3	Development.....	11
1.6.4	Analysis of Progress	12
1.6.5	Future Work.....	12
2	<i>Background</i>	12
2.1	Understanding the project	12
2.1.1	UI Design.....	12
2.1.2	Programming language	12
2.1.3	Framework and libraries	12
2.1.4	Database.....	12
2.2	Methodologies	13
2.2.1	Considered methodologies	13
2.2.2	Selected methodology	15
2.3	Similar projects.....	17
2.3.1	Preply.....	17
2.3.2	Mero Tutor.....	17
2.3.3	Teacheron	19
2.3.4	mySecondTeacher.....	20
2.3.5	TutorMate Nepal.....	20
2.4	Comparison table.....	21

2.5	Critical analysis	21
3	<i>Development to date</i>	22
3.1	Use case diagram	22
3.1.1	High level use case	22
3.1.2	Expanded use case	23
3.2	Initial entity relationship diagram	23
3.3	Activity diagram	24
3.3.1	Register account.....	25
3.3.2	Login account	26
3.4	Sequence diagram.....	27
3.4.1	Login sequence diagram	28
3.4.2	Register sequence diagram	29
3.5	Collaboration diagram.....	30
3.5.1	Login collaboration diagram.....	30
3.5.2	Register collaboration diagram	31
3.6	Data Flow Diagram	32
3.6.1	Context level DFD	32
3.6.2	Level 1 DFD	33
3.7	Mockup design	36
3.7.1	Login mockup	36
3.7.2	Register mockup	37
3.7.3	Booking/home page mockup	38
3.7.4	Profile dropdown mockup	39
3.7.5	View profile mockup	40
3.7.6	Edit profile mockup	41
3.7.7	Map view mockup	42
3.7.8	Tutor mini profile in map mockup	43
3.8	Developed features.....	44
3.8.1	Login screen with validation.....	44
3.8.2	Register screen with validation.....	45
3.8.3	Authorized access to home page.....	45
4	<i>Analysis of Progress.....</i>	46

4.1	Progress table.....	46
4.2	Progress review.....	47
4.3	Progress timeline	48
4.4	Revised Milestone	49
5	<i>Future works</i>	49
6	<i>Appendix.....</i>	51
6.1	Survey results.....	51
6.2	Project introduction	57
6.3	Understanding the project	57
6.4	Use case diagram	58
6.4.1	High level use case	58
6.4.2	Expanded use case	61
6.5	Considered methodologies	71
6.5.1	Scrum methodology	71
6.5.2	Waterfall methodology	72
6.6	Researched projects.....	72
6.6.1	mySecondTeacher.....	72
6.6.2	TutorMate Nepal.....	73
6.7	Software Requirement Specifications	76
6.7.1	Introduction.....	76
6.7.2	Overall Description.....	77
6.7.3	Functional requirements	81
6.7.4	External User Interface Requirements	89
6.8	Code	90
6.8.1	Login client side	90
6.8.2	Login server side.....	91
6.8.3	Register client side.....	92
6.8.4	Register server side.....	93
7	<i>Bibliography</i>	94

Table of figures

Figure 1: V model	14
Figure 2: Structure of evolutionary prototype	16
Figure 3: Preply booking page	17
Figure 4: MeroTutor booking page.....	18
Figure 5: MeroTutor teacher profile	18
Figure 6: Teacher listing	19
Figure 7: Teacher profile for booking.....	20
Figure 8: Use case diagram.....	22
Figure 9: Initial ERD	23
Figure 10: Activity diagram.....	24
Figure 11: Activity diagram - Register account.....	25
Figure 12: Activity diagram - Login account	26
Figure 13: Sequence diagram.....	27
Figure 14: Sequence diagram - Login.....	28
Figure 15: Sequence diagram - Register.....	29
Figure 16: Collaboration diagram.....	30
Figure 17: Collaboration diagram - Login	30
Figure 18: Collaboration diagram - Register	31
Figure 19: Data flow diagram – Context level.....	32
Figure 20: Data flow Diagram - Level 1.....	33
Figure 21: Context level DFD - Login.....	34
Figure 22: Level 1 DFD - Login	34
Figure 23: Context Level DFD - Register.....	35
Figure 24: Level 1 DFD - Register	35
Figure 25: Login mockup.....	36
Figure 26: Register mockup.....	37
Figure 27: Home page mockup.....	38
Figure 28: Profile dropdown mockup	39
Figure 29: View profile mockup.....	40
Figure 30: Edit profile mockup.....	41

Figure 31: Map view mock up	42
Figure 32: Tutor map mini profile mockup	43
Figure 33: Developed login page with validation.....	44
Figure 34: Developed sign up page with validation	45
Figure 35: Authorization.....	45
Figure 36: Revised Gantt chart	48
Figure 37: Revised milestone.....	49
Figure 38: Have you used any application/website to book a tutor?	51
Figure 39: Was the hiring process for these application well managed and simplified?.....	52
Figure 40: Select some features that the application had.....	52
Figure 41: For what reasons have you had to book a tutor?	53
Figure 42: What do you look for when booking a tutor?.....	53
Figure 43: Is the tutor's location a major concern for you?	54
Figure 44: Would you be more likely to book a tutor closer to your location?	54
Figure 45: Is the tutor's authenticity a major concern when using this application?	55
Figure 46: What measures should be taken by an application to validate the authenticity of tutors? .55	
Figure 47: What documents must a tutor submit for qualifications and authenticity to be approved as a registered tutor?	56
Figure 48: Should there be a panel for parents to view pending and completed assignments by their children particularly for students in primary education?	56
Figure 49: Should users of the application receive announcements about major updates?	57
Figure 50: mySecondTeacher assignment page:.....	73
Figure 51: Finding a tutor	74
Figure 52: Tutors recommended to me based on preferences	75
Figure 53: Product perspective diagram	77
Figure 54: Login client side code.....	90
Figure 55: Login server side code.....	91
Figure 56: Register client side code.....	92
Figure 57: Register server side	93

Table of tables

Table 1: Advantages and disadvantages of evolutionary prototype model	15
Table 2: Comparison table	21
Table 3: Progress table	46
Table 4: Sign up expanded use case	61
Table 5: Log in expanded use case	62
Table 6: View profile expanded use case	63
Table 7: Become tutor expanded use case	64
Table 8: Book a tutor expanded use case.....	65
Table 9: Make payments expanded use case	66
Table 10: Rate tutor expanded use case.....	67
Table 11: Receive assignment expanded use case.....	67
Table 12: Submit assignment expanded use case	68
Table 13: View assignment expanded use case	69
Table 14: View students expanded use case	70
Table 15: Perform administration activities expanded use case	71
Table 16: Functional requirement for register user.....	81
Table 17: Functional requirement for login user	82
Table 18: Functional requirement for book tutors	83
Table 19: Functional requirement for payment	84
Table 20: Functional requirement for payment	85
Table 21: Functional requirement for notifications	86
Table 22: Functional requirement for assignment	87
Table 23: Functional requirement for rating and review	88

1 Introduction

1.1 Project Introduction

Searching for a tutor can be a difficult process for many different reasons. First of all, it might be difficult to locate a tutor who specializes in a subject that you are looking for. Then, there is the matter of availability where finding someone who aligns with your schedule. Additionally, the location of the tutor is also important so that the tutor does not have any problem getting to your location. Also, finding a teacher whose teaching style matches your preference could also be challenging. On top of all this, finding a tutor that is affordable might not be easy. For these reasons, it can be hard to find the best tutor. MyTutor Nepal aims to provide a platform which makes it convenient for struggling students to find and book the right tutor to their location.

More on how the web application works is shown in the appendix. ([Project introduction](#))

1.2 Current scenario

Education is a social institution through which people are taught basic academic knowledge, learning skills and cultural norms. (Sujan, 2021) In the context of Nepal, the education system currently has several problems. The lack of provisions for training and practical experience has affected the student's ability to develop to their full potential. Nepal suffers from an outdated educational structure. Many students face problem in higher education as their root in primary education is weak. (Saud, 2023) In the recent Grade 12 National Education Board (NEB) Examination, only 51% of the students earned passing grades and almost half of the participants could not graduate to university. (Whose failure?, 2023) In a survey taken from 50 people in Nepal, 47 had booked a tutor. 25 people booked a tutor due to difficult course, 11 booked it due to poor academic teaching and 23 booked it for exam preparation where multiple selections were allowed. ([Figure 41](#))

Looking at the global statistics 90 percent of the world's population had completed primary education. (Dyvik, 2023) Comparatively, Nepal has a completion rate of 79% according to UNICEF as of 2022 (Nepal Education Fact SheetsI 2022 - UNICEF Data, n.d.). Globally, the world has a higher completion rate at root level compared to Nepal which can be due to the poor course structure discussed above.

1.3 Problem Scenario

- As studied above, there is a lack of sufficient mentorship and guidance for students to support their education needs.
- Tutors that are located close to a student's area may be hard to find. In the survey, among 50 people, 76% thought that the tutor's location is a major concern ([Figure 43](#)) and 88% would be more likely to book a tutor closer to their location. ([Figure 44](#))
- There is a lack of well managed platform for students to find tutors. ([Figure 39](#)) In this survey result, it can be seen that 72% of the 47 that responded, felt that the existing tutor booking applications were not well managed. Also, here it can be seen that half of the users that used existing application to book tutors, did not find features listed in the figure. ([Figure 40](#)[Figure 40](#))
- Tutoring services can be expensive with some tutors charging more than others. Without a proper system, it can be hard to find an affordable tutor.

1.4 Project as a solution

- This application displays listed tutors along with their price, rating and subject proficiency, giving students multiple tutors to choose from.
- The application makes it possible for users to view tutors in a map-based interface where the location of the tutors can be pinpointed.
- The application will include features that the survey responders lacking in other systems such as maps, payment integration, rating system, profile customization, notifications and online assignments.
- This application makes it easy for tutors and students to come to a mutual agreement in terms of pricing as tutors can set their price which will be displayed to users.

1.5 Aims and Objectives

1.5.1 Aim

The aim of this project is to create a reliable web application which provides a platform for students to find and connect with qualified tutors based on their specific needs.

1.5.2 Objectives

The objectives of this project are:

- To build a user-friendly interface which allows users to perform operations with simplicity.
- To include parental involvement in the student's academic activities.
- To assist tutors in finding and teaching students.
- To help users find the right tutors through AI assisted rating system.
- To help tutors and users decide on a non-negotiable price.

1.6 Structure of the Report

1.6.1 Introduction

The introduction consists of a brief overview of the project, current scenario of education in Nepal and how tutors contribute in minimizing the problems, problems with booking tutors and the proposed project as a solution.

1.6.2 Background

Background consists of information about understanding of the project. It contains information like tools and resources being used, similar projects and what features are missing in them and the methodology that will be followed to develop the web application.

1.6.3 Development

The development section consists of various diagrams that have been designed so far which include data flow diagram, entity relationship diagram, use case diagram, activity diagram, sequence diagram and collaboration diagram. It also contains the mockup design made using Figma and the developed functionalities to this date.

1.6.4 Analysis of Progress

This section dives into the progress achieved so far into the project. It gives a clear visual representation of the work completed and the remaining work needed to be done. The progress to this date are reviewed in this section.

1.6.5 Future Work

The final section of this document highlights the expected work that remains. The different topics that are yet to be completed are categorized into different topics. Research based insights are provided on how the work will be carried out.

2 Background

2.1 Understanding the project

The following are the tools and libraries being used in the project. The choice of each tool and their relevance are in the appendix. ([Understanding the project](#))

2.1.1 UI Design

- **Figma**

2.1.2 Programming language

- **JavaScript**

2.1.3 Framework and libraries

- **ReactJS**
- **ChakraUI**
- **NodeJS**
- **ExpressJS**

2.1.4 Database

- **MongoDB**

2.2 Methodologies

2.2.1 Considered methodologies

2.2.1.1 Spiral methodology

The spiral methodology is an iterative model in which objectives are determined, requirements are gathered, alternate solutions are proposed and after risk analysis, development and testing is done. Although good for projects that require high risk management, this model is not the best solution for small and medium scale projects. The reason for not choosing this methodology are:

- The risk analysis phase in each stage is time consuming.
- This model gives more emphasis on documentation which is time consuming.
- Project deadlines are missed as number of phases is not known while planning.

(Rana, 2021)

2.2.1.2 Rapid Application development methodology (RAD)

Rapid Application development is an agile software development approach in which more focus is given on developing the software without any planning. The plan is formulated as the software is being developed. (SDLC - RAD Model, n.d.) RAD involves developing a demonstrable prototype, gather feedback, improve the prototype and finalize the product. The reasons for not choosing this methodology are:

- Requires strong and collaborative effort of many people.
- It is more suitable for projects that need to be delivered quickly.

(Egeonu, 2022)

2.2.1.3 V Model

The V Model, also known as Verification and Validation model is a SDLC model in which testing takes place in each stage of development cycle. The next phase can only start after the completion of the previous phase. The phases in this model are, requirements gathering, system design, architecture design and module design. Corresponding to each development phase, testing needs to be done to catch defects early. Although easy to understand and use, I did not choose this model because:

- After testing is done, it is difficult to go back to previous stage.
- It is a poor model for long projects.

- It is suitable for projects where requirements are clear from the start.

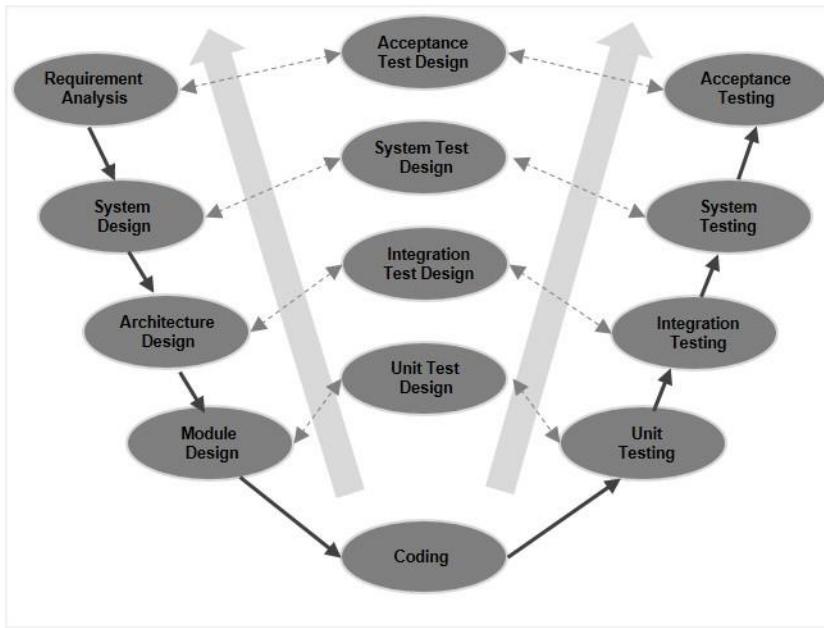


Figure 1: V model

2.2.1.4 Scrum methodology

The reasons for not selecting this methodology are mentioned in the appendix. ([Scrum methodology](#))

2.2.1.5 Waterfall methodology

The reasons for not selecting this methodology are mentioned in the appendix. ([Waterfall methodology](#))

2.2.2 Selected methodology

2.2.2.1 Evolutionary prototype

Evolutionary prototype is a type of methodology where requirements are gathered before the project begins. After requirements are analysed and understood, design and development begins. Initially, a usable prototype of the final product is developed and tested by the user. The user gives feedback which is used to refine the previous prototype and develop a better version of the product. Each prototype consists of those features which are clearly identified understood to develop a minimum viable product (MVP). (Lal, 2022)

This methodology allows me have significant client involvement which ensures that requirements for the project are met. Changes are integral part of development and helps make the application satisfactory for client and users. Requirements can be added and features can be removed with each client feedback. The prototype can be changed multiple times without throwing away the previous versions. This allows me to understand and prioritize client requirements and take the best path to deliver the final product without adding any unnecessary features. When the client decides that the product is good enough, the prototype cycles will stop and the final product will be developed. The final system will be tested and distributed to production. Its advantages and disadvantages are as follows:

Advantages	Disadvantages
Allows incremental development in the form of prototypes.	It is unknown how many prototypes will be there so deadlines could be missed.
Facilitates client feedback which will lead to satisfaction for the end users.	Could require more time as changes are unpredictable.
Reduces risk of failure as each stage allows client to identify issues early.	The customer might lose interest if prototype does not meet their expectations.
It makes it easier to detect any missing functionalities.	Documentation could be difficult as the project is continuously changing.

Table 1: Advantages and disadvantages of evolutionary prototype model

Evolutionary Prototyping Life Cycle for a Project

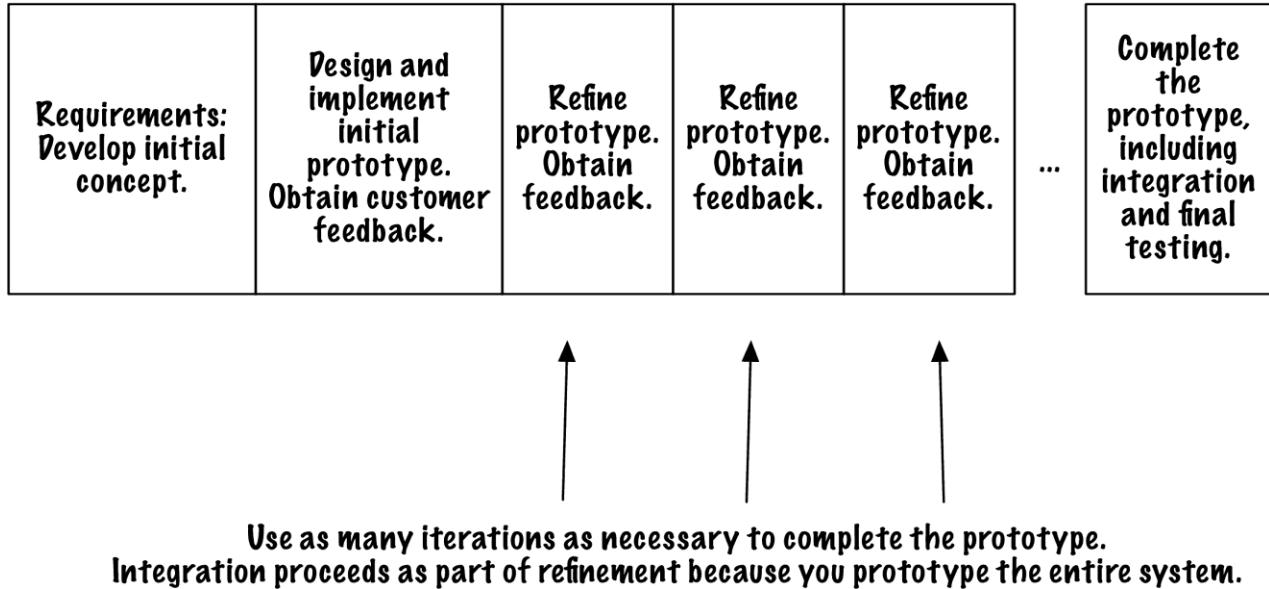


Figure 2: Structure of evolutionary prototype

The project will be done in the following way:

- i. **Requirement Gathering** – This step involves discussing with the client about what is expected from the system and what features must be added. The next step can only start when requirements are clearly understood. This step has already been completed and requirements have been finalized.
- ii. **Develop initial prototype** – With the requirements well understood, I will develop an initial prototype which will consist of all the major features that were well understood to form a minimum viable product. For each functionality, the necessary diagrams will be designed first. This designed prototype will be demonstrated to the client for feedback.
- iii. **Refine prototype and obtain feedback** – As the initial prototype is not expected to be fully ready for production, the client feedback will be used to refine the previous prototype. Since it is unknown when the client will be satisfied, this process could be repeated several times if not accepted the first time. With each feedback, the product will update the requirements specified by the client.
- iv. **Test and release final product** – After the final prototype is accepted by the client, it is ready to be released. Before releasing the product, it is important that quality is assured so

testing is conducted to prevent any breaking defect that may occur. This ensures a smooth release which will mark the completion of the project.

2.3 Similar projects

2.3.1 Preply

Preply is an online language learning platform that connects tutors to learners throughout the world and provides online lessons. Preply uses machine learning algorithm to recommend the best tutors for each learner. Preply also allows users to become tutors and teacher other learners. Similar to this web application, my project will also consist of filters for subjects, pricing and availability to search for tutors and users will also be able to become tutors.

The screenshot shows the Preply booking page interface. At the top, there are four dropdown filters: 'I want to learn Spanish', 'Price per lesson \$1 - \$40+', 'Country of birth Any country', and 'I'm available Any time'. Below these are four smaller dropdowns: 'Specialties', 'Also speaks', 'Native speaker', and 'Super tutor'. To the right are buttons for 'Sort by: Our top picks' and a search bar 'Search by name or keyword'. A message indicates '6,107 Spanish teachers available'. Two tutor profiles are displayed. The first profile is for 'Silvia R.' (Super tutor), who is Spanish and speaks English (Advanced). She has a 5-star rating, 27 reviews, and charges \$42 for a 50-min lesson. The second profile is for 'Catalina J.', a Spanish Language Teacher at Instituto Cervantes, with a 5-star rating, 35 reviews, and charges \$22 for a 50-min lesson. Both profiles include a 'Book trial lesson' button and a 'Send message' button. A video preview for Silvia R. shows her speaking and includes a play button.

Figure 3: Preply booking page

2.3.2 Mero Tutor

Mero Tutor is an online platform used for hiring private tutors for different subjects and classes. Mero Tutor allows users to find listed teachers for home tutoring and contact them through phone or email. Although this platform is good for just finding tutors, it lacks functionalities for payments and automating requests. My project aims to refine and implement features that

this application is lacking. MyTutor Nepal will implement payment integration, google maps to assist with finding tutors, feature to give assignments to students and many more.

Health Education Teachers For Grade 6 To 8 In Kathmandu, Nepal

The screenshot shows a search interface for finding teachers. At the top, there are dropdown menus for 'Grade 6 to 8', 'Health Education', and 'KATHMANDU'. On the left, there are several filter categories with checkboxes:

- Advanced Filters**
- Teaching Experience**: Options include 0 to 1 Years, 2 to 4 Years, 5 to 7 Years, 8 to 10 Years, and 10+ Years.
- Gender**: Options include Male, Female, and Other.
- Teaching Location**: Options include Student's Home, Tutor's Place, and Both.
- Qualification**: Options include Higher Secondary, Bachelors Degree, and Masters Degree.

Below the filters, four teacher profiles are listed:

- Priyanka Nayak**: Female, 25 years, Bachelors Degree (Completed), 4 years experience. [View Profile](#)
- Seema Bhandari**: Female, 30 years, Masters Degree (Completed), 5 years experience. [View Profile](#)
- Sadiksha Khadgi**: Female, 25 years, Bachelors Degree (Pursuing), not yet certified. [View Profile](#)
- Puskar Balal**: Male, 33 years, Masters Degree (Pursuing), not yet certified. [View Profile](#)

Figure 4: MeroTutor booking page

A teacher profile for **Sean Cho**, a male, 26 years old, from Kathmandu. He has a Bachelor's degree in Pursuing and 3 years of teaching experience. He offers tutoring at Student's home or My own place.

About Me

Hello, I am currently studying at Baylor University, Waco, Texas. I will be here till 27th of July and I am willing to help during that time. In addition to this, I am not Nepali, I am Korean.

Figure 5: MeroTutor teacher profile

2.3.3 Teacheron

Teacheron is a free to use website used by thousands of tutors all over the world. Tutors can be listed for either online or in person teaching. The application has an assignment job feature that allows users to book tutors for the purpose of completing an assignment. The application also allows users to post learning resources which can be bought by other users. To find a tutor the user must select the country and subject that they want to be tutored in. Additionally, users can put a request to find tutors by posting their location, phone number and requirements. The application shows the tutor's experience, fees, education, reviews, specialization subject and other relevant information.

Ayush Pandey Student of pulchowk campus.maths and physics tutor

Maths Physics Computer



Hi,I am Ayush Pandey . I am currently pursuing my bachelor of engineering from pulchowk college lalitpur. i can fluently teach students maths and physics upto class 12. My gpa in class 11 and 12 are 4 and 3.82 respectively. pls contact me for further information. currently living in budhanilkantha and have scooter for travelling so i am very...

Bagmati Province ₹ 5,000/month 0.0 yr. 1.0 yr. 3 km

Pratyusha Uperty Maths, Science

Maths Science



Completed my +2 taking major science (biology) and is currently a 1st semester bachelor taking IT major. Will help in problem solving and analysis. Will teach in a fun way. It will be my 1st year in teaching experience but can ensure that i will be of great help. I may not have a good experience in a teaching job but i have been tutoring my...

Bagmati Province ₹ 5,000–15,000/hour 0.0 yr. 0.0 yr. 20 km

Rabin Acharya Interactive teaching, Software Engineer

Maths Computer

Figure 6: Teacher listing

Ayush Pandey Student of pulchowk campus.maths and physics tutor

★ No reviews yet

Hi,I am Ayush Pandey . I am currently pursuing my bachelor of engineering from pulchowk college lalitpur. i can fluently teach students maths and physics upto class 12. My gpa in class 11 and 12 are 4 and 3.82 respectively. pls contact me for further information. currently living in budhanilkantha and have scooter for travelling so i am very flexible. i have experience pf teaching for grade 8 students for ble in the school nearby me. My methodoly for teaching-

- 1)Conceptual Understanding
- 2)Step-by-Step Problem Solving
- 3)Real-World Applications
- 4)Problem-Solving Techniques

Subjects

Maths Grade 8-Grade 12
Physics Grade 8-Grade 12
Computer Grade 8-Grade 10

Experience

No experience mentioned.

Education

bachelor in civil engineering (May, 2023–now) from IOE Pulchowk Campus, Lalitpur –scored 3.82 +2

Profile Summary



Message Phone Pay Review

- Location: Budhanilakantha, Nepal
- Can travel: 1.86 mi (3.0 km)
- Last login: Dec 23
- Registered: Dec 23
- Total Teaching exp: 1.0 yrs.
- Teaches online: Yes
- Online Teaching exp: 0.0 yrs.
- Teaches at student's home: Yes
- Homework Help: Yes
- Gender: Male
- Works as: Individual teacher
- Fee: ₹5,000/month (USD 37.59/month)

Figure 7: Teacher profile for booking

2.3.4 mySecondTeacher

Research on mySecondTeacher is shown in the appendix. ([mySecondTeacher](#))

2.3.5 TutorMate Nepal

Research on TutorMate Nepal is shown in the appendix. ([TutorMate Nepal](#))

2.4 Comparison table

S.N	Features	Preply	mySecondTeacher	MeroTutor	TutorMate Nepal	Teacheron	MyTutor Nepal
1.	Payment integration	✓	✓	✗	✓	✓	✓
2.	Map based booking feature	✗	✗	✗	✗	✗	✓
3.	Listing based booking feature	✓	✗	✗	✗	✓	✓
4.	Assignments provided to students	✗	✓	✗	✗	✗	✓
5.	AI Based rating	✗	✗	✗	✗	✗	✓
6.	Review/rating system	✓	✗	✗	✓	✓	✓
7.	Parent panel	✗	✗	✗	✓	✗	✓
8.	Become a tutor	✓	✗	✓	✓	✓	✓
9.	Set schedule	✓	✗	✗	✓	✗	✓

Table 2: Comparison table

2.5 Critical analysis

The proposed application, MyTutor Nepal will include all of the features mentioned in the table above. The motive for developing this application is not to replace existing applications but to refine them by providing more useful functionalities. In the similar projects mentioned above, none of the projects had a map based booking feature, which is useful for locating nearby tutors. This functionality aims to provide an alternate interface for booking a tutor while taking their location into account for several reasons, leading to a better user experience. The map view will show a mini profile for each tutor which will among other things, show the AI rating for the tutor which is calculated using sentiment analysis. This feature, which is also lacking in all other similar projects, will sum up all the reviews and rating of tutors and categorize them as either positive, negative or neutral. This feature is useful as it aids the user's decision making and provides an efficient way of comparison between tutors.

3 Development to date

3.1 Use case diagram

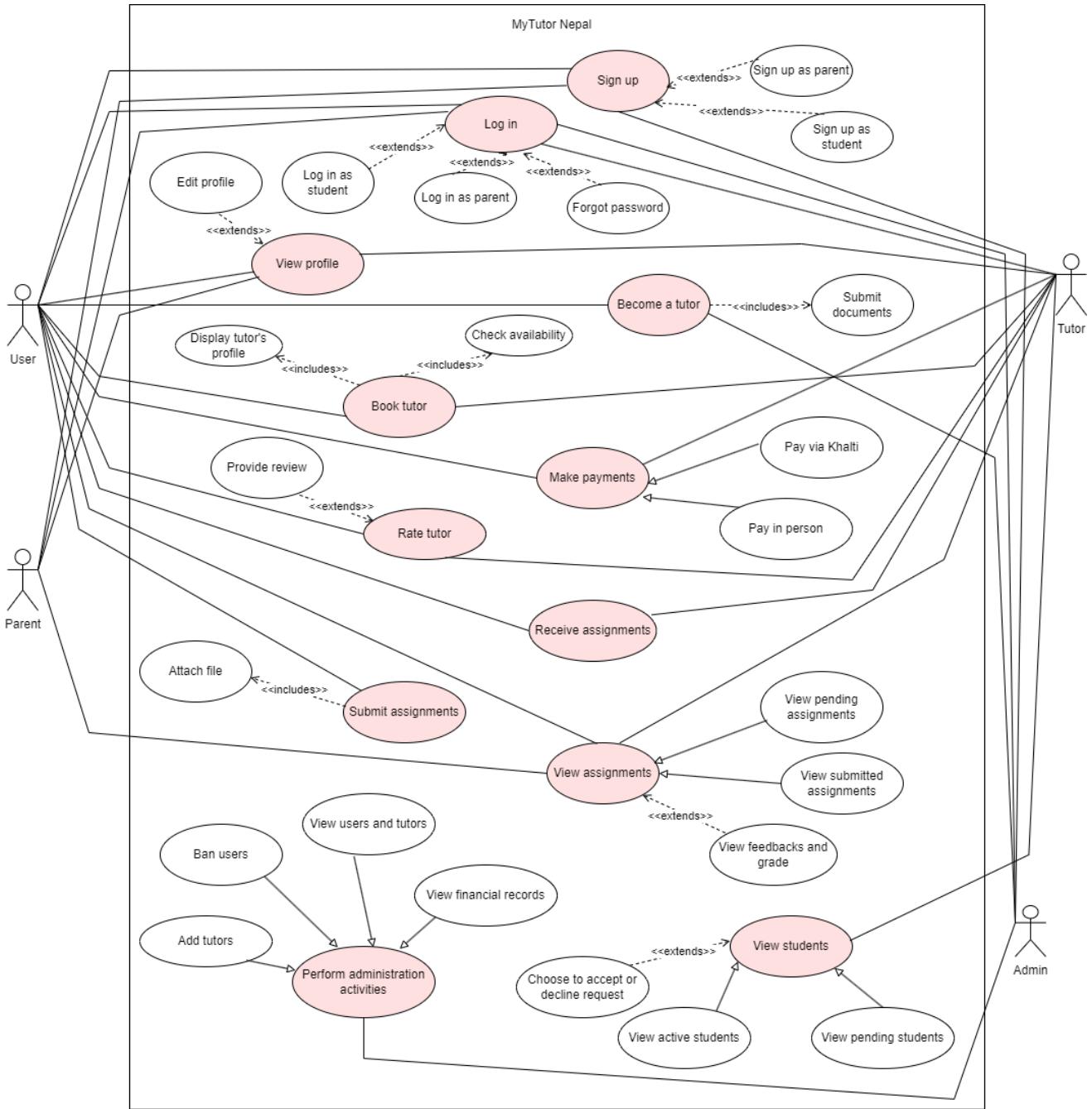


Figure 8: Use case diagram

3.1.1 High level use case

The high level use cases are in the appendix ([High level use case](#)).

3.1.2 Expanded use case

The high level use cases are in the appendix ([Expanded use case](#)).

3.2 Initial entity relationship diagram

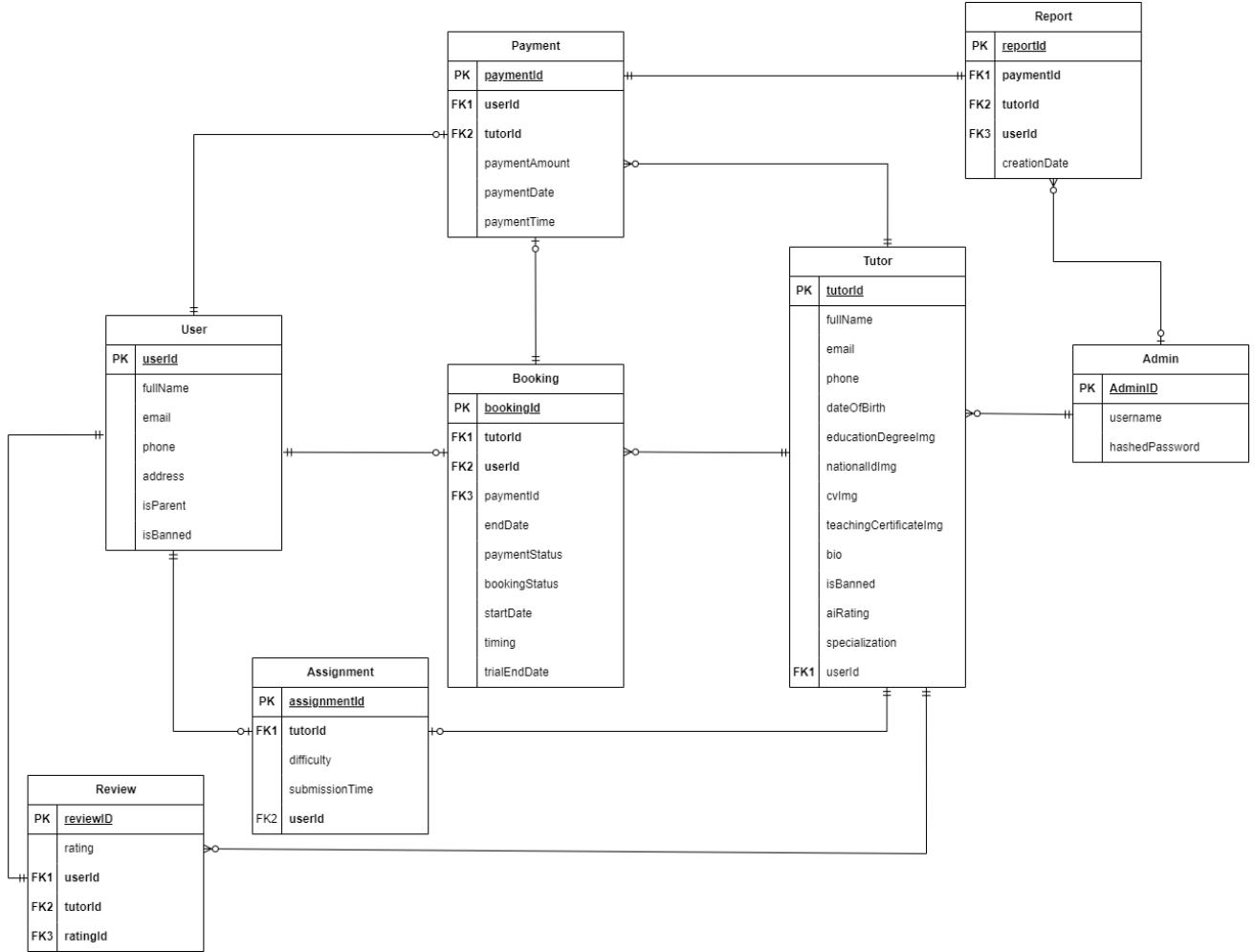


Figure 9: Initial ERD

3.3 Activity diagram

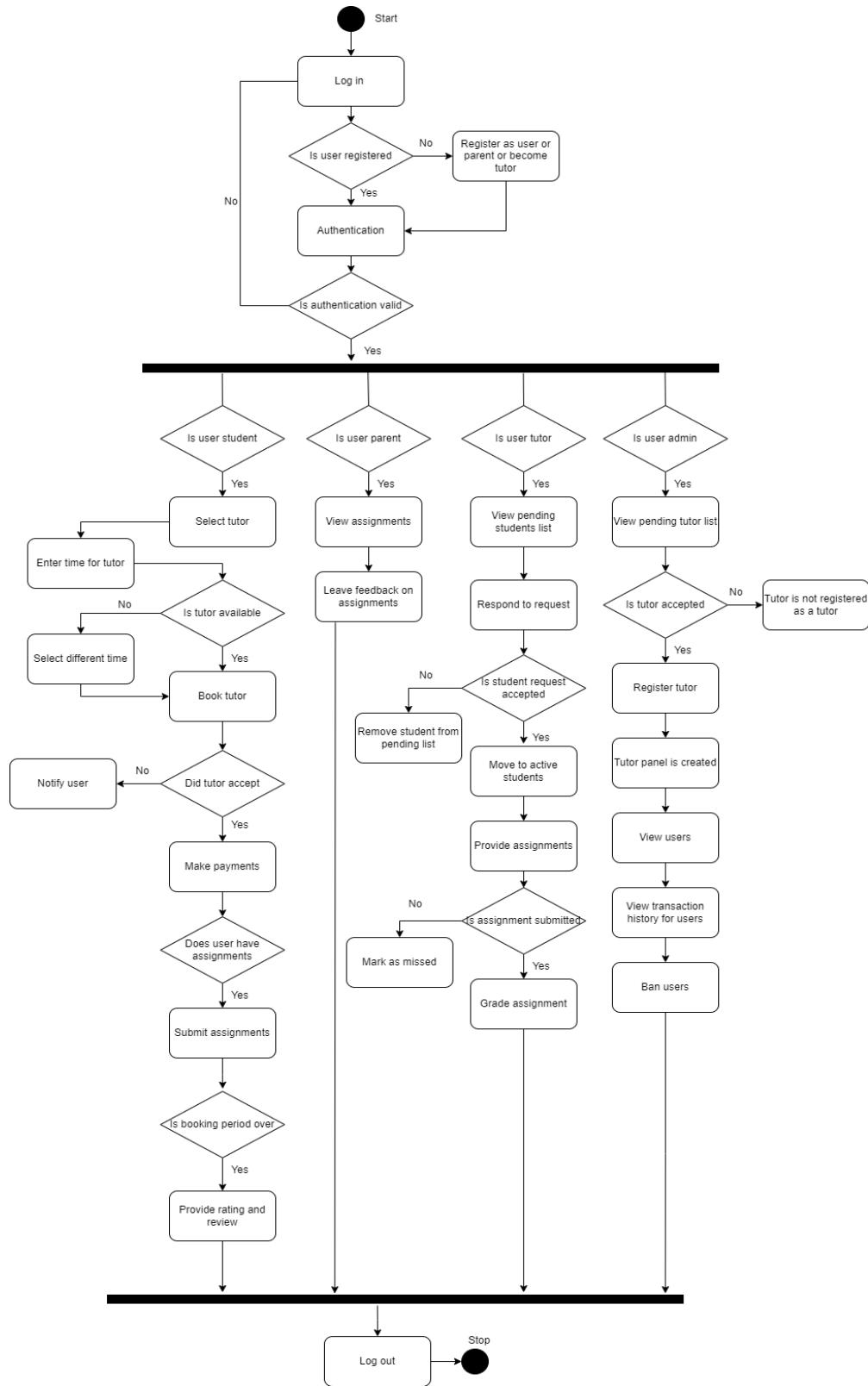


Figure 10: Activity diagram

3.3.1 Register account

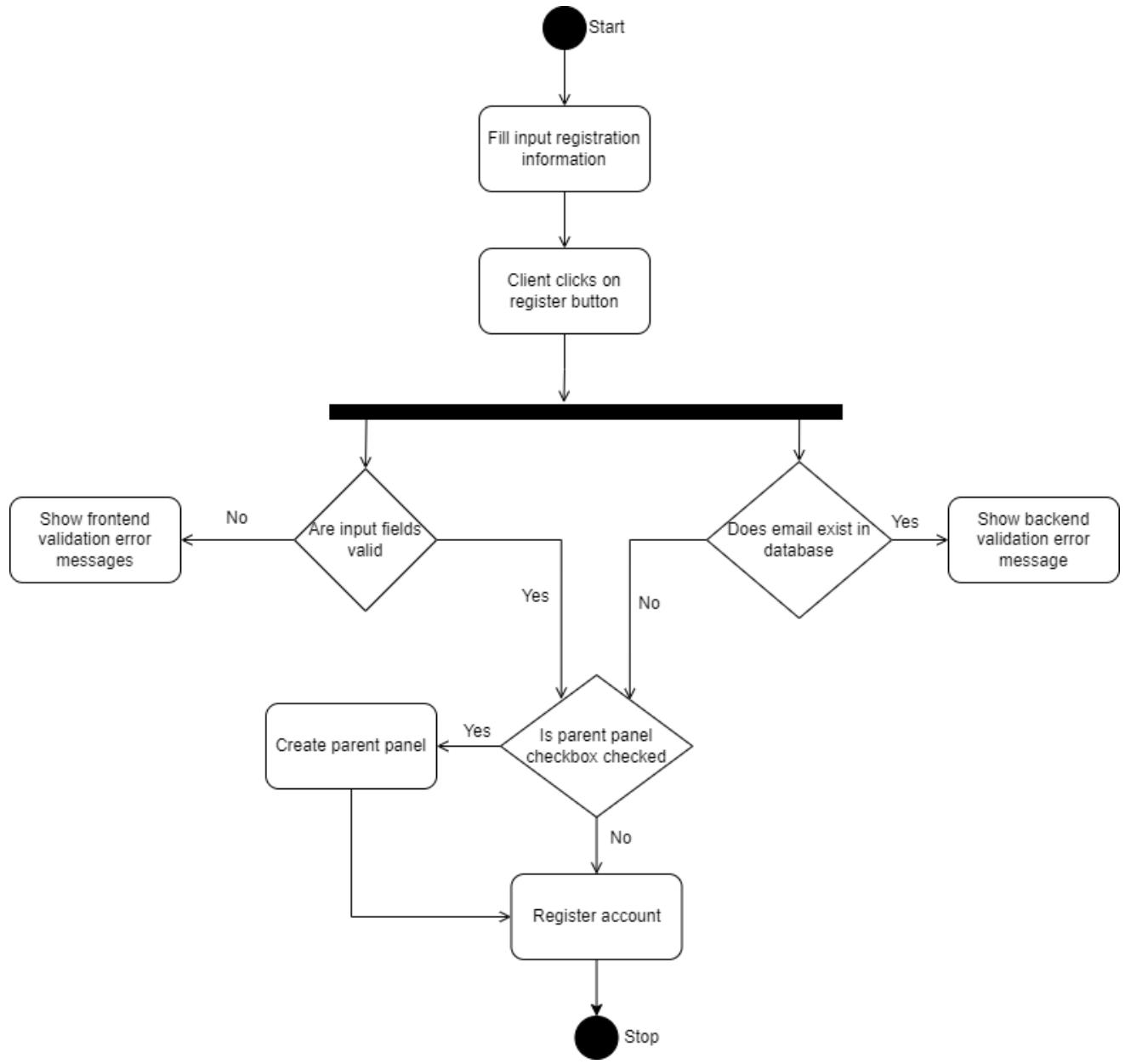


Figure 11: Activity diagram - Register account

3.3.2 Login account

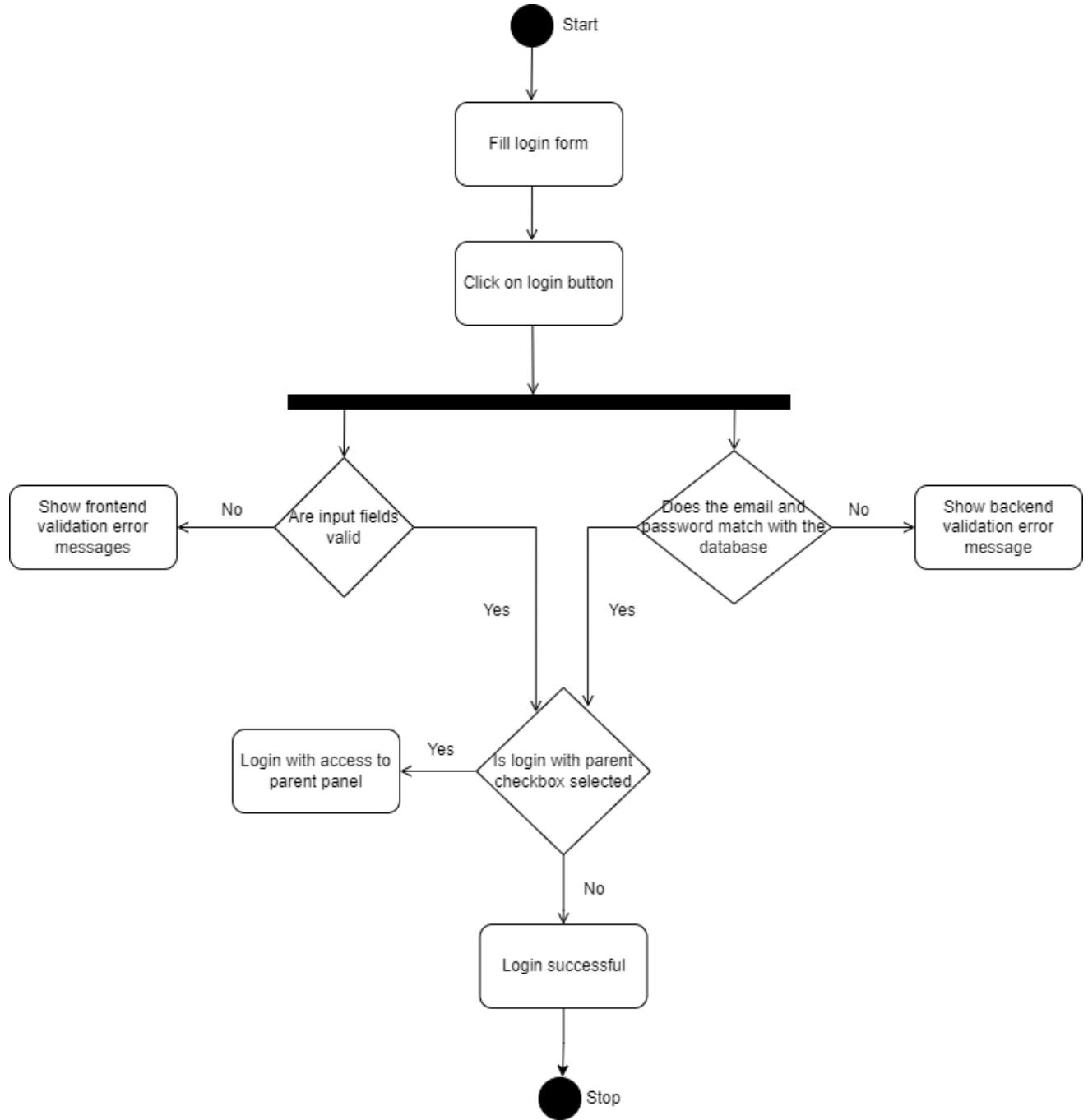


Figure 12: Activity diagram - Login account

3.4 Sequence diagram

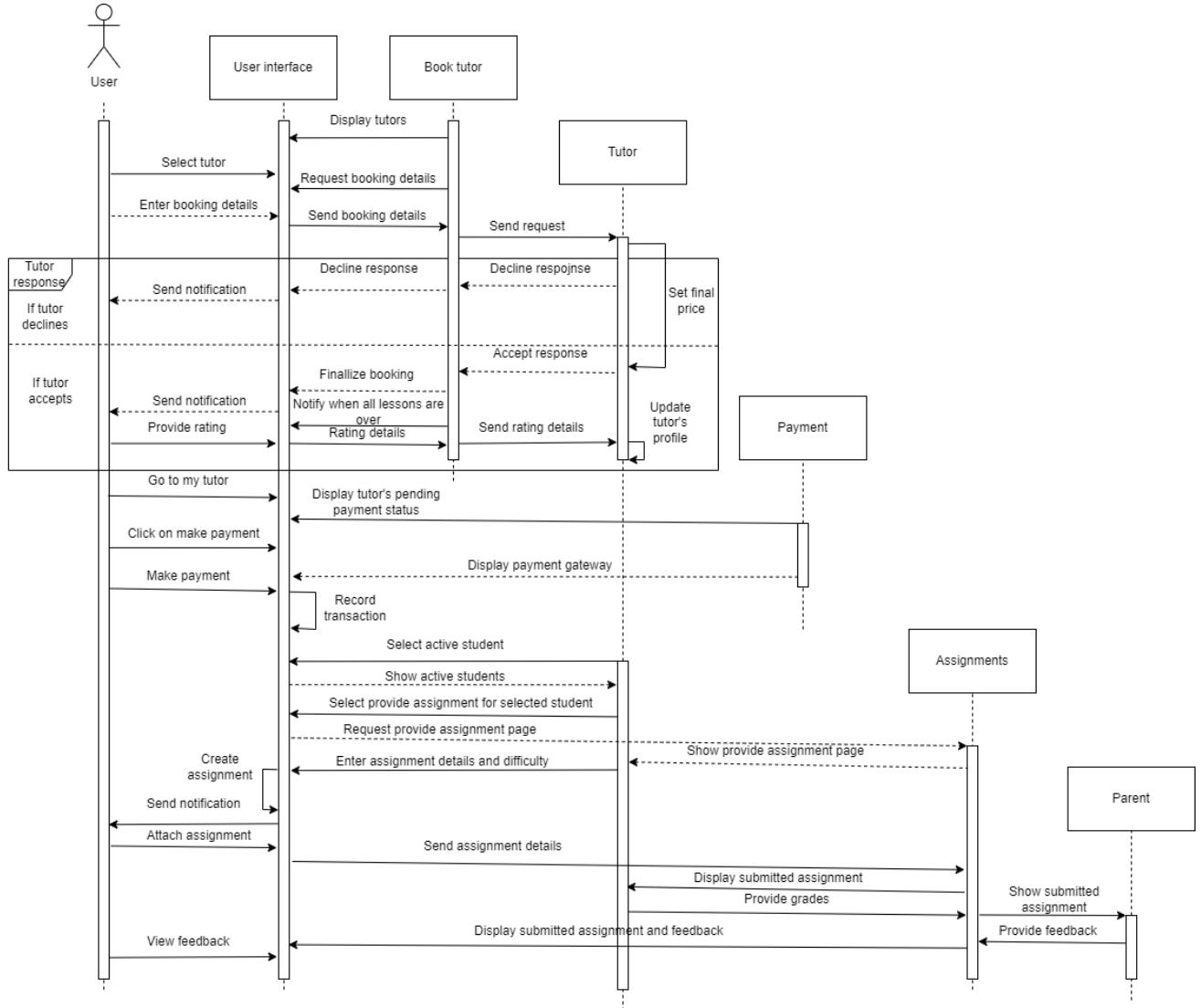


Figure 13: Sequence diagram

3.4.1 Login sequence diagram

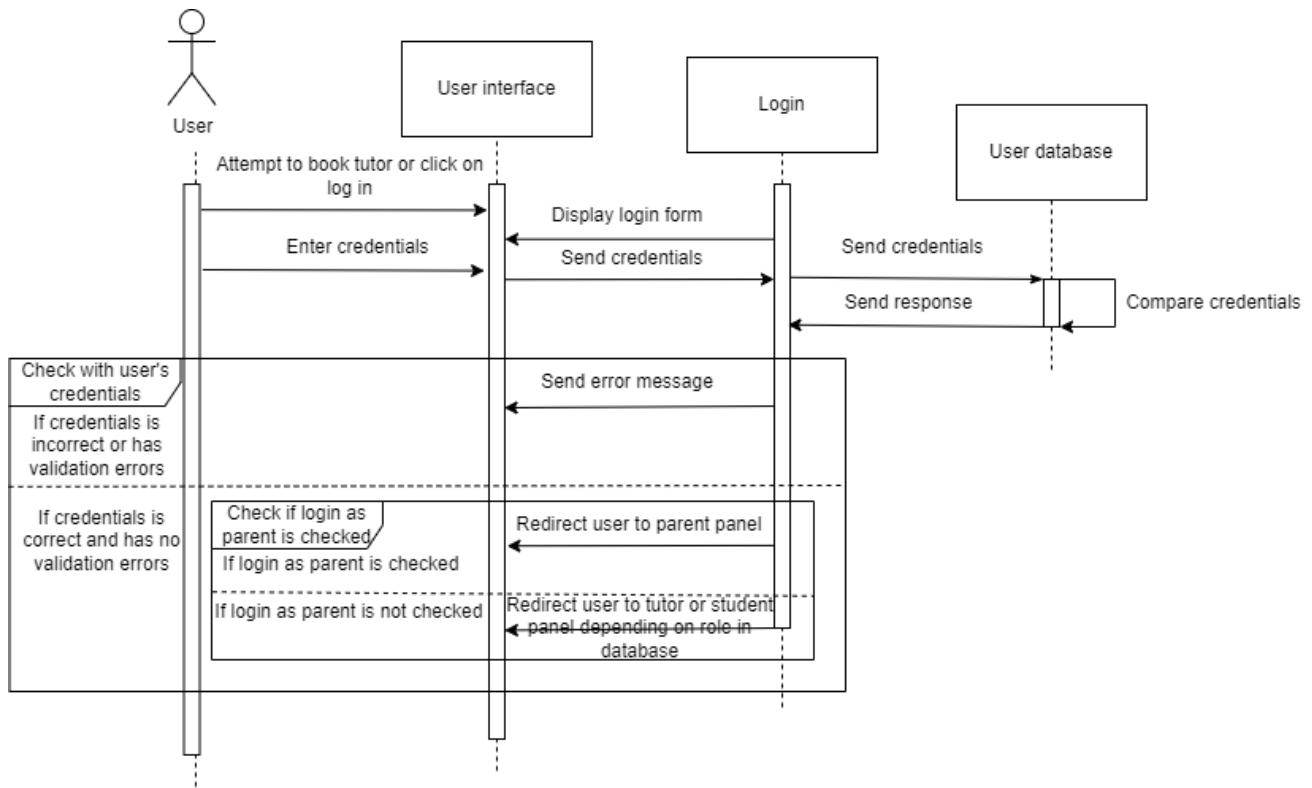


Figure 14: Sequence diagram - Login

3.4.2 Register sequence diagram

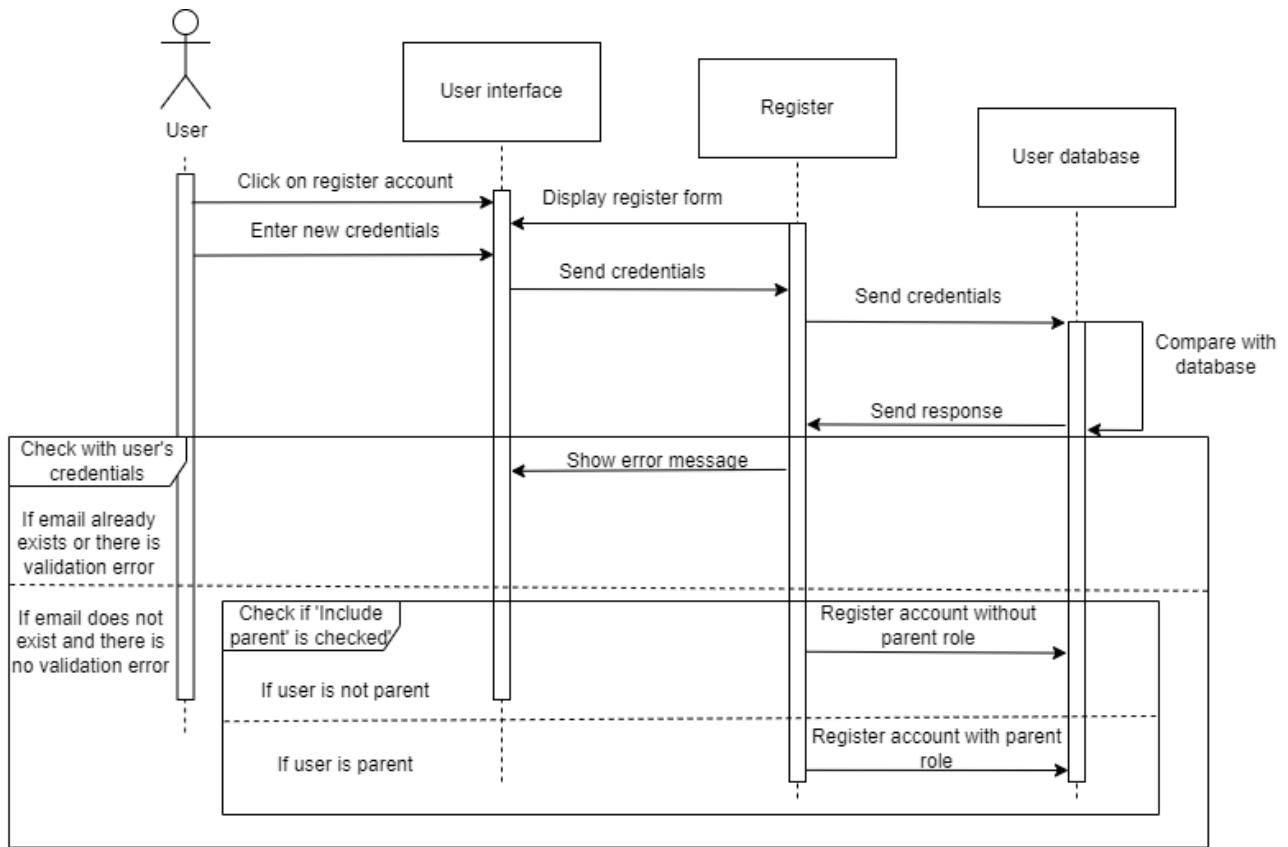


Figure 15: Sequence diagram - Register

3.5 Collaboration diagram

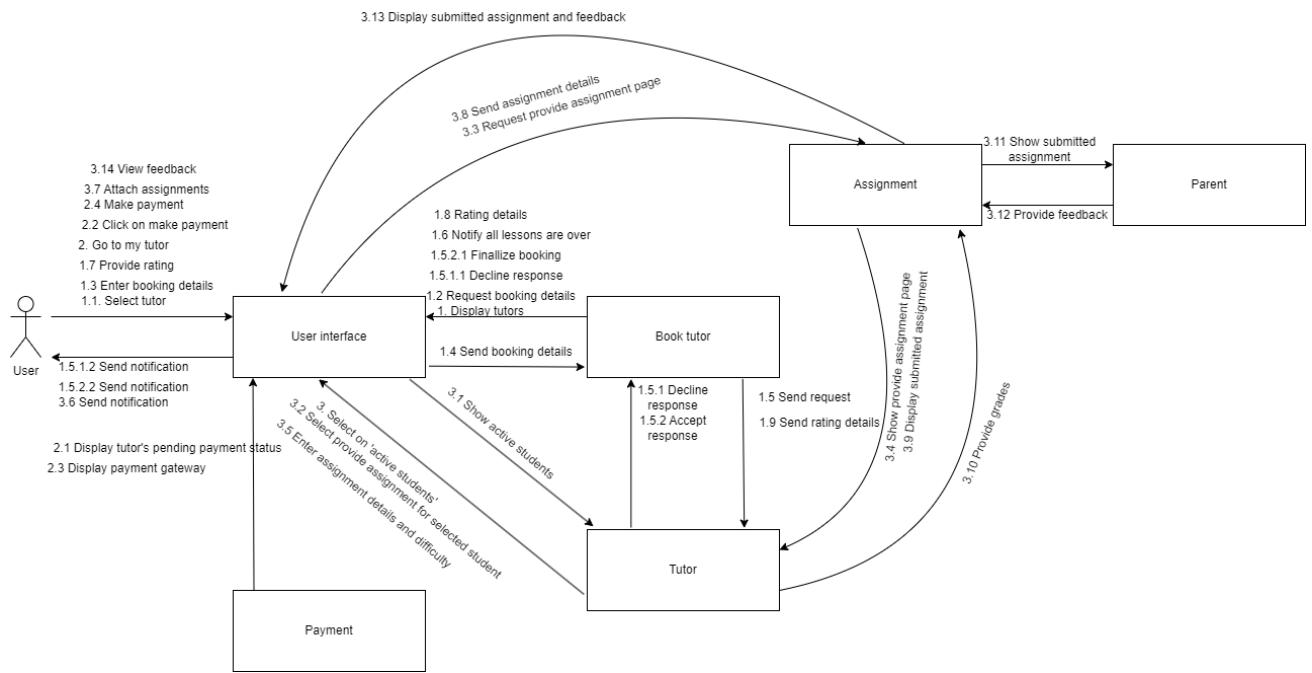


Figure 16: Collaboration diagram

3.5.1 Login collaboration diagram

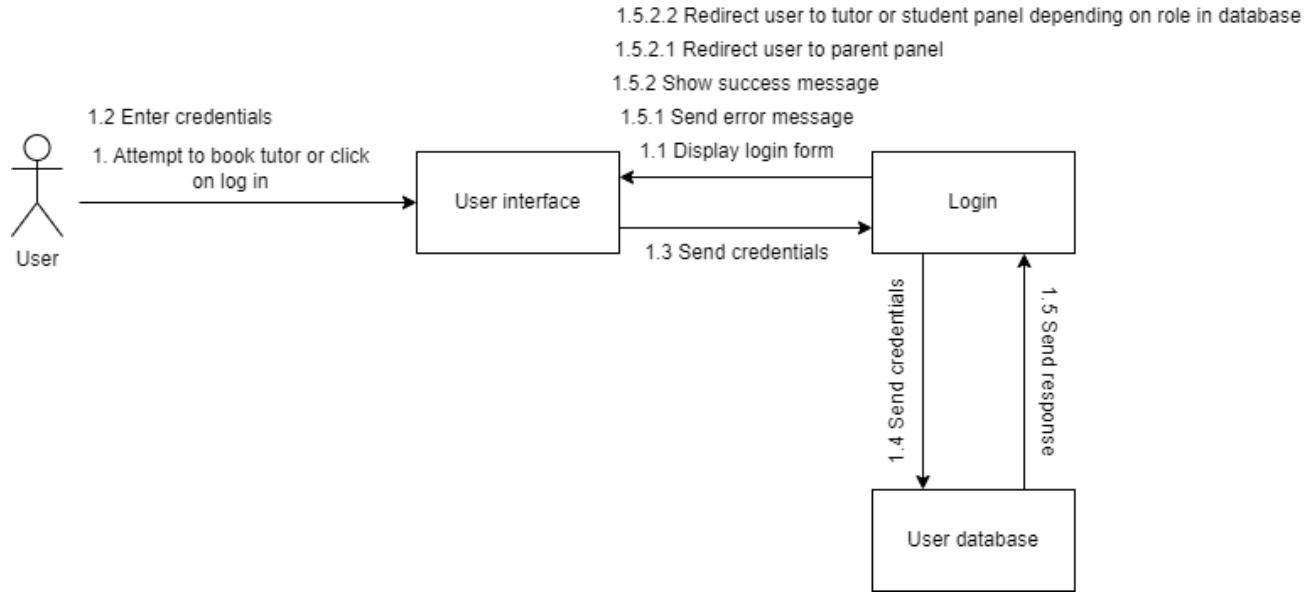


Figure 17: Collaboration diagram - Login

3.5.2 Register collaboration diagram

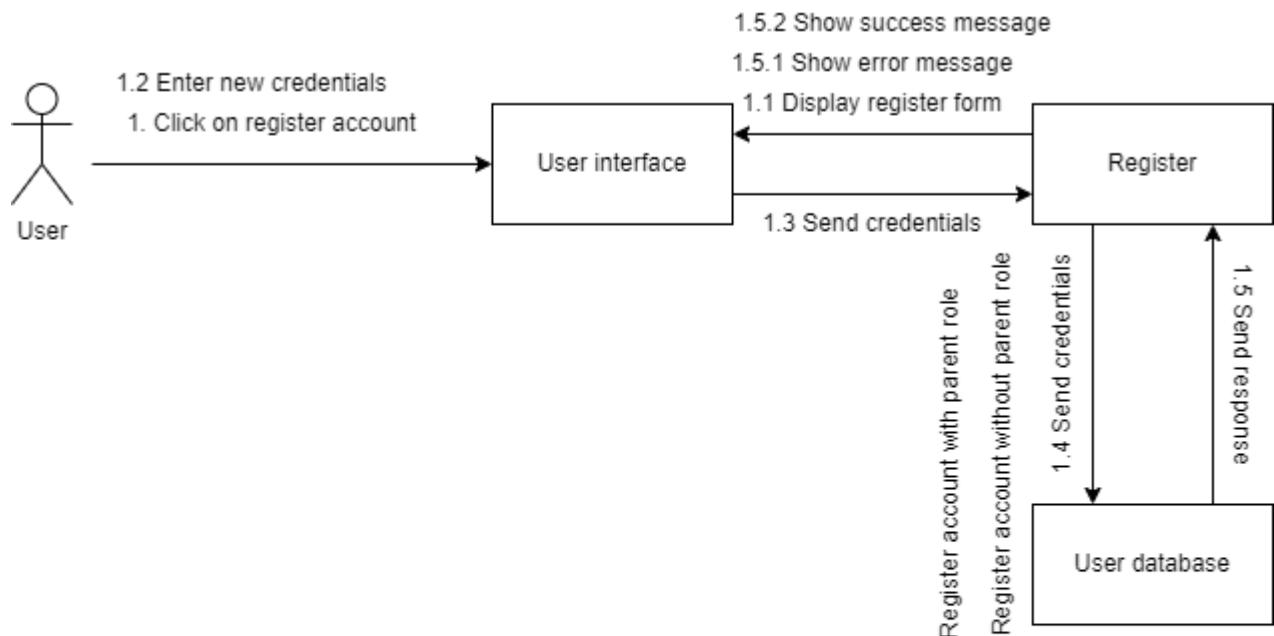


Figure 18: Collaboration diagram - Register

3.6 Data Flow Diagram

3.6.1 Context level DFD

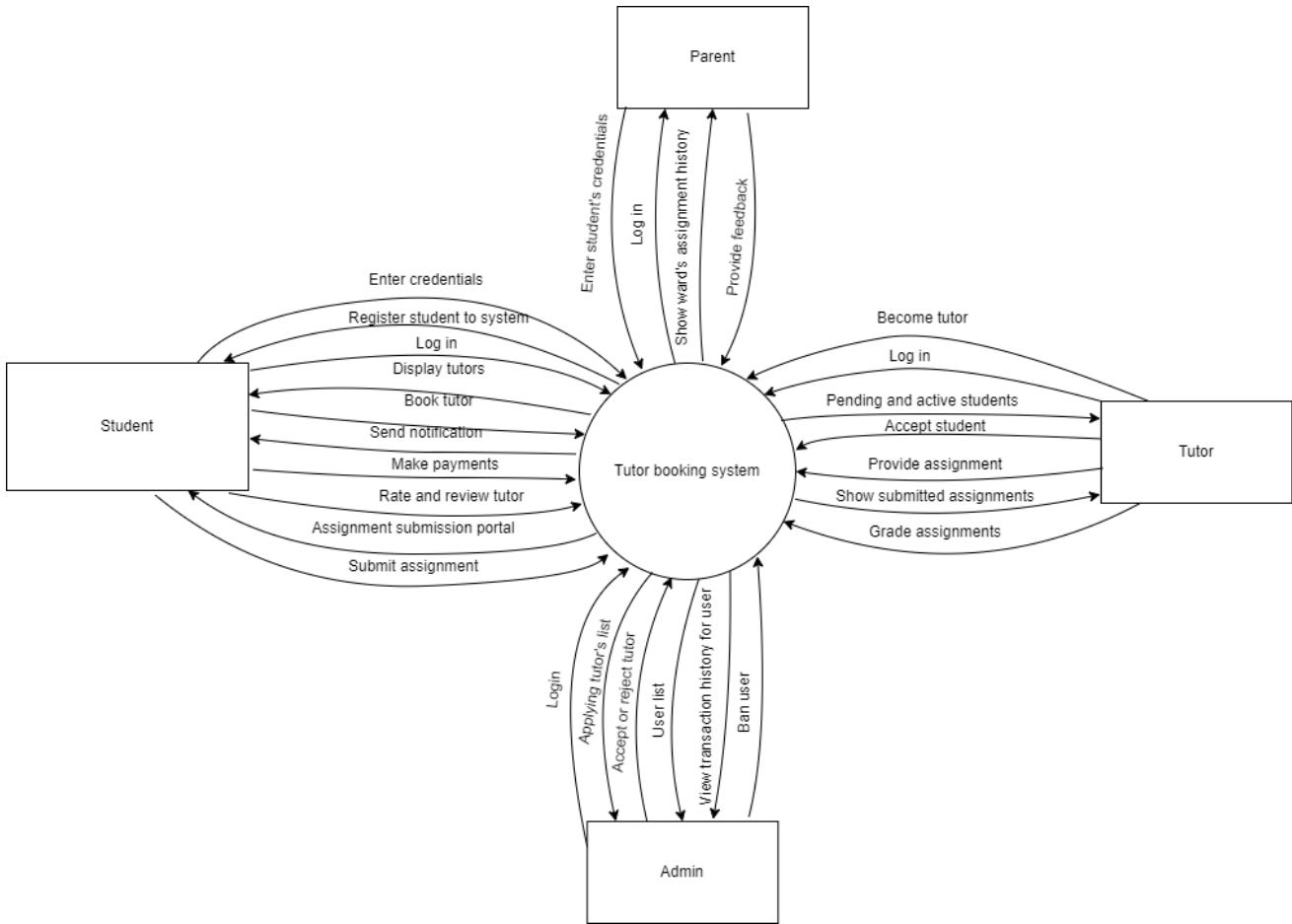


Figure 19: Data flow diagram – Context level

3.6.2 Level 1 DFD



Figure 20: Data flow Diagram - Level 1

3.6.2.1 Login – Context Level DFD

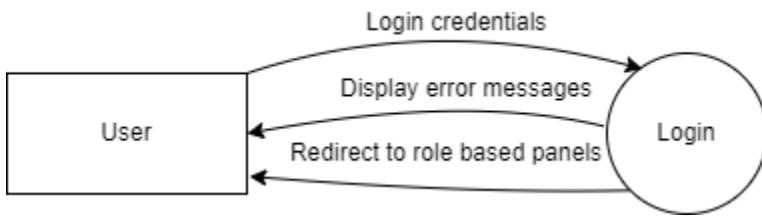


Figure 21: Context level DFD - Login

3.6.2.2 Login – Level 1 DFD

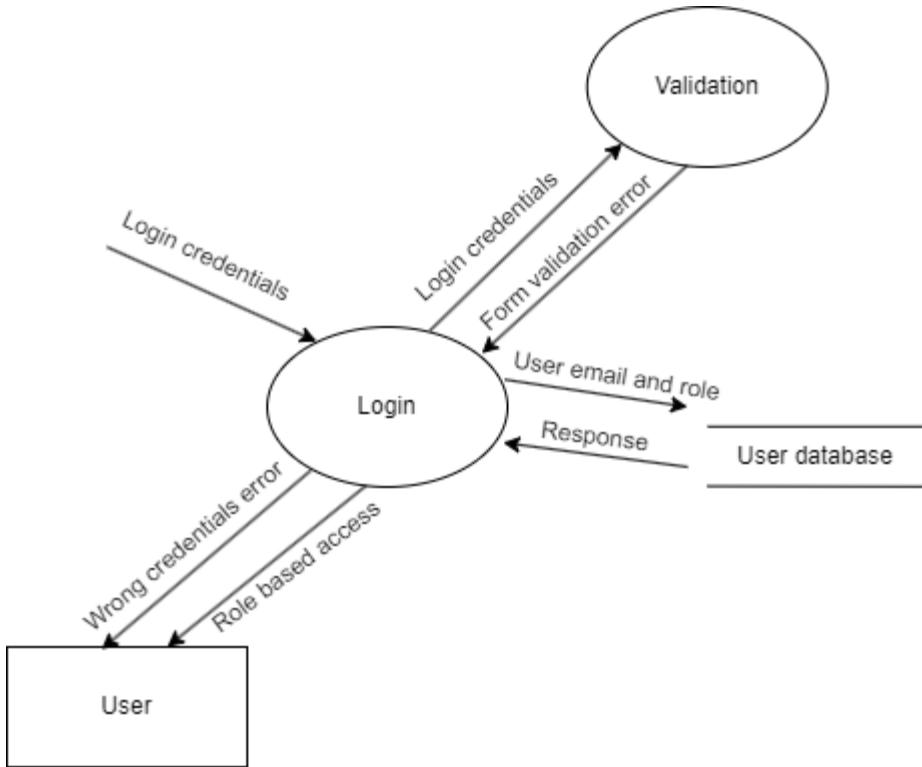


Figure 22: Level 1 DFD - Login

3.6.2.3 Register – Context Level DFD

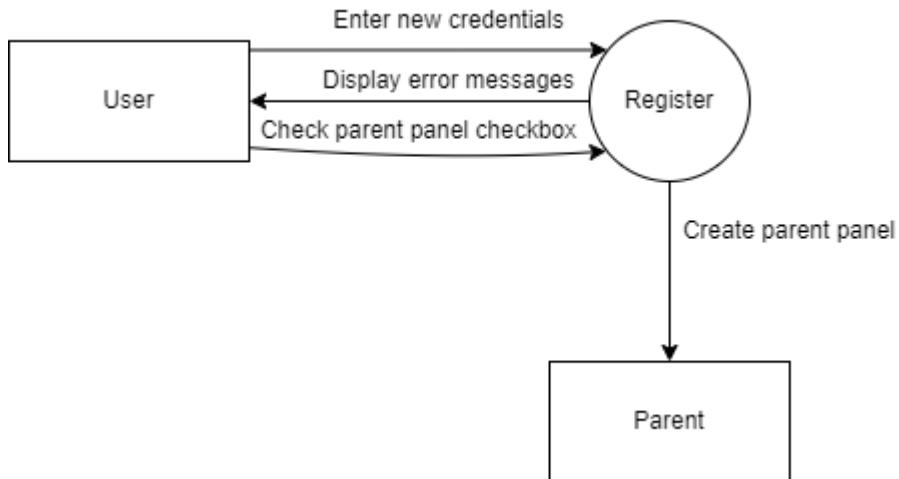


Figure 23: Context Level DFD - Register

3.6.2.4 Register – Level 1 DFD

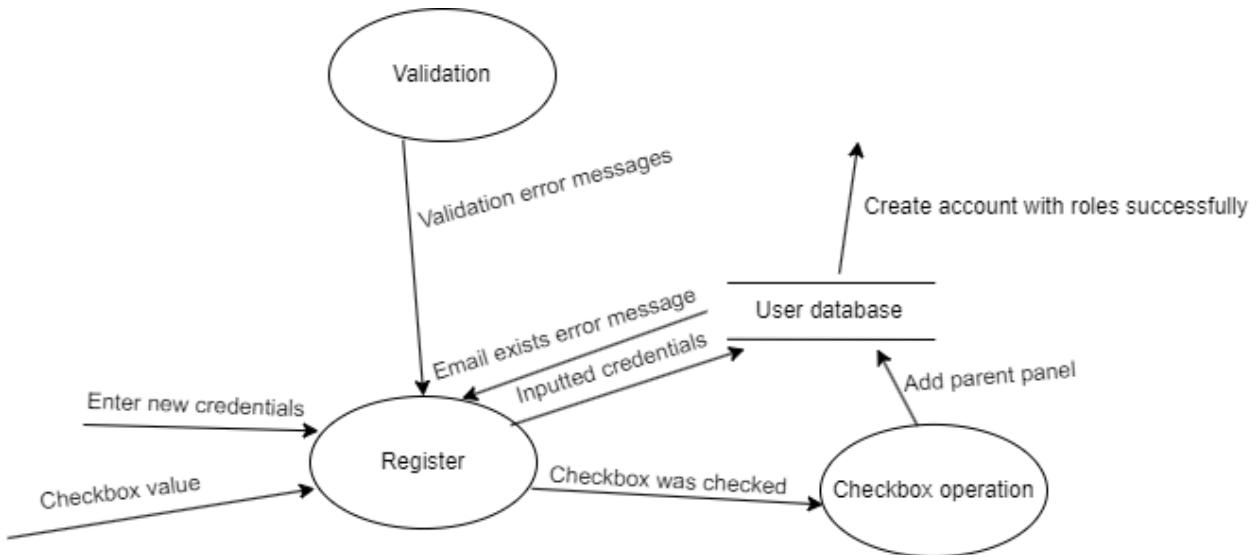


Figure 24: Level 1 DFD - Register

3.7 Mockup design

3.7.1 Login mockup

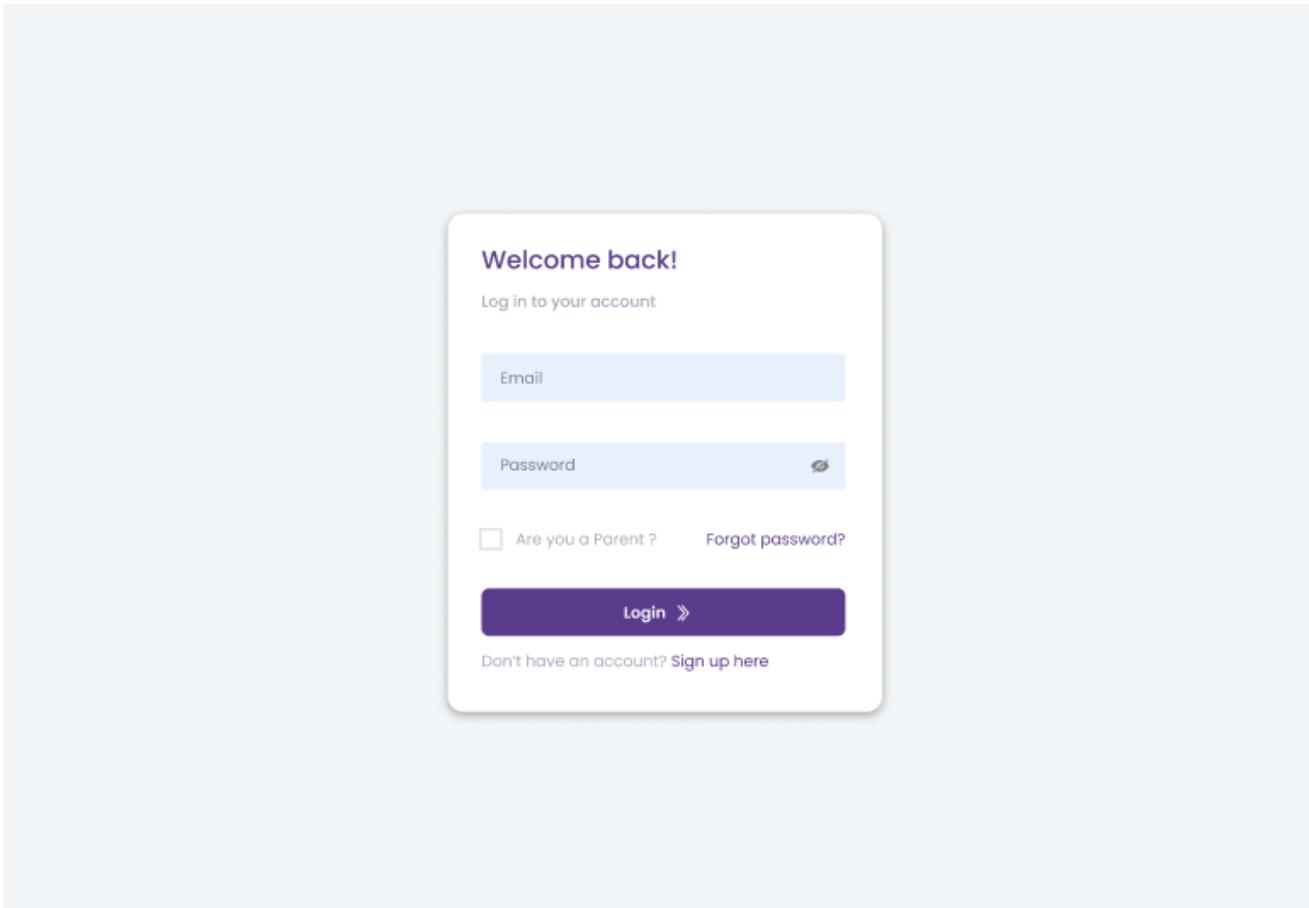
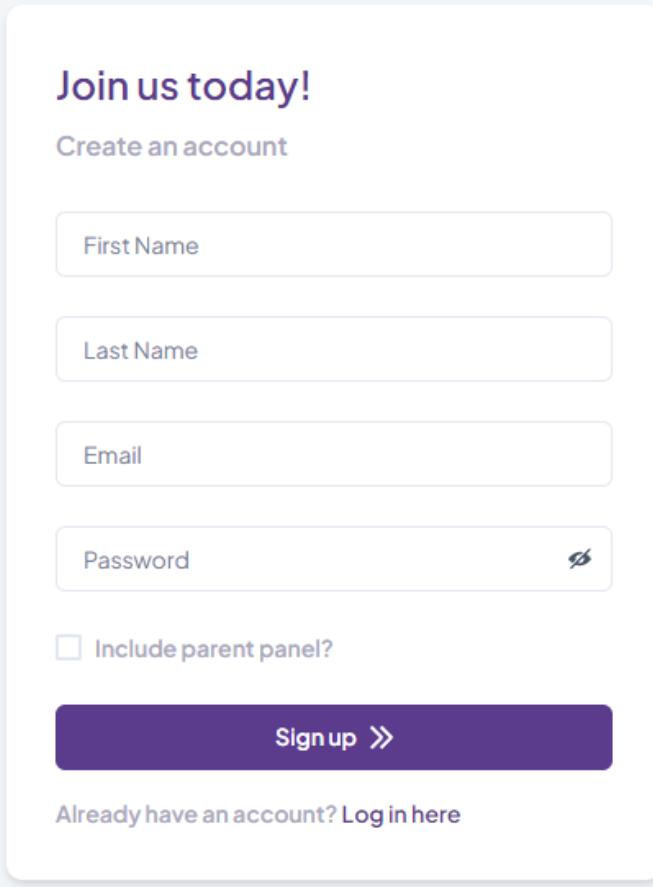


Figure 25: Login mockup

3.7.2 Register mockup



The image shows a registration form titled "Join us today!" with the subtitle "Create an account". It includes fields for First Name, Last Name, Email, and Password, along with a checkbox for "Include parent panel?" and a "Sign up" button.

Join us today!

Create an account

First Name

Last Name

Email

Password 

Include parent panel?

Sign up »

Already have an account? [Log in here](#)

Figure 26: Register mockup

3.7.3 Booking/home page mockup

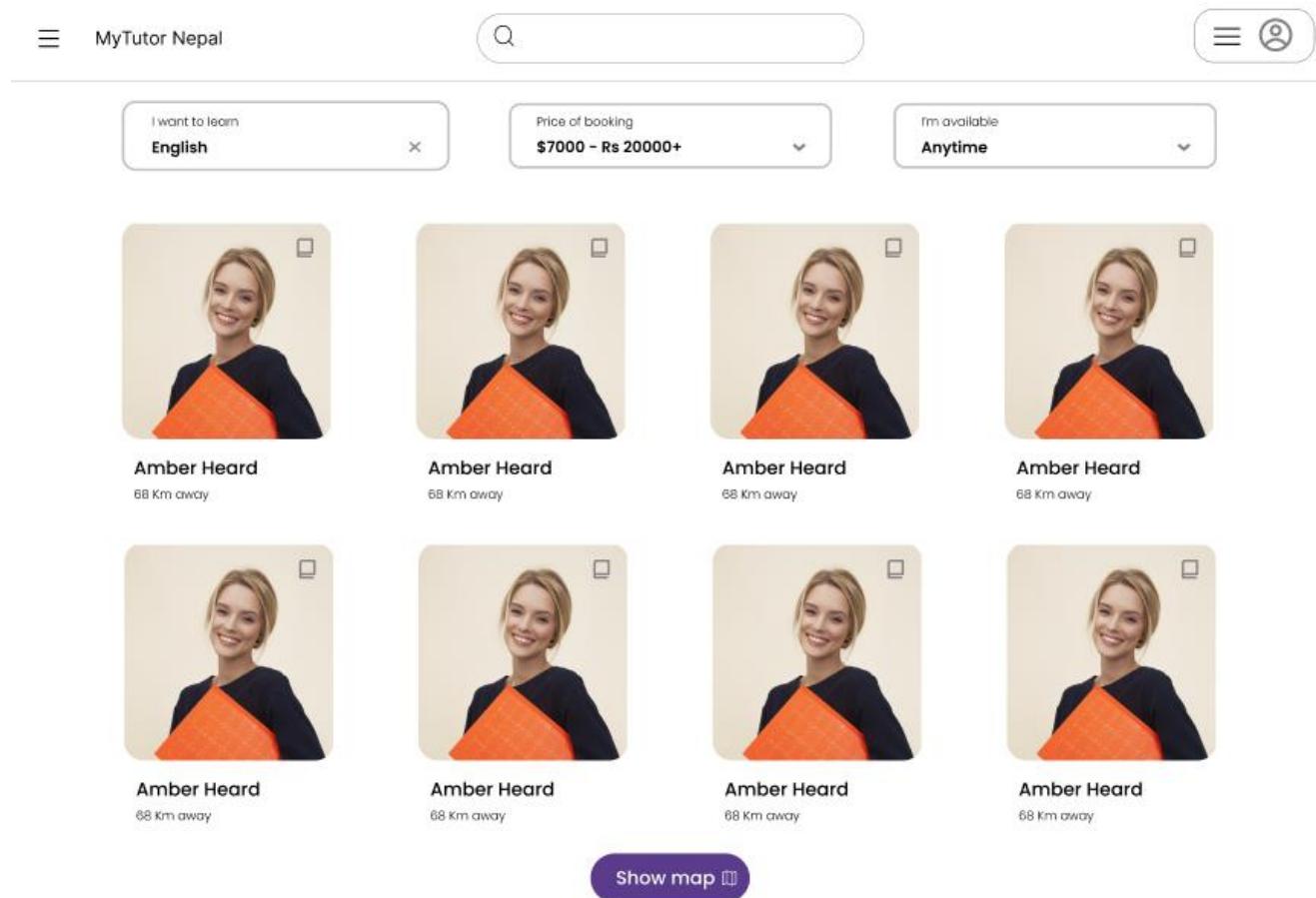


Figure 27: Home page mockup

3.7.4 Profile dropdown mockup

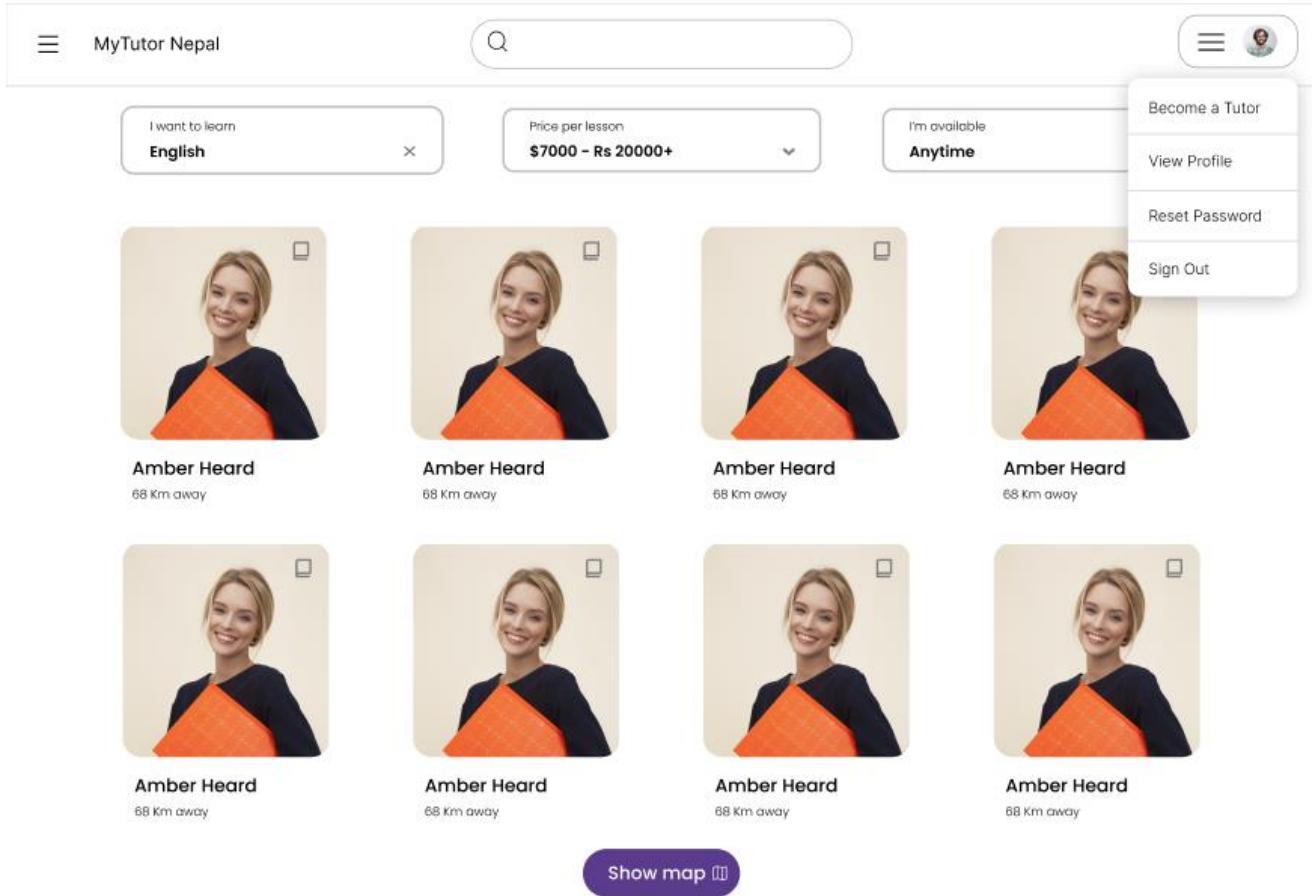


Figure 28: Profile dropdown mockup

3.7.5 View profile mockup

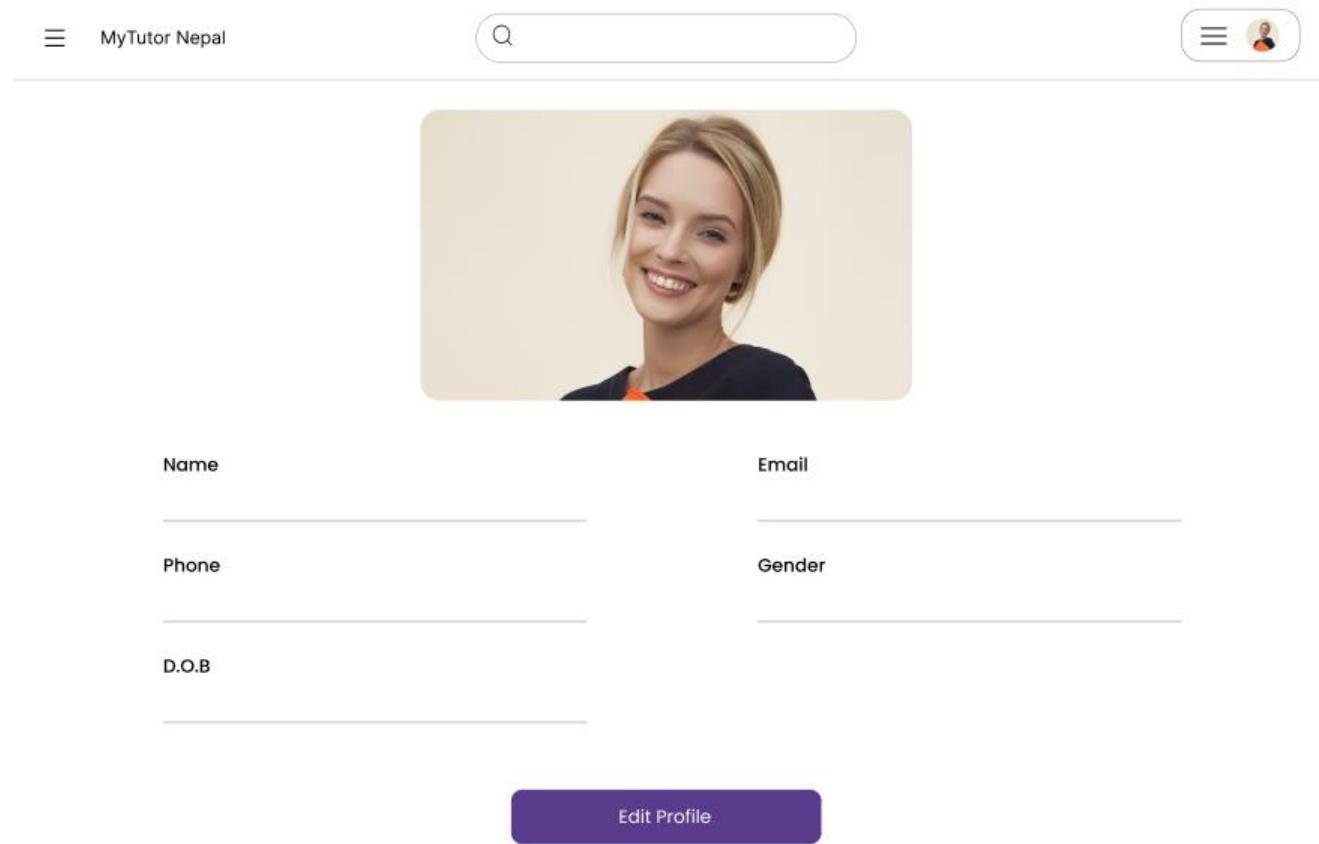


Figure 29: View profile mockup

3.7.6 Edit profile mockup

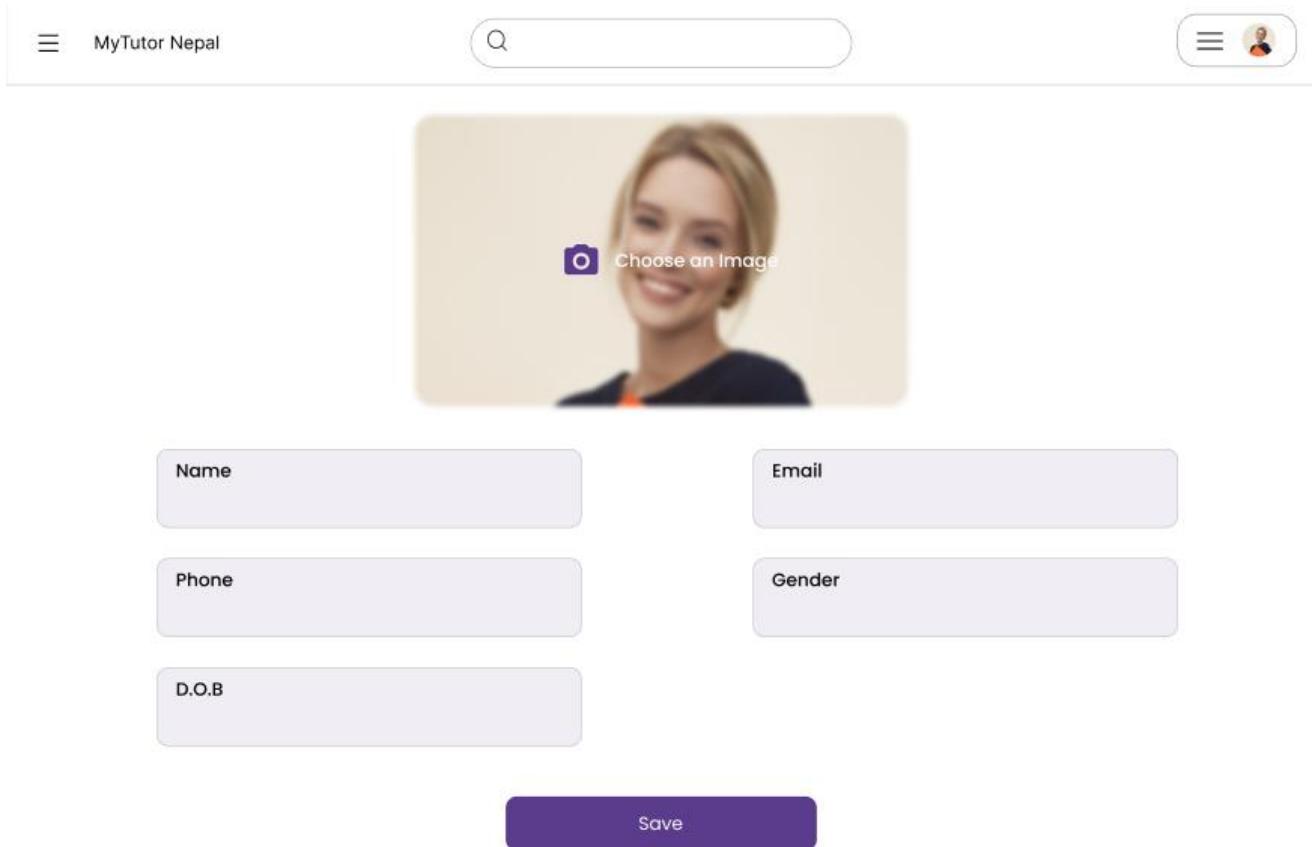


Figure 30: Edit profile mockup

3.7.7 Map view mockup

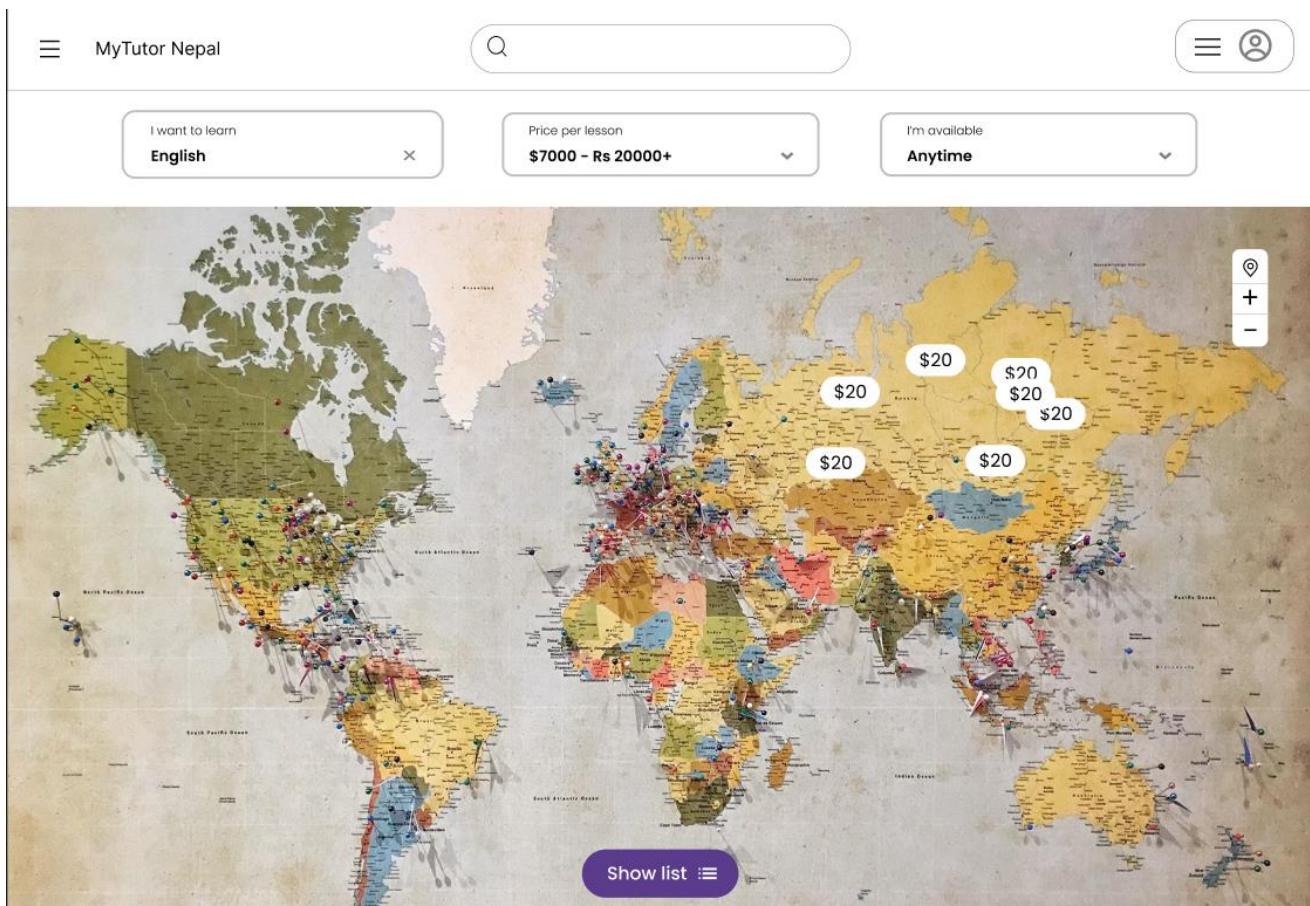


Figure 31: Map view mock up

3.7.8 Tutor mini profile in map mockup

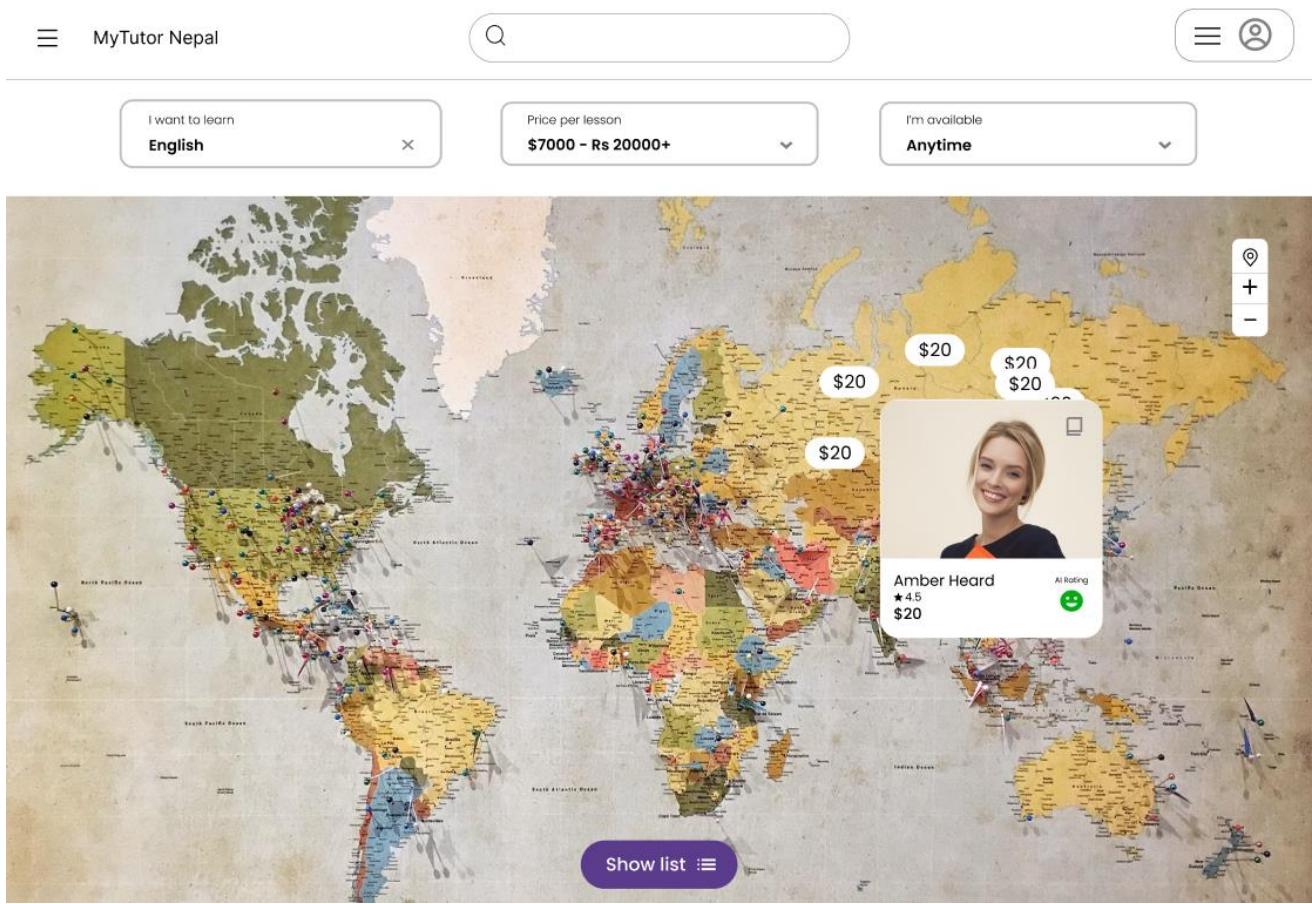
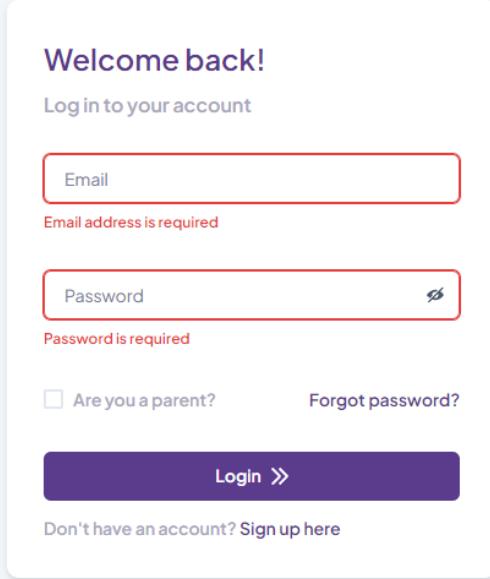


Figure 32: Tutor map mini profile mockup

3.8 Developed features

3.8.1 Login screen with validation



The image shows a developed login page with validation. The page has a light gray background and a central white rectangular form. At the top left of the form, it says "Welcome back!". Below that is a link "Log in to your account". The first input field is labeled "Email" and has a red border, indicating it is required. Below the field, the error message "Email address is required" is displayed in red. The second input field is labeled "Password" and also has a red border. To the right of the password field is a small icon of a lock. Below the password field, the error message "Password is required" is shown in red. Underneath the input fields are two links: "Are you a parent?" with a checkbox and "Forgot password?". At the bottom of the form is a large purple button with the text "Login »". Below the button, there is a link "Don't have an account? Sign up here".

Figure 33: Developed login page with validation

3.8.2 Register screen with validation

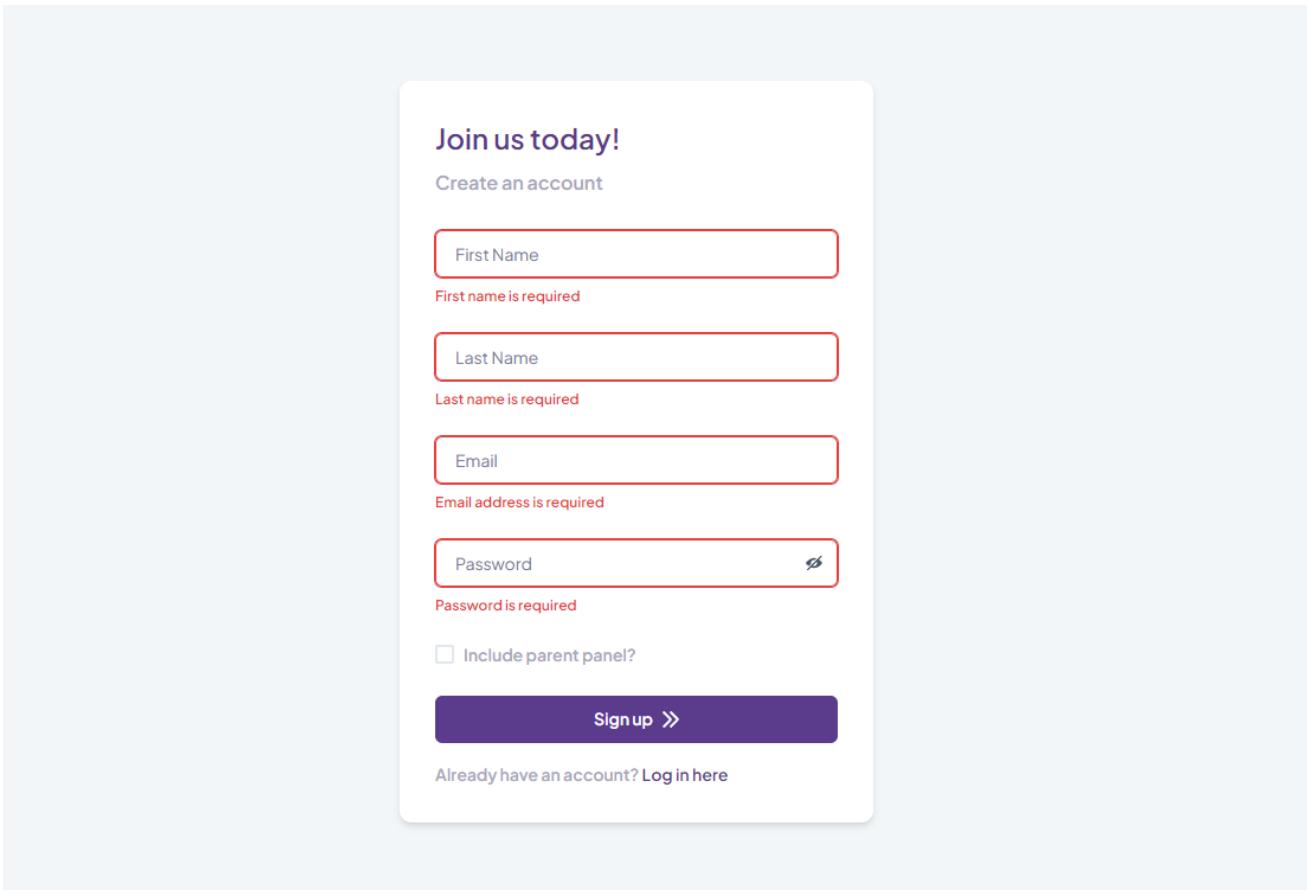


Figure 34: Developed sign up page with validation

3.8.3 Authorized access to home page

Home

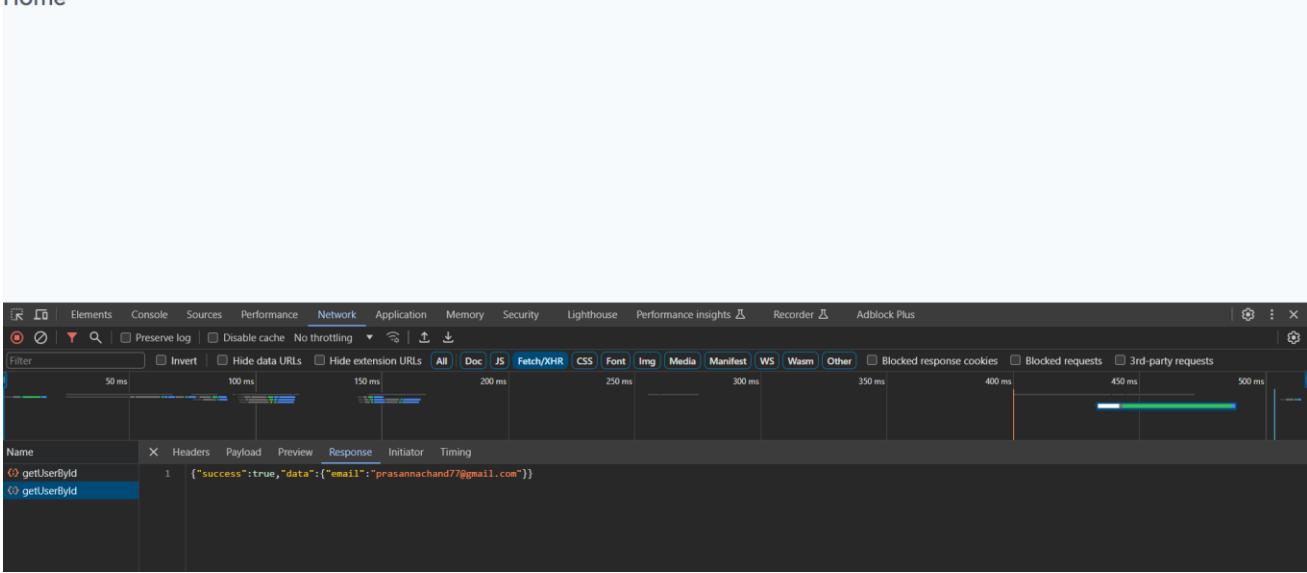


Figure 35: Authorization

4 Analysis of Progress

4.1 Progress table

SN	Tasks	Status	Progress
1.	Topic finalization	Completed	100%
2.	Technical research	In progress	80%
3.	Requirement analysis	Completed	100%
4.	Time estimation	Completed	100%
5.	Research on similar projects	Completed	100%
6.	Develop SRS document	Completed	100%
7.	Develop use case diagram	Completed	100%
8.	Develop high level and expanded use case	Completed	100%
9.	Design a mockup UI using Figma	In progress	40%
10.	Develop Entity Relationship Diagram	Completed	100%
11.	Design UML diagrams for each feature	In progress	30%
12.	Gather survey responses	Completed	100%
13.	Development of application	In progress	8%
14.	Development of authentication	In progress	50%
15.	Development of individual panel features	Not started	0%
16.	Development of booking feature	Not started	0%
17.	Development of payment feature	Not started	0%
18.	Development of assignment feature	Not started	0%
19.	Development of profile feature	Not started	0%
20.	Development of reviews and sentiment analysis		
21.	Complete testing	Not started	0%
22.	Deploy web application	Not started	0%
23.	Write final report	Not started	0%

Table 3: Progress table

4.2 Progress review

Overall, the project has made good progress in terms of documentations, research and diagrams, making a strong foundation for the development stage. Major milestones were reached in the initial stage which include topic finalization, forming a requirement analysis, estimating time in the form of Gantt charts, preparing a SRS document, gathering survey responses, and making UML diagrams for the whole system such as Use case diagram, ER diagram, Data Flow diagram, Sequence diagram, Collaboration diagram, and Activity diagram. These diagrams play a key role in each stage of prototype development. Before development begins in a prototype stage, UML diagrams and mockup designs for each feature set are prepared to provide a proper graphical illustration, simplify logic flows and make information easy to digest.

Although not fully completed, the diagrams and mockup designs are already in progress with mockup currently at 40% completion and UML diagrams at 30% completion. With the help of them, I have started developing the authentication feature for the application. Here, I have developed the process for logging in and registering to the system which covers 50% of my work for authentication. Authentication will be completed after making functionalities for forget password, reset password and logging out. Overall the progress for development is 8% which is slightly lagging behind expectations.

In terms of development, only authentication is in progress but the rest are not started. Core features like developing panels, booking, payment, assignment and profile are yet to be developed. Testing will only begin after the completion of development with deployment being the final milestone for the completion of the web application. The final part of the project will focus on report writing.

Overall, in terms of building a strong foundation for developing the web application, the project is in good health but development is lagging behind expectations. Moving forward, more focus will be provided on meeting project deadlines and balancing the process of making mockup designs, UML diagrams and development simultaneously with emphasis on pace.

4.3 Progress timeline

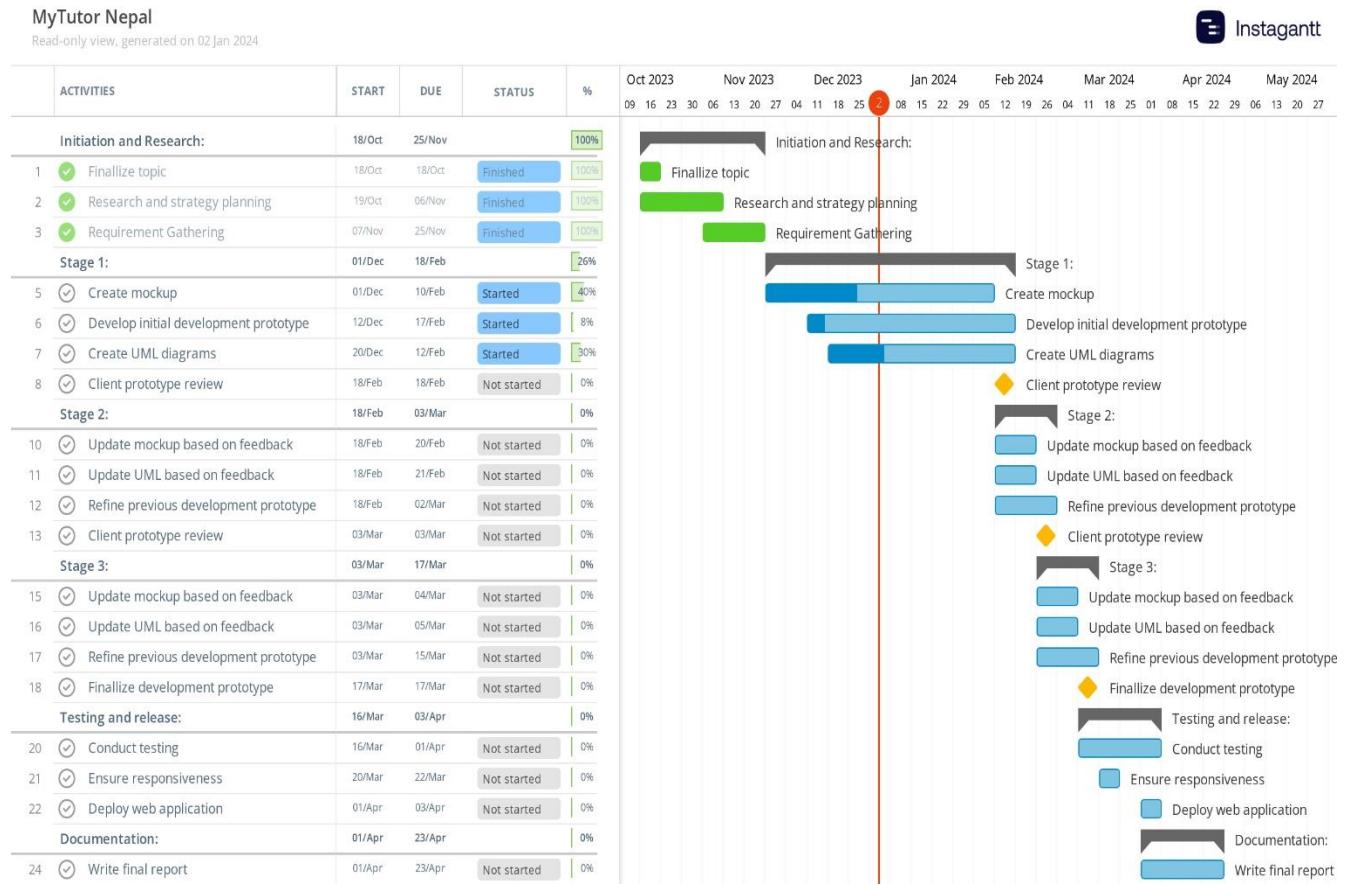


Figure 36: Revised Gantt chart

The Gantt chart submitted in the project proposal had faults due to mistakes made during research on the methodology. Therefore, a new Gantt chart was developed with new deadlines. The new Gantt chart corrects the old meaning of prototype where they were thought to be design prototypes and are now referred to as development prototypes. According to the new Gantt chart, the project's tasks are being completed on time.

4.4 Revised Milestone

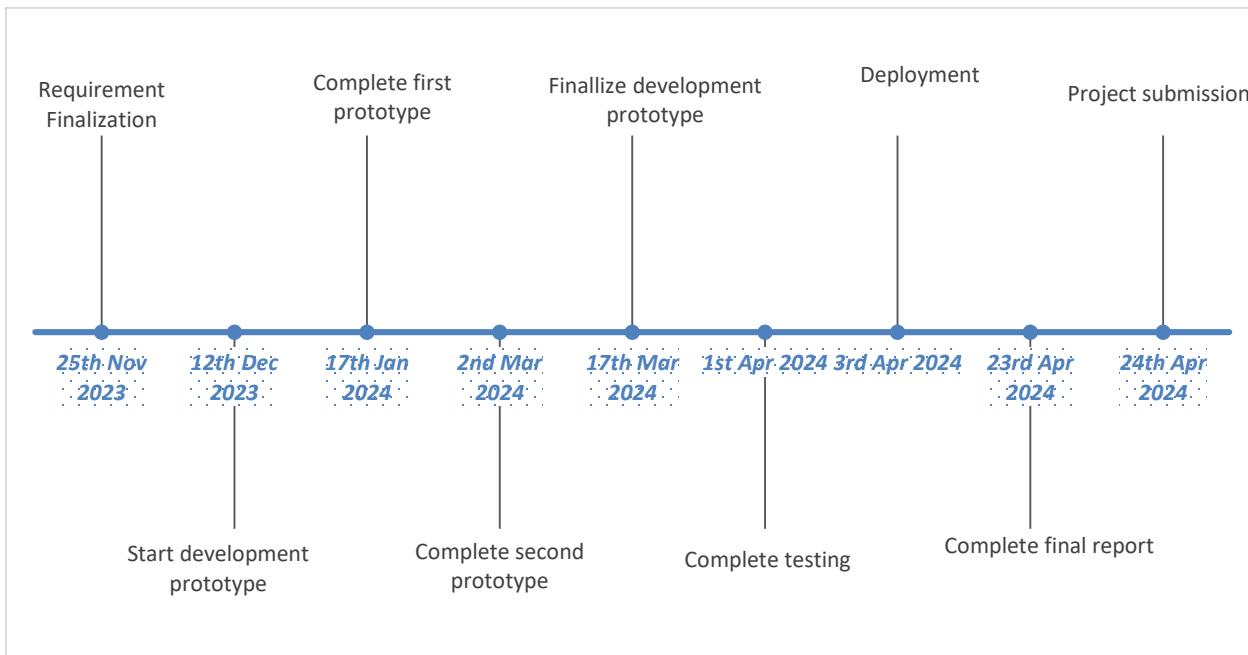


Figure 37: Revised milestone

5 Future works

With the completion of the Interim Report, I will shift my focus on developing a prototype according to the Gantt chart while maintaining the Artifact folder. Moving forward, I will have to carry out the following tasks:

- i. **Authentication** – As of writing, I have finished login and register for the authentication system. Future works include forgot password, reset password and logout. Logging out will be achieved by redirecting user to login page and deleting the JWT token which provides authorized access to resources. Forgot password and reset password will be researched before implementation.
- ii. **Role based access** – The next step will be to develop panels for each user along with respective permissions. Then, I will work on providing role based access to users to direct them to their respective panels. For admin panel, I will make an account with admin credentials and manually set the role of the user to admin in the Mongo database. The tutor will get their role when they become a tutor to create an account. Parents will have to use the same credentials as their ward's account to gain access to the parent panel.

- iii. **Accept tutor** - When a tutor registers their account after submitting details a post method will be sent to the backend. When the request hits an API endpoint, the backend logic will run and display applying tutor to the admin where they can accept or reject the tutor. When accepted, a tutor table is formed in the database.
- iv. **Tutor list and map view** - The home page will show the list of accepted tutors in a grid format. Users can either filter out tutors or can use the map view feature (Figure 31) where each tutor is displayed by their location. This feature will be developed using a library called Mapbox.
- v. **Notification feature** – For the notification feature, the model for each user will consist of a notification attribute which is an array. Whenever an action is performed that invokes a notification, an API endpoint is hit which runs a controller in the backend. The logic involves pushing the notification that is sent from the frontend into the array. Hence the notification will show whatever is inside the notification array in the database for that user.
- vi. **Profile feature** – The profile feature simple feature where whatever information was inputted during registration will be saved to the database and that information will be displayed on the user's profile page. For tutors, this information will be shown to the user when booking. Each user can edit their profile and when saving changes, a put request is fired to the backend which updates the database.
- vii. **Rate tutor** – For this feature, the tutor will have an attribute in the database for review which will be an array. Each rating and review made for the tutor will be stored in this array. This information will be updated to the tutor's profile.
- viii. **Payment and Sentiment analysis** – These features have not been researched enough due to the time spent on researching other features. They will be researched in the coming weeks of the first prototype stage, before there is a need to implement them
- ix. **Booking and Assignment feature** – These topics have been surfacely explored and there is a basic understanding for how the feature should work. The client and server side logics have not been formed in depth and will be done after implementation of previous features.

- x. **Testing** – After completion of the development of the application, testing needs to be completed. I will form several test cases to see if the system contains any error. Any error that arises will be fixed before deployment
- xi. **Final report** – The final report will be prepared after testing is complete.

6 Appendix

6.1 Survey results

50 people responded to the survey. The feedbacks provided by users will be implemented in my project. Here are the results of the survey.

1. Have you used any application/website to book a tutor?

50 responses

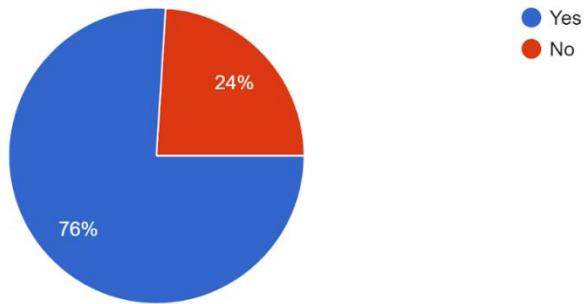


Figure 38: Have you used any application/website to book a tutor?

2. Was the hiring process for these application well managed and simplified?

47 responses

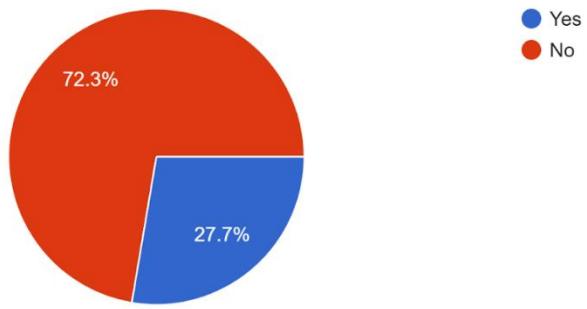


Figure 39: Was the hiring process for these application well managed and simplified?

3. Select some features that the application had.

46 responses

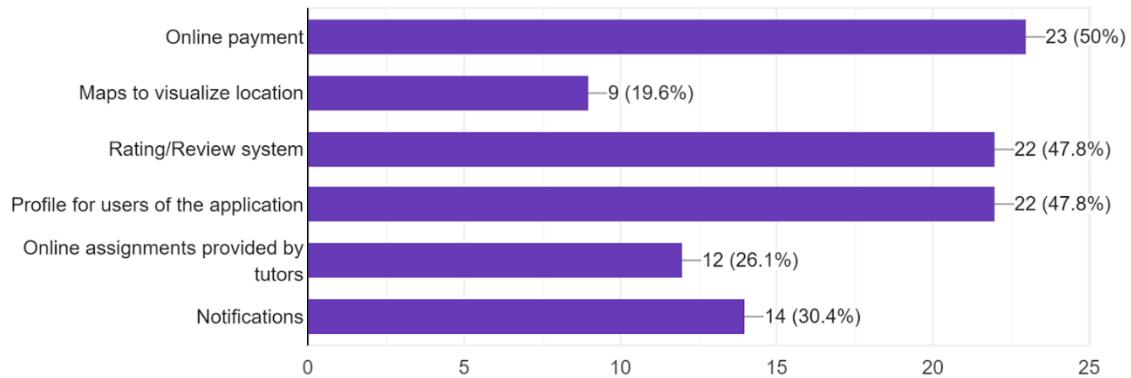


Figure 40: Select some features that the application had.

4. For what reasons have you had to book a tutor?

47 responses

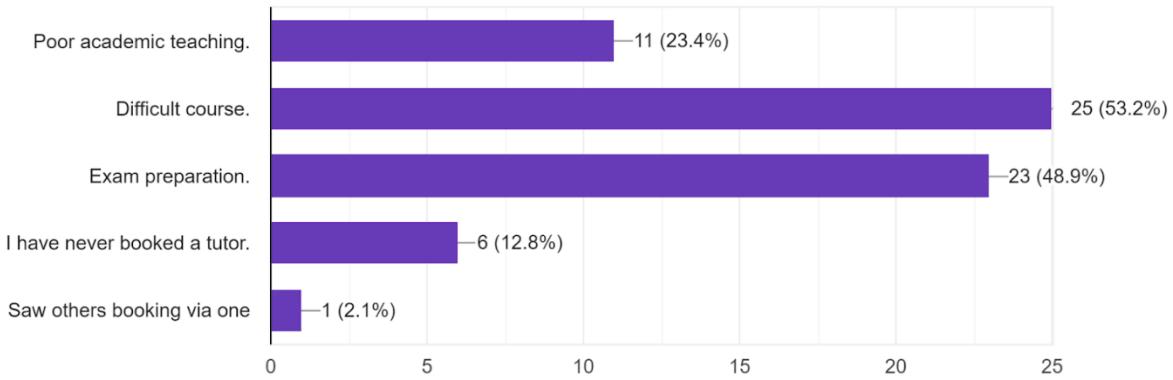


Figure 41: For what reasons have you had to book a tutor?

5. What do you look for when booking a tutor?

50 responses

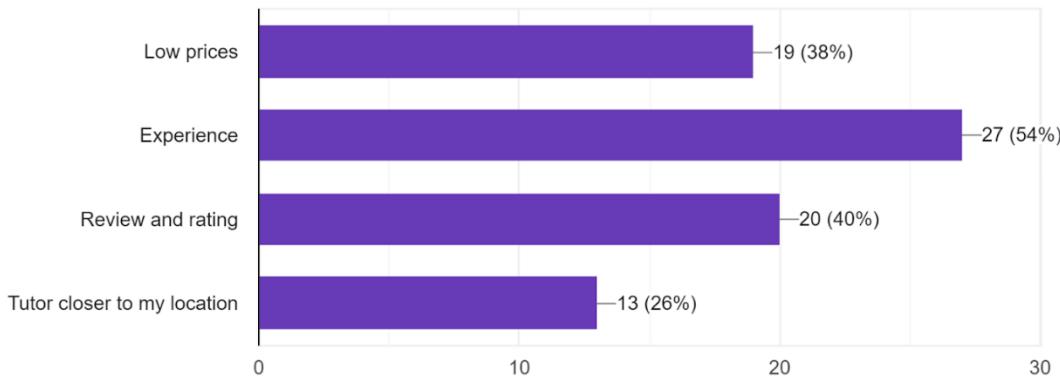


Figure 42: What do you look for when booking a tutor?

6. Is the tutor's location a major concern for you?

50 responses

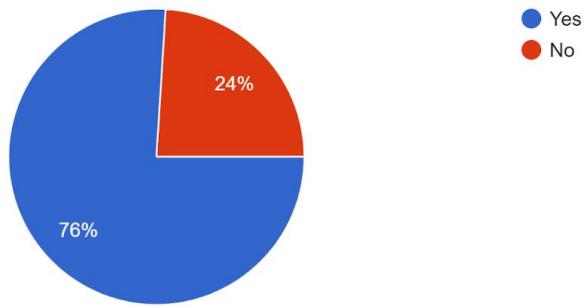


Figure 43: Is the tutor's location a major concern for you?

7. Would you be more likely to book a tutor closer to your location?

50 responses

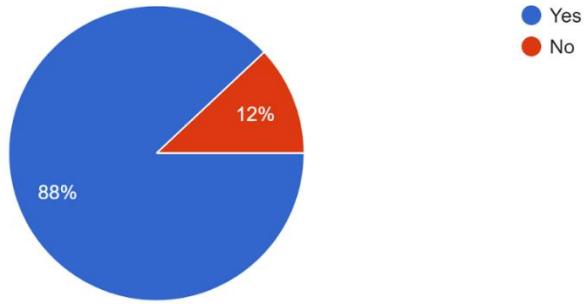


Figure 44: Would you be more likely to book a tutor closer to your location?

8. Is the tutor's authenticity a major concern when using this application?

50 responses

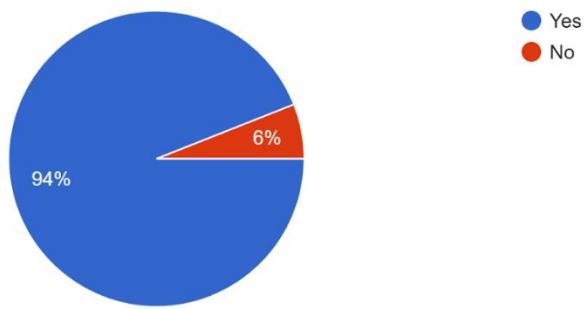


Figure 45: Is the tutor's authenticity a major concern when using this application?

9. What measures should be taken by an application to validate the authenticity of tutors?

50 responses



Figure 46: What measures should be taken by an application to validate the authenticity of tutors?

10. What documents must a tutor submit for qualifications and authenticity to be approved as a registered tutor?

50 responses

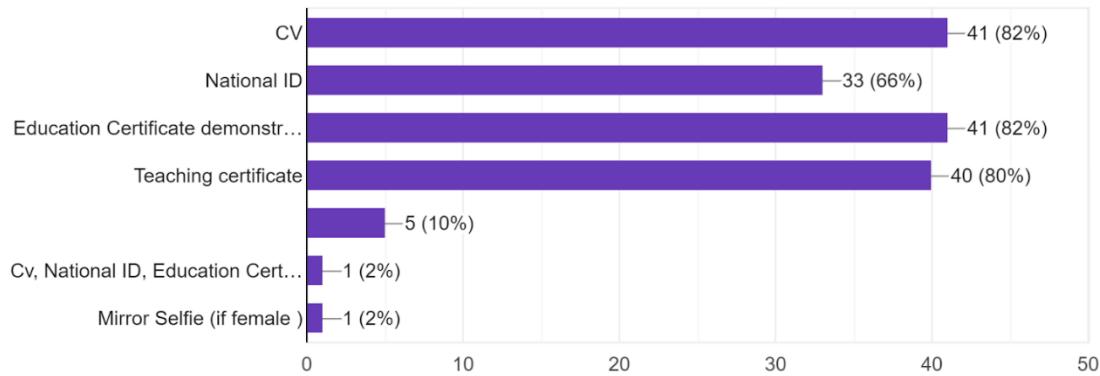


Figure 47: What documents must a tutor submit for qualifications and authenticity to be approved as a registered tutor?

11. Should there be a panel for parents to view pending and completed assignments by their children particularly for students in primary education?

50 responses

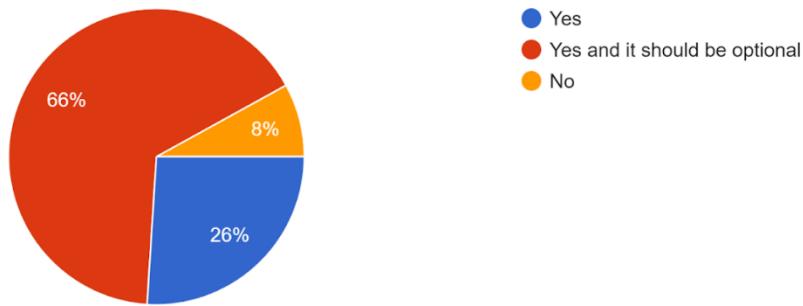


Figure 48: Should there be a panel for parents to view pending and completed assignments by their children particularly for students in primary education?

12. Should users of the application receive announcements about major updates?

50 responses

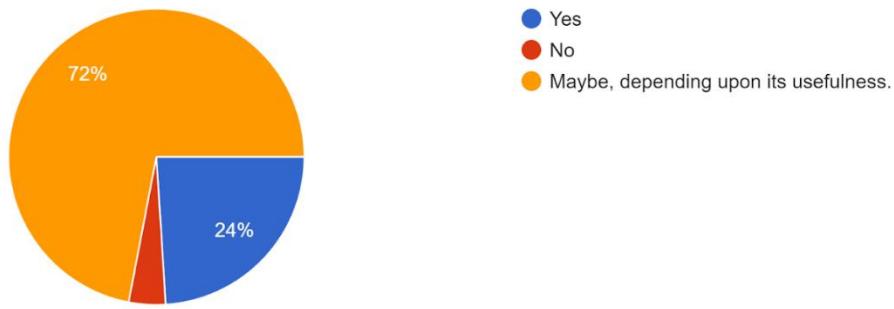


Figure 49: Should users of the application receive announcements about major updates?

6.2 Project introduction

By using this application, one will be able to filter out tutors according to their preferred time. The application will offer a map based interface where one can book tutors using a map box that shows all tutors by their location. Additionally, the application will have a review and rating system, enhanced by artificial intelligence which helps students find the best tutor for them. The application will make it easier to find a tutor for the right price which will be displayed for each tutor. Payment will be done through the application although there will be an option to make payment in person. Students will not have to worry about the teacher's legitimacy as all tutors will go through the process of submitting documents when registering as one. The application will also have a feature allowing tutors to give assignments to students and grade them. Parents of students will be allowed to view pending and completed assignments of the students through the parent panel and leave remarks.

6.3 Understanding the project

- **Figma** - Figma is a versatile web designing tool that allows creation of interactive user interfaces and prototypes collaboratively. (What is Figma?, n.d.) I will be using Figma as it provides a clean way to visualize a mockup design of the system.
- **Javascript** - JavaScript is a programming language widely used to make web pages interactive. It will be used primarily throughout the development of the application as it allows me to use powerful frontend libraries like React and backend frameworks like Express.

- **ReactJS** – ReactJS is one of the most popular JavaScript libraries. React makes use of components, which are just functions that contain HTML and JavaScript (known as jsx) to make the code more reusable. This feature allows me to save time and make the code more maintainable.
- **Chakra UI** – Chakra is a component library built into React that provides pre-defined but customizable styles for HTML elements. I will be using Chakra by overriding the default theme provided by it to make the frontend UI as it provides a more efficient way to style HTML elements without writing additional CSS.
- **NodeJS** – NodeJS is a JavaScript runtime environment which will serve as the foundation for backend development in the application. It will be used to handle requests, manage databases and executing server-side logics.
- **ExpressJS** – ExpressJS is a framework of NodeJS which will be used in my application for routing, running middleware, handling HTTP requests, etc.
- **MongoDB** – MongoDB is a non-relational database that uses JSON-like format to store data and develop scalable applications. I will make use of MongoDB to make various models that will be required to perform queries on the database collections.

6.4 Use case diagram

6.4.1 High level use case

- i. **Use case:** Sign up

Actors: User, Parent, Admin

Description: A user fills their necessary information and clicks on sign up. If the user has selected the checkbox to include parents, a parent panel will be created which can be logged in to with the same credentials. If the email does not exist, a new account is registered to the database.

- ii. **Use case:** Log in

Actors: User, Parent, Tutor, Admin

Description: The user inputs their credentials and clicks on log in. Depending on their role, they are directed to their own panel. If parent panel was not created during sign up, it will show an error message.

- iii. **Use case:** View profile

Actors: User, Parent, Tutor

Description: Users of the application have their own personalized profile which will be created when signing up. The profile can be viewed and edited by user, parents and tutors.

- iv. **Use case:** Become tutor

Actors: User, Admin

Description: An unregistered user clicks on become tutor button. The person is required to fill necessary information including login credentials. The tutor submits identification and qualification documents and a notification is sent to admin. If no same email exists, a tutor account will be created upon request acceptance by admin.

- v. **Use case:** Book tutor

Actors: User, Tutor

Description: The user selects a tutor and clicks on book. After selecting book time and checking availability, the user is allowed to book the tutor which sends a request to the tutor. The tutor can accept or reject the request. If accepted, a notification will be sent to the user and booking is successful.

- vi. **Use case:** Make payments

Actors: User, Tutor

Description: When booking is successful, user can make payments until the trial period is over which lasts three days. If not paid, the booking will be cancelled. Payment can be done via Khalti or in person.

- vii. **Use case:** Rate tutor

Actors: User, Tutor

Description: After the lessons are over, the user has the option to rate the tutor and provide a review. After rating is done, the tutor's public profile gets updated and the review can be viewed by the tutor.

- viii. **Use case:** Receive assignment

Actors: User, Tutor

Description: The tutor creates an assignment and the user receives a notification.

ix. **Use case:** Submit assignment

Actors: User

Description: The user can submit the assignment by attaching a file. Assignments have deadline before which the file can only be submitted.

x. **Use case:** View assignment

Actors: User, Parent, Tutor

Description: Each assignment can be viewed by the user, parent and tutor. This contains information about deadline, grades, remarks and pending status. Feedback can be left on assignments by parents, users and tutors.

xi. **Use case:** View students

Actors: Tutor

Description: Tutors have the ability to view students in their panel. Here, they can accept or reject pending students. When accepted, the students are moved to active students list.

xii. **Use case:** Perform administration activities

Actors: Admin

Description: In the admin panel, admins can perform administration activities such as accepting tutor requests, banning users, view users and tutors list, and viewing financial history for each user.

6.4.2 Expanded use case

- i. **Use case:** Sign up

Actors: User, Parent, Admin

Description: A user fills their necessary information and clicks on sign up. If the user has selected the checkbox to include parents, a parent panel will be created which can be logged in to with the same credentials. If the email does not exist, a new account is registered to the database.

Typical course of events

Actor action	System response
The user fills up sign up form and presses 'Sign up'.	
	The system sends an error message if email exists in database.
	The system creates an account if email does not exists in database.
	The system creates a parent account with the same credentials if specified checkbox is checked.

Table 4: Sign up expanded use case

ii. **Use case:** Log in

Actors: User, Parent, Tutor, Admin

Description: The user inputs their credentials and clicks on log in. Depending on their role, they are directed to their own panel. If parent panel was not created during sign up, it will show an error message.

Typical course of events

Actor action	System response
User clicks on log in on home page or tries to book a tutor	
	System redirects user to login page.
The user fills up log in form and presses 'Log in'.	
	The system sends validation message if there is any error.
	The system directs user to their respective panels depending on their role.

Table 5: Log in expanded use case

iii. **Use case:** View profile

Actors: User, Parent, Tutor

Description: Users of the application have their own personalized profile which will be created when signing up. The profile can be viewed and edited by user, parents and tutors.

Typical course of events

Actor action	System response
User clicks on 'View profile'.	
	System shows profile with user's information.
User clicks on 'Edit profile'.	
	System allows user to edit input fields.
User changes profile information and clicks on 'Save'.	
	System updates user's information to database.

Table 6: View profile expanded use case

iv. **Use case:** Become tutor

Actors: User, Admin

Description: A user clicks on become tutor button. The person is required to fill necessary information including login credentials. The tutor submits identification and qualification documents and a notification is sent to admin. If no same email exists, a tutor account will be created upon request acceptance by admin.

Typical course of events

Actor action	System response
User clicks on 'Become tutor' button.	
	System logs out user if logged in.
	System asks user to fill up a form which also includes sign up credentials.
User fills up form and submits.	
	System sends a request to admin.
Admin accepts or rejects tutor.	
	If accepted, tutor account is created.

Table 7: Become tutor expanded use case

v. **Use case:** Book tutor

Actors: User, Tutor

Description: The user selects a tutor and clicks on book. After selecting book time and checking availability, the user is allowed to book the tutor which sends a request to the tutor. The tutor can accept or reject the request. If accepted, a notification will be sent to the user and booking is successful.

Typical course of events

Actor action	System response
User clicks on a tutor.	
	System opens tutor's profile displaying information about the tutor.
User clicks on 'Book this tutor' button.	
User enters time and clicks on 'Book now' button.	
	System checks if tutor is available at specified time.
	If available, tutor gets a request.
Tutor can adjust final price and accept.	
	System sends final price to user.
If user accepts, booking is successful.	

Table 8: Book a tutor expanded use case

vi. **Use case:** Make payments

Actors: User, Tutor

Description: When booking is successful, user can make payments until the trial period is over which lasts three days. If not paid, the booking will be cancelled. Payment can be done via Khalti or in person.

Typical course of events

Actor action	System response
	System shows remaining time on payment period.
User selects ‘My tutor’ and clicks on make payment.	
User chooses pay with Khalti.	
	System shows Khalti payment gateway.
User makes payment.	
	System prepares a report and finalizes booking by creating a booking table.
User chooses pay in person.	
	System prepares a report and finalizes booking by creating a booking table.

Table 9: Make payments expanded use case

vii. **Use case:** Rate tutor

Actors: User, Tutor

Description: After the lessons are over, the user has the option to rate the tutor and provide a review. After rating is done, the tutor's public profile gets updated and the review can be viewed by the tutor.

Typical course of events

Actor action	System response
	System ends the tutoring lessons after specified time.
User gets a modal for rating a reviewing tutor.	
User provides rating and review.	
	System updates the rating and review on tutor's profile.

Table 10: Rate tutor expanded use case

viii. **Use case:** Receive assignment

Actors: User, Tutor

Description: The tutor creates an assignment and the user receives a notification.

Typical course of events

Actor action	System response
Tutor creates assignment after inputting details.	
	System sends notification to user.

Table 11: Receive assignment expanded use case

xiii. **Use case:** Submit assignment

Actors: User

Description: The user can submit the assignment by attaching a file. Assignments have deadline before which the file can only be submitted.

Typical course of events

Actor action	System response
User goes to ‘My assignments’ and selects ‘Pending assignments’.	
	System shows pending assignments to user.
User clicks on ‘Submit assignment’.	
User attaches a file and submits.	

Table 12: Submit assignment expanded use case

ix. **Use case:** View assignment

Actors: User, Parent, Tutor

Description: Each assignment can be viewed by the user, parent and tutor. This contains information about deadline, grades, remarks and pending status. Feedback can be left on assignments by parents, users and tutors.

Typical course of events

Actor action	System response
User clicks on 'My assignments'.	
	System shows pending and submitted assignments.
Parents click on 'My ward's assignments'.	
	System shows pending and submitted assignments.
Tutor clicks on 'Active students'.	
	System shows assignment submitted for each student.
Tutor leaves remarks and grades on assignment	
	System notifies parents and users.
Users, parents and tutor leave feedback on assignment.	

Table 13: View assignment expanded use case

x. **Use case:** View students

Actors: Tutor

Description: Tutors have the ability to view students in their panel. Here, they can accept or reject pending students. When accepted, the students are moved to active students list.

Typical course of events

Actor action	System response
Tutor clicks on 'My students'.	
	System shows a list of pending and active students.
Tutor clicks on 'Pending students'.	
	System shows pending students list.
Tutor clicks on accept or reject.	
	System notifies rejected students.
	System adds accepted students to active students list.

Table 14: View students expanded use case

xi. **Use case:** Perform administration activities.

Actors: Admin

Description: In the admin panel, admins can perform administration activities such as accepting tutor requests, banning users, view users and tutors list, and viewing financial history for each user.

Typical course of events

Actor action	System response
Admin clicks on ‘Tutor requests’.	
	System shows a list of pending tutors waiting to be approved.
Tutor accepts or rejects tutor.	
	System adds accepted tutors to the database.
Tutor selects ‘Users’ or ‘Tutors’.	
Tutor selects view financial history for a user.	
	System shows financial history
Tutor select ‘Ban user’ for a user	
	System deactivates user’s account, showing a banned message when they try to log in.

Table 15: Perform administration activities expanded use case

6.5 Considered methodologies

6.5.1 Scrum methodology

Scrum method is a part of the Agile methodology in which a project is divided into multiple sprints that last two to four weeks and each sprint involves completing a task from the product backlog. The product backlog contains a list of all tasks with their priorities sorted out. Daily meeting is conducted where the team shares their updates on tasks done. At the end of each sprint, the results are reviewed. (List of Scrum Advantages and Disadvantages | Indeed.com, 2023) The principle of scrum is to deliver the software quickly with less focus on documentation. Although good for the context of my project, it was not selected for the following reasons:

- This methodology works best with teams as each sprint requires collaborative effort to finish any given task on time.
- Each sprint requires intensive work to meet the deadline but tasks may not always go to plan when working alone.

6.5.2 Waterfall methodology

In waterfall methodology, each phase has to be completed linearly before the next phase can begin. This methodology is mostly used for small scale projects where requirements are clear and there are not any constant changes. The disadvantages of using this methodology are:

- It is risky because it has limited flexibility for changes. Alterations in requirements is difficult once project is underway.
- Working software is not produced until later in the life cycle which means there is no room for user input and feedbacks.
- It is a poor model for complex and long going medium and large scale projects
- Testing and validation is done during the end of the project which means bugs and other issues are not detected until later in the project.

(SDLC - Waterfall Model, n.d.)

6.6 Researched projects

6.6.1 mySecondTeacher

MySecondTeacher is an online academic delivery platform that provides interactive educational videos, custom test papers, and a virtual classroom feature where teachers can give assignments to students. My system will have a similar feature where teachers can assign tasks to students and the students can submit their work in image format. These assignments will then be graded by the teacher and the parent can leave their feedback.

The screenshot shows a web application interface for managing assignments. At the top, there are two dropdown menus: 'CS5002 Software Engineer' and 'L2C16_22_CO_AU_CS500'. Below these are three navigation tabs: 'Teacher's Sessions', 'Assignments' (which is currently selected), and 'Teacher's Content'. A button 'View All Upcoming Deadlines' is located in the top right corner. To the right of the tabs, there are options to 'View as: Cards' or 'List'. The main content area is divided into two sections: 'Open Assignments (0)' and 'Closed Assignments (1)'. The 'Closed Assignments' section contains one item named 'test', which was assigned on March 1, 2023, at 10:00 AM and has a deadline of March 3, 2023, at 10:00 AM. The status is 'Normal'. Below the assignment details, it says 'Missed' and 'Not yet graded'. A 'Full Report' button is available for this assignment. In the bottom right corner of the content area, there is a 'Go to Top ↑' link.

Figure 50: mySecondTeacher assignment page:

6.6.2 TutorMate Nepal

TutorMate Nepal is an online learning platform that allows students to request a tutor based on the needs that are specified by students. The tutors in this application are approved by the admins. Tutors are only allowed to teach after they submit qualification and identification documents. The algorithm of this web application makes it possible for approved tutors to not be listed and shown to users but only be recommended to students after they specify their needs such as level, postcode, location and subject. The price of tutors is based on hourly pay and teaching can take place either online or in person. Similar to mine, this application consists of a student, parent, admin and tutor panel.



Figure 51: Finding a tutor

5 Tutors found

0 REPEAT STUDENTS **TRAVEL RADIUS** **5Mi**

BOOK ME

SAGAR RAJ, KATHMANDU

Mathematics (A Level)
Physics (EDXcel)
Physics- Science Faculty (Higher Secondary Education (Nepal +2))
Mathematics - Science Faculty (Higher Secondary Education (Nepal +2))
Grade 7 - Maths (A Level, Higher Secondary Education (Nepal +2))

[View More](#)

DISCOUNT OFFER : 20%

Rs.750/Hr

Hi, I am a mathematics & Science tutor. I am passionate and have a positive attitude with which i ai... [Read More](#)

CALLBACK TIME **HOURS TAUGHT**

0 REPEAT STUDENTS **TRAVEL RADIUS** **2Mi**

BOOK ME

JASHMIN, NA

Grade 3 - Maths (Advanced, Beginner, Intermediate)
Grade 3 - Computer (Advanced, Beginner, Intermediate)
Grade 3 - Social Studies (Advanced, Beginner, Intermediate)
Grade 3 - Social Studies (Advanced, Beginner, Intermediate)
Grade 3 - Nepali (Advanced, Beginner, Intermediate)

[View More](#)

DISCOUNT OFFER : 10%

Rs.1000/Hr

Hi, I am an accounts tutor. I am passionate and have a positive attitude with which i aim to encoura... [Read More](#)

CALLBACK TIME **HOURS TAUGHT**

0 REPEAT STUDENTS **TRAVEL RADIUS** **0Mi**

BOOK ME

RAMESH, KATHMANDU

Accounting (EDXcel)
Accounting (CIE AS Level)
Accounting - Management Faculty (Higher Secondary Education (Nepal +2))
English (SEE)

DISCOUNT OFFER : 10%

Rs.1000/Hr

dynamic and dedicated business graduated

CALLBACK TIME **HOURS TAUGHT**

Figure 52: Tutors recommended to me based on preferences

6.7 Software Requirement Specifications

6.7.1 Introduction

6.7.1.1 Purpose

The purpose of this SRS document is to describe what the tutor booking system will do and how it will perform. It outlines the functional and non-functional requirements of my application. This document will serve as a vital reference for clients, developers and testers involved in the development and evaluation process. The SRS will define the project's scope, boundaries and explore in depth how the system should work functionally so that requirements are met accordingly. Overall, this SRS will act as a foundational guide, providing clear understanding of the project's goals.

6.7.1.2 Intended Audience

This SRS document is intended for developers, testers and clients. This document will give the readers an overview of the functional and non-functional requirements. Developers can refer to this document to view requirements of the product. Testers can refer to determine that the requirements have been met. Client can also view this document in order to see the system requirements, high priority use cases and so on.

6.7.1.3 Project scope

The project's scope is to deliver an online web application that connects students to tutors, allowing them to book a preferred tutor. To facilitate this experience, students can leave ratings and feedbacks for tutors to improve other user's experience and AI review for each tutor will also be provided which will be based on reviews from other users. Additionally, the application must also include a three-day trial period so that users are satisfied with the tutors. A map will be used to display tutors to the users as an alternate view so that they can locate tutors closer to their location. When booking is done by the user, the tutors will receive a request and can adjust their prices and accept the request. Khalti will be integrated for payment. Admins will have the ability to register tutors that have applied and also ban users for any misconduct. Parents can view their student's pending and completed assignments and grades from tutors. Tutors can view their active students, pending orders, provide assignments and grade them.

6.7.2 Overall Description

6.7.2.1 Product perspective



Figure 53: Product perspective diagram

6.7.2.2 Product features

Registration

Users are classified into four types: admin, tutors, parents and students. Parents will be able to log in using the student's credentials. Admin will be able to log in using admin credentials. Tutors will be able to log in to the tutor panel after becoming a tutor.

Notifications

All users will receive notifications from the system when certain event is triggered.

Become tutors

Users can become a tutor after filling out a form. An admin will approve the user and they will be registered as a tutor.

Booking feature

Users will book a tutor that are listed in the home page or using maps. A request will be made to the tutor and the tutor can adjust their price. If both parties come to an agreement the booking is complete. During this, tutor will be able to view the student's location.

Review and rating

After tutoring lessons are over, the student can leave a review and rating for the tutor. An AI assessment will summarize the overall reviews and categorize them as positive, negative and neutral.

Payment

Payment will be done within three days of booking using Khalti as the payment method.

Profile

Users will be able to view and edit their profile.

Assignments feature

Tutors will be able to provide assignments to students and grade once completed. Parents will be able to view pending or completed assignments and also the grades.

6.7.2.3 User class and characteristics

1) Student

- Can view tutors either directly in the maps with the tutor's location or in a listed card view.
- Can edit their profile.
- Can receive notifications.
- Can leave review and rating for tutors.
- Can make payments through Khalti.
- Can submit assignments.

2) Tutor

- Can give out assignments with difficulty levels.
- Can view and grade assignments.
- Can view active and pending students.
- Can edit their profile.
- Can receive notifications.
- Can set initial price and adjust price after request.

3) Parent

- Can view the student's pending and completed assignments along with grades.
- Can leave feedback under assignments.
- Can receive notifications.

4) Admin

- Can register applying tutors to the system.
- Can ban users for any misconduct.
- Can view financial records for users.

6.7.2.4 Operating environment

- i. **Client and server system:** The system needs a client and a server to function as an application. Users of the application will run the client side and request resources from the server which will be running on my computer.
- ii. **Web browser:** Since this is a web application, a web browser is required to use the system. The application is supported by Google chrome, Mozilla Firefox, Microsoft Edge, and so on.
- iii. **Database:** The system needs to store different information about different entities in order to have dynamic content. For this purpose, MongoDB will be used to store data.
- iv. **Operating system:** As of now, the web application will have to be able to run on MacOS, Windows and Linux.
- v. **Platform:** Client side logic will be handled using ReactJs and server side logic will be handled using NodeJS and Express.

6.7.2.5 Assumption and Dependencies

a) Assumptions

- i. There will be a single admin whose credentials will be pre made.
- ii. Payment can be done in person.
- iii. When creating account, the role and permissions are tailored for student by default but when the user becomes a tutor, the same account will be used to access tutor panel which is protected from other roles.
- iv. Maps will be integrated using google maps but is prone to change if a better alternative is found.
- v. Tutors will have to come to the student's house always.

b) Dependencies

- i. The system relies on external SDK or API will be used for map integration.
- ii. Khalti will be integrated through the use of SDK or API.

6.7.3 Functional requirements

6.7.3.1 Register User

Requirement No.	Requirement Description	
FR.1	Registration for parent and student and tutor.	
	System Requirement. ID	System Requirement
	SR.1	User fills out the form.
	SR.2	System asks user in the form checkbox if they want a parent panel.
	SR.3	If selected, parent panel and regular user panel will be accessible using the same credentials.
	SR.4	Form validations will take place and user will be created.
SR.5		Tutors will have to make a regular account and become a tutor after logging in.

Table 16: Functional requirement for register user

6.7.3.2 Login user

Requirement No.	Requirement Description	
FR.2	System Requirement. ID	System Requirement
	SR.6	User fills out form
	SR.7	System asks user in a checkbox if they are a parent.
	SR.8	If selected, user is sent to parent panel and if not, they are sent to the home page. If it is a tutor, they are sent to tutor panel.
	SR.9	Forgot password to reset user's password and add a new password.

Table 17: Functional requirement for login user

6.7.3.3 Book tutors

Requirement No.	Requirement Description	
FR.3	Allow users to book a tutor.	
	System Requirement. ID	System Requirement
	SR.10	User will filter out tutors.
	SR.11	User will book preferred tutor and send a request.
	SR.12	Tutor will receive the request, adjust price and send this response back.
	SR.13	User will accept new price and booking will be completed.

Table 18: Functional requirement for book tutors

6.7.3.4 Payment

Requirement No.	Requirement Description	
FR.4	Allow users to make payments.	
	System Requirement. ID	System Requirement
	SR.14	After hiring a tutor, users will have three days to pay the tutor.
	SR.15	When user selects the pay option, the system will take them to the payment gateway.
	SR.16	Users will pay using Khalti.

Table 19: Functional requirement for payment

6.7.3.5 Profile

Requirement No.	Requirement Description	
FR.5	Allow tutors and students to have a profile which can be updated.	
	System Requirement. ID	System Requirement
	SR.17	During sign up, all information taken in the form will be used to populate the user's profile.
	SR.18	User can go into their profile and click on edit profile and provide new details.
	SR.19	The information provided in the profile will be used to make dynamic personalized content in the application.

Table 20: Functional requirement for payment

6.7.3.6 Notifications

Requirement No.	Requirement Description	
FR.6	The application should show notifications to all users.	
	System Requirement. ID	System Requirement
	SR.20	Users get notified when all tutor sessions are over, on getting assignment, on having request accepted and when assignments are graded.
	SR.21	Admins get notified when tutor applies.
	SR.22	Parent gets notification when a submitted assignment gets graded and when an assignment is posted.
	SR.23	Tutors get notification when student sends a request and when parent posts a feedback.

Table 21: Functional requirement for notifications

6.7.3.7 Assignment

Requirement No.	Requirement Description	
FR.7	System Requirement. ID	System Requirement
	SR.23	Tutor provides assignment with due date and difficulty level.
	SR.24	User submits the assignment in image format and the tutor grades them.
	SR.25	Parents can view all submitted and pending assignments and the grades provided on them.

Table 22: Functional requirement for assignment

6.7.3.8 Rating and review

Requirement No.	Requirement Description	
FR.8	Users must be able to provide ratings and reviews on tutors. AI review will be based on the reviews.	
	System Requirement. ID	System Requirement
	SR.26	User leaves a review and rating of the tutor after the tutoring session is completed.
	SR.27	An overall rating of the tutor can be seen on their profile.
	SR.29	AI will be used to summarize all the reviews on the tutor and give a singular review which will be categorized as positive negative or neutral.

Table 23: Functional requirement for rating and review

6.7.4 External User Interface Requirements

6.7.4.1 User interfaces

- The user interface should be made simple without users having to face unnecessary complexities.
- Appearance should be kept minimalistic and appear appealing without overwhelming visual effects.
- User should achieve their goals with straightforward interactions.

6.7.4.2 Hardware interfaces

- The web application must run as long as there is an internet connection established.
- The web application should not cause any freezes due to the application using heavy graphical animations.

6.7.4.3 Software interfaces

- Application should be compatible with modern web browsers like Microsoft Edge, Mozilla Firefox, Safari, Google Chrome and so on.
- The application should run on modern operating systems like Windows, Linux, Mac and so forth.

6.7.4.4 Other non-functional requirements

- **Security requirements** – The application will take necessary measures to ensure security of user's sensitive information. Data will be encrypted using necessary measures. A hashing algorithm will be used to keep the user's password safe in the database so no attackers can breach the password. A trial lesson will take place for a three-day period to avoid any fraud cases. Admins will register the users to the system after verifying that the tutor is legit.
- **Performance requirements** – The application will ensure that waiting time is minimized. The response, speed, availability and recovery time will be kept in mind when developing the application. Testing will take place after development to ensure no app breaking bugs will be encountered after deployment.
- **Responsiveness requirement** – It is important that the application should be responsive and should support all screen sizes. The application will be developed keeping this in mind.

6.8 Code

6.8.1 Login client side

```
29  const {
30    register,
31    handleSubmit,
32    formState: { errors },
33  } = useForm({
34    resolver: yupResolver(schema),
35  });
36
37  const onSubmit = async (data) => {
38    try {
39      const response = await axios.post("http://localhost:4000/api/user/login", data);
40      const { success, message } = response.data;
41      if (success) {
42        toast.success(message);
43        localStorage.setItem("token", response.data.token);
44        navigate("/");
45      } else {
46        toast.error(message);
47      }
48    } catch (err) {
49      console.log(err);
50      toast.error("Something went wrong");
51    }
52  };
53
```

Figure 54: Login client side code

6.8.2 Login server side

```
25 module.exports.login_post = async(req,res) => {
26   const maxAge = 3*24*60*60;
27   try {
28     let { email, password } = req.body;
29     const user = await User.findOne({ email });
30     if (!user) {
31       return res.status(200).send({ message: 'Email does not exist', success: false });
32     }
33     const isMatch = await bcrypt.compare(password,user.password);
34     if (isMatch) {
35       const token = jwt.sign({id: user._id}, process.env.JWT_SECRET, {
36         expiresIn: maxAge
37       })
38       res.status(200).send({ message: "Login successful", success: true, token})
39     } else {
40       return res.status(200).send({ message: "Password is incorrect", success: false})
41     }
42   }
43   catch (err) {
44     console.log(err);
45     res.status(500).send({ message: "Error logging in", success: false, err})
46   }
47 }
```

Figure 55: Login server side code

6.8.3 Register client side

```
31  const {
32    register,
33    handleSubmit,
34    formState: { errors },
35  } = useForm({ resolver: yupResolver(schema) });
36
37  const onSubmit = async (data) => {
38    try {
39      const response = await axios.post(
40        "http://localhost:4000/api/user/signup",
41        data
42      );
43      const { success, message } = response.data;
44      if (success) {
45        toast.success(message);
46        navigate("/login");
47      } else {
48        toast.error(message);
49      }
50    } catch (error) {
51      toast.error("Something went wrong");
52    }
53  };
54
```

Figure 56: Register client side code

6.8.4 Register server side

```
5  module.exports.signup_post = async(req,res) => {
6    try {
7      let { email, password } = req.body;
8      const user = await User.findOne({ email });
9      if(user) {
10        return res.status(200).send({ message: 'User already exists', success: false });
11      }
12      const salt = await bcrypt.genSalt();
13      const hashedPassword = await bcrypt.hash(password, salt);
14      req.body.password = hashedPassword;
15      const newUser = new User(req.body);
16      await newUser.save() //save to mongodb
17      res.status(200).send({ message: "User created successfully", success: true });
18    }
19    catch(err) {
20      console.log(err);
21      res.status(500).send({message: "Error creating user", success: false, err });
22    }
23  }
```

Figure 57: Register server side

7 Bibliography

- Dyvik, E. H. (2023). Education worldwide - statistics & facts. *Statista*.
- Egeonu, E. (2022, 6 7). *The Advantages and Disadvantages of the RAD Model / DistantJob - Remote Recruitment Agency*. Retrieved from DistantJob: <https://distantjob.com/blog/rad-model-advantages-and-disadvantages/>
- Lal, R. (2022, 2 23). *What Is Software Prototyping And Why Do You Need It?* Retrieved from Stackify: <https://stackify.com/what-is-software-prototyping-and-why-do-you-need-it/>
- List of Scrum Advantages and Disadvantages / Indeed.com*. (2023, 8 24). Retrieved from Indeed: <https://www.indeed.com/career-advice/career-development/disadvantages-of-scrum>
- Nepal Education Fact SheetsI 2022 - UNICEF Data*. (n.d.). Retrieved from UNICEF: https://data.unicef.org/wp-content/uploads/2023/06/Nepal_factsheet_Mar_2023.pdf
- Rana, K. (2021, April 28). *Spiral Model - Definition, Phases, Advantages and Disadvantages*. Retrieved from ArtOfTesting: <https://artoftesting.com/spiral-model>
- Saud, M. S. (2023). Stagnation in Nepal's education system is hindering progress and development. *onlinekhabar*.
- SDLC - RAD Model*. (n.d.). Retrieved from tutorialspoint: https://www.tutorialspoint.com/sdlc/sdlc_rad_model.htm
- SDLC - Waterfall Model*. (n.d.). Retrieved from TuorialsPoint: https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm
- Sujan. (2021, 7 16). *What is Education? Meaning, Types, Functions, and Approaches*. Retrieved from tyonote: https://tyonote.com/what_is_education/
- What is Figma?* (n.d.). Retrieved from noble desktop: <https://www.nobledesktop.com/learn/figma/what-is-figma>
- Whose failure?* (2023, 8 16). Retrieved from The Kathmandu Post: <https://kathmandupost.com/editorial/2023/08/16/whose-failure#:~:text=The%20Grade%2012%20National%20Education,upgrading%20to%20the%20University%20level.>