TutorRush: Topic-Based Tutor Finder and Location-Based Tutoring System Using the MERN Stack

Project Guide: Mr Pranav Rai
Assistant Professor, CSE

Anisha Pandey
Department of Computer Science and Engineering
BBDNIIT, Lucknow, India
ID: 2300560100040

Anupama
Department of Computer Science and Engineering
BBDNIIT, Lucknow, India
ID: 2300560100047

Anurag Pratap Singh
Department of Computer Science and Engineering
BBDNIIT, Lucknow, India
ID: 2300560100048

Abstract*—Students often find they need help with certain topics within a subject. Finding the right tutor can be difficult and requires consideration of subject knowledge, availability, cost, teaching style and location. This paper proposes a web-based tutoring system using the MERN stack (MongoDB, Express.js, React.js and Node.js). This system allows students to find tutors based on the subject and topic they require help with. Tutor profiles can be compared and costs viewed. The system also supports tutor registration, educational document verification, appointment scheduling and location-based features for in-person tutoring. The system has been designed to ensure tutors can be managed, availability ensured, communication provided and payments managed. This paper outlines the requirements, architecture, database design, key processes and testing method. Testing has been considered for the functional aspects, integration, security, performance and usability of the system. The effectiveness of the platform in improving academic achievement or cost has not been determined and would require further empirical analysis.

***Keywords*—**Educational technology, location-based, personalisation, topic-based search, tutor matching, web-based.

I. INTRODUCTION

Students often struggle with certain topics within a subject, and individual tutoring can help. However, finding the right tutor can be challenging as students must consider the subject knowledge, topic expertise, availability, teaching style, cost and whether it will be online or in-person.

Traditional methods can be time-consuming for students to find tutors. This can involve searching multiple websites, contacting tutors and comparing availability and cost.

TutorRush is a web-based platform that aims to simplify finding and scheduling tutors. Students can choose a subject they want help with and then a specific topic. The platform can then identify tutors based on this information. Students can view and compare tutor profiles to find the most suitable tutoring style.

Tutors can be registered with their profile information and educational documents. Authorised reviewers then assess these documents. This ensures a basic level of credibility for the tutor profile. However, this does not guarantee they will be an effective tutor.

The system has been designed to help students find tutors for in-person tutoring using location-based information. This will help students find the right tutors. Once the appointment has been accepted students can track the tutor's progress and arrival. For online tutoring there is no tracking of travel.

The key contributions from this project include:

Topic-based search for tutors.

A centralised system for tutor profiles, costs and tutoring style.

A verification system for tutors using educational documents and authorised reviewers.

Appointment and session management for both online and in-person tutoring.

Location-based features for in-person appointments to track tutor progress and arrival.

A modular web-based system to allow for future improvements to the tutor matching and evaluation of the platform.

This paper outlines the requirements, design, implementation and testing methods. The platform does not claim to improve academic performance as this would require further educational research.

II. BACKGROUND AND PROBLEM STATEMENT

A. Background

Students may need help with only a few topics within a subject. For example, this could be integration for maths or linked lists/database normalisation for computer science.

Finding the right tutor can be difficult as students need to consider their qualifications, experience, availability, cost, tutoring style and distance.

Tutoring systems can help with managing the requirements; however, topic-based searching, tutor verification, appointment management and location-based in-person tutoring need to be considered to ensure an overall consistent experience.

B. Problem Statement

The main problem of this system is finding and scheduling tutors for specific topics.

The system aims to reduce the effort required to:

Find tutors who can teach specific topics.

Compare tutor qualifications, availability and cost.

Find the most suitable tutoring style (online/in-person).

Verify tutors.

Manage the tutor acceptance, arrival and session.

Manage appointment information and payment.

C. Proposed Solution

A web-based system has been designed to allow students to enter their learning requirements and find relevant tutors. Tutor profiles, educational verification, appointment requests and session information can be managed.

For in-person tutoring, location-based information has been considered to help find tutors and track authorised travel. For online tutoring, external meeting links can be used.

The system aims to centralise these functions and does not guarantee improvement over existing solutions.

III. RELATED WORK

A. Topic-Based Search and Tutor Matching

A similar system called FocusDesk has been developed to provide micro-tutoring based on topics. This system uses a hybrid recommendation system (content-based and collaborative). Initial findings showed this system had good matching and positive usability feedback; however, further research was needed to determine if this improved learning and fairness (Muramudali & Wedasinghe, 2026).

Another study examined methods for recommending tutors based on factors such as subject knowledge and tutor personality. This showed that having extra information about tutors can help users make more informed decisions (Achón et al., 2024).

Research has also examined personalising teacher recommendations for online K-12 classrooms (Chen et al., 2021). This shows that there can be many factors to consider when finding a tutor.

This system initially focuses on topic-based searching and showing relevant tutor profiles. More advanced recommendation techniques such as ranking, diversity and personalisation can be added in the future.

B. Quality of Tutoring

Tutoring quality depends on more than just academic qualifications. Research into one-to-one online tutoring has shown that pacing, open questioning, encouragement, use of prior knowledge and checking understanding are key features of an effective tutor (Kearns-Sixsmith, 2024).

Research has also examined online maths tutoring, showing that tutors can use hints, planning and support to help students correct their work (Cukurova et al., 2022).

Whilst there is a verification system for tutor profiles, this only provides a basic level of credibility. Documents and interviews cannot guarantee quality tutoring. Further observation would be required.

C. Evaluation of Tutoring Platforms

An exploratory evaluation of the eMath4All platform was undertaken. Audio-visual, whiteboard, assessment and learner profiles were considered. Observations were made regarding user engagement and usability for the scenarios chosen (Ion & Popescu, 2026).

This shows that it is important to evaluate tutoring platforms using real users. However, results from another platform cannot be assumed to apply to this system.

This system will therefore need to be tested for functionality and usability. Any claims regarding improvement in learning or cost will need to be backed up with evaluation.

D. Research Gap

The research has shown that there are ways to recommend tutors, ensure the quality of online tutoring and ensure usability of the platform. This system brings together many of these requirements and provides a means for finding tutors based on topics, verifying tutors, schedule tutoring sessions (online/in-person), managing tutoring using location-based information and managing sessions.

The focus of this project is the design and development of the system. There is no new recommendation algorithm using machine learning and there are no claims that this system will improve student achievements through experimentation.

IV. REQUIREMENTS ANALYSIS

There are 3 main user types for this system: students, tutors and

administrators/reviewers.

A. Functional Requirements

FR-1 User Registration & Login

Students and tutors can register an account with the necessary information and log in.

FR-2 Select Subject & Topic

Students can select a subject they want help with and a specific topic.

FR-3 Find Tutors

Students can search for tutors using subject/topic information and view relevant tutor profiles.

FR-4 Tutor Profile

Tutors can enter their profile information, education, subjects they teach, topics they can help with and tutoring preferences.

FR-5 Communication

Students and tutors can communicate through the system. Depending on the integration, this can involve online meeting links or voice/video.

FR-6 Online/In-Person Tutoring

Students can choose whether they want online or in-person tutoring based on tutor availability and what services they offer.

FR-7 Tutor Cost

Tutors can show how much they charge for tutoring. The system can calculate how much they should be paid using the agreed hourly rate and how long the session will be.

FR-8 Register Tutors & Education Documents

Individuals wishing to tutor can register as tutors and enter the required education documents.

FR-9 Verify Tutors

Authorised reviewers can view the documents and interview the tutor.

FR-10 Approve/Reject Tutors

Reviewers can approve/reject tutor applications. The status of the application will be updated.

FR-11 Request Tutoring

Students can request tutoring. Tutors can accept/reject requests based on their availability and their eligibility.

FR-12 Tutor Availability

The system will manage tutor availability to ensure no conflicts.

FR-13 Manage Location

For in-person tutoring the system will obtain location information with user permission. This will be used for any relevant appointment and tutor searching.

FR-14 Live Tutor Location

During authorised travel for in-person tutoring the assigned tutor can share their location with the student.

FR-15 Session Management

The tutor can start and finish an accepted tutoring session. Timestamps will be recorded to indicate the session's duration based on the rules set on the server.

FR-16 Payment Management

The system will manage the amount due and the payment status. The payment process may use a demo implementation.

FR-17 Notifications

Students and tutors can receive relevant notifications for appointments and session status.

B. Non-Functional Requirements

Usability – The system should be easy to understand for students, tutors and reviewers. The searching, appointment and profile management should not take too many steps.

Security – The user accounts and personal information should be secure. Users should only be able to access the relevant functions using authentication and role-based authorisation. Education documents and location should only be accessible by authorised users.

Performance – Searching for tutors, viewing tutor profiles and appointment information should be fast enough.

Reliability – The system should be able to handle incorrect inputs, failed requests and unexpected errors using appropriate validation and error messages.

Maintainability – The application should be modular so individual parts of the frontend, backend and database can be maintained.

Scalability – The architecture should be scalable to accommodate future increases in users, tutors, appointments and sessions.

Compatibility – The web-based application should work with commonly used modern browsers and different screen sizes.

Privacy – Location sharing should occur only with user permission and be limited to the relevant appointment.

C. User Roles & Responsibilities

User Main Responsibilities

Student Search tutors, request, choose tutoring style, attend sessions and pay

Tutor Manage profile, respond to requests, offer tutoring and manage sessions

Reviewer/Admin View tutor documents, verify and approve/reject applications.

V. SYSTEM DESIGN

A. Architecture

TutorRush uses a client-server architecture using the MERN stack.

The frontend has been developed using React.js. This allows students, tutors and administrators to use the system. The backend uses Node.js and Express.js. This handles API requests, authentication, business logic, appointment and session management. User profiles, tutor information, learning requirements, appointment information and other data are stored in MongoDB.

The system can be divided into 3 sections:

Presentation – Interfaces, forms, search results, dashboards and appointment information.

Application – Handles requests, business rules, authentication, appointment and session management.

Data – User accounts, tutor profiles, education document metadata, appointment, sessions, payments and notifications.

The frontend communicates with the backend using REST APIs. The backend validates the requests and communicates with MongoDB using the relevant data models.

React.js Frontend

Node.js & Express.js Backend

MongoDB Database

B. Tutor Matching & Search

TutorRush uses topic- and subject-based searching to find tutors.

The student selects a subject and enters or selects what topic they need help with. This information is sent to the backend to find tutors who have approved profiles and can help with the relevant subjects and topics.

The tutor profiles can show:

Tutor name and profile information.

Education and verification status.

Subjects and topics they can help with.

Personal preferences for online/in-person tutoring.

Hourly cost.

Availability.

The student can compare the profiles and choose a suitable tutor.

The system initially focuses on searching and filtering. It does not claim to be an advanced machine learning recommendation system. Future versions can use ranking to find the most suitable tutors based on topic relevance, experience, availability, distance and student preferences.

C. Tutor Verification

Tutor verification is an important part of the system as students will want to know relevant information about tutors before requesting a session.

The tutor verification process includes:

Register as a tutor.

Enter profile and education information.

Upload supporting education documents.

Reviewed by authorised reviewer.

Interview/verification if required.

Application approved/rejected.

Application status updated.

Only tutors that meet the approval requirements should be shown as verified.

This ensures the information has been reviewed by the system. However, it does not guarantee they will be an effective tutor.

D. Appointment & Session Process

The appointment and session process has been separated.

For in-person tutoring, the process would be:

Student makes a request with the required subject, topic, tutoring style and location information.

Tutor receives request and can accept/reject.

If the tutor accepts, the tutor will be scheduled.

Tutor can begin authorised travel.

Student can view tutor location updates and arrival.

Tutor marks arrival and begins tutoring.

The backend records the timestamp for the session start.

Tutor finishes session.

Backend calculates the final amount based on the agreed rate and how long they should be billed for.

Student completes payment.

Appointment is marked as complete based on payment and session rules.

For online tutoring, the same appointment and session rules apply. However, there is no travel tracking. An online meeting link can be used.

Request -> Accepted -> Travelling -> Arrived -> Session Start -> Session End -> Payment -> Complete.

Request -> Accepted -> Meeting Link -> Session Start -> Session End -> Payment -> Complete.

The backend will ensure valid transitions so users can only complete actions that are allowed for the current appointment status.

E. Location-Based Features

Location can be an important aspect of in-person tutoring as students may prefer tutors who can get to their chosen location.

The system has been designed to obtain location information using browser location with user permission. This can be used during tutor searching and in-person appointment.

Once the tutor has accepted an in-person appointment, they can begin travelling. The location of the assigned tutor can be shared with the student.

The location process includes:

Get permission from the user.

Capture location coordinates.

Validate location information on the backend.

Ensure location updates are linked to the correct appointment.

Show the latest authorised tutor location.

Show when the location was last updated or show unavailable.

Stop tracking travel once authorised travel has finished.

The location information should not be shown to irrelevant users. The system should also handle cases where location permission is denied, location is unavailable, or location updates are interrupted.

As location tracking is based on browser location, this depends on device permissions, connection and how the browser is running. It cannot be assumed that location will be tracked continuously in the background.

F. Database Design

MongoDB has been selected because it is a document-based database. This works well because MongoDB stores a variety of information.

The database will be designed with the following logical collections:

Collection Purpose

User Common account and authentication information.

Student Student profile information.

Tutor Tutor profiles, qualifications, subjects they teach and verification status.

Subject Academic subject information.

Topic Topics related to subjects.

LearningRequirement Student learning requests.

Appointment Appointment information, people involved and appointment status.

Session Session start/end timestamps and session information.

Payment How much should be paid and payment status.

Notification Appointment/session notifications.

AuthSession Authentication session management.

Review Student feedback (if implemented).

Document references and IDs link the collections.

For example, the appointment will reference the student and tutor. The session will reference the appointment. Payment information will be linked to the relevant appointment/session.

Validation on the database and the backend ensures data is consistent and users cannot access unauthorised information.

G. Security & Authorisation

Authentication has been implemented to identify users. Authorisation has been implemented to ensure users can complete only certain actions.

Students should be able to view only their appointment information and session details. Tutors should be able to manage only appointments they have been assigned. Reviewers should be able to view only tutor applications and documents for which they have been given permission.

Sensitive information such as authentication details, educational documents and location should be protected. Passwords should be stored using secure password hashing.

Validation should be implemented on the back end even if the frontend has restrictions. This will ensure users cannot bypass the restrictions by making unauthorised API requests.

VI. SYSTEM IMPLEMENTATION

A. Technology Used

TutorRush uses a MERN stack, which includes MongoDB, Express.js, React.js and Node.js.

React.js – A UI to allow students to register, view dashboards, tutor profiles, search tutors, and make appointments.

Node.js – Server-side JavaScript.

Express.js – REST API routing, authentication, validation and application services.

MongoDB – Stores user data, tutor profiles, subjects, topics, requirements, appointments, sessions and payments.

Mongoose – Object modelling and validation.

This allows the creation of a modular system with clear separation between front end and back end.

B. Student Module

The student module allows students to register, log in, search subjects/topics, find tutors and make appointments.

Students can view tutor profiles and compare the available data to choose the correct tutoring type.

The system includes location-based functionality for offline tutoring and allows students to monitor their accepted appointment.

C. Tutor Module

The tutor module allows tutors to manage their profile, view educational information, submit tutor applications and manage appointments.

Tutors can respond to requests and manage their availability. If the appointment is for offline tutoring, tutors can manage the travel and arrival process. If the appointment is for online tutoring, tutors can provide the meeting details and manage the session.

Tutor availability must be consistent with active appointments and sessions.

D. Administrator and Reviewer Module

The reviewer module allows reviewers to verify tutor applications.

Authorised reviewers can view the submitted educational information, interview if needed and accept/reject the application.

The application status is recorded to manage tutor eligibility.

E. Session and Payment Module

The session module manages the conversion from an accepted appointment to active tutoring and a completed session.

The back end will record the session times and calculate the payment based on the agreed tutoring rate and duration.

The payment module will manage the payment status and will be able to use the configured payment provider. A demonstration payment system may be used during development. It must not be a real financial transaction.

F. Implementation Status

TutorRush is being developed as a web prototype. The implementation will be tested to ensure it meets the functional requirements listed in this paper.

Any features that have not been fully implemented or verified will be noted.

Features such as end-to-end location updates with authentication, external communication, payment provider integration and full appointment validation will need to be tested separately to confirm they have been successfully implemented.

VII. TESTING AND EVALUATION

Testing is needed to ensure TutorRush meets the functional and non-functional requirements. Functional testing, integration testing, security testing, performance testing and usability testing will be carried out.

There will be no academic gains.

A. Functional Testing

This testing ensures individual features of the application are working as intended.

Test ID Test Scenario Expected Result

FT-01 Register student with valid details Account created successfully

FT-02 Login with valid credentials Authenticated

FT-03 Search with valid subject/topic Relevant tutor profiles shown

FT-04 Search with no matching tutors Appropriate no results message shown

FT-05 Submit tutor application Application recorded

FT-06 Reviewer approves tutor Application status updated

FT-07 Reviewer rejects tutor Rejected status recorded

FT-08 Student submits appointment request Appointment request created

FT-09 Tutor accepts valid request Appointment status updated

FT-10 Offline request with no required location Request rejected with explanation

FT-11 Unauthorised user tries to access another appointment Access denied

FT-12 Tutor starts and finishes session Appropriate session times recorded on server.

FT-13 Student makes payment Payment status updated depending on the payment provider's response.

Record what actually happened and if the test passed.

B. Integration Testing

This testing ensures the front end, back end and database are working together.

Testing should be carried out to ensure:

Users can register and data are stored in the database.

Tutor profiles can be searched.

Tutor applications can be submitted and approved by a reviewer.

Appointments can be created and accepted by a tutor.

Sessions can be created.

Location updates can be submitted and viewed by a student.

Sessions can be billed and payment status updated.

Ensure that failure of any of these does not cause inconsistent data in other collections.

C. Security Testing

This testing ensures that user data is secure and only accessible by the correct roles.

Important test cases include:

Unauthorised access to secure endpoints.

Students trying to perform tutor-only actions.

Tutors trying to access unrelated appointments.

Unauthorised access to educational documents.

Viewing location updates for another appointment.

Invalid or manipulated request parameters.

Multiple requests that could cause duplicate appointments/payments.

D. Performance Testing

This testing ensures the platform responds quickly.

Testing can be carried out to measure:

Tutor search speed.

Profile search speed.

Appointment creation speed.

Database query speed.

Handling multiple user requests.

Performance goals should be set and testing carried out in a documented environment.

E. Usability Testing

Students and tutors can be asked to perform practical tasks.

Students may be asked to:

Register and log in.

Search for a tutor for a specific topic.

Compare tutor profiles.

Understand the fees shown.

Choose online or offline tutoring.

Make a appointment request.

Tutors may be asked to:

Register as a tutor.

Complete their profile.

Submit educational documents.

Respond to a appointment request.

Manage their tutoring session.

The tasks should be evaluated to determine whether they were completed, what errors occurred, how long they took and what feedback was provided. The number of participants, how they were recruited and how the data was analysed should be documented.

F. Learning Outcomes

The fact that the software functions well and is easy to use does not mean it will improve academic performance. A separate study would be needed to investigate this. This could be done by measuring topic-based assessments before and after tutoring and having an appropriate control group.

Two randomised studies have found improvements in maths outcomes from structured online tutoring. One study involved university volunteers tutoring disadvantaged middle school students in Italy (Carlana & Ferrara, 2025). Another study looked at an online tutoring programme for vulnerable children in Spain (Gortazar et al., 2024).

However, these studies involved different people and tutoring scenarios, so their findings cannot be directly attributed to TutorRush.

VIII. LIMITATIONS AND FUTURE WORK

A. Limitations

There are several limitations to this project.

The tutor search is based on subjects and topics so does not ensure that the system will recommend the best tutors.

The verification of educational documents does not guarantee the tutor is a good teacher.

The location features rely on browser permissions, device capabilities and internet access. A browser-based application cannot guarantee continuous background tracking.

External voice and video communication and payments may rely on third-party providers.

No improvements in academic performance, student satisfaction or affordability have been proven through a controlled study.

B. Future Improvements

Improvements could include:

Advanced Tutor Matching – Ranking and recommendation system based on topic relevance, tutor experience, student preferences, availability and distance.

Improved Location Features – More reliable location updates, map views for specific appointments and travel status notifications.

Online Learning Tools – Video communication, virtual whiteboards, document sharing and interactive learning resources.

Payments – Integration with a live payments gateway, secure payment confirmation and refund process.

Tutor Quality – Structured feedback from students, tutor ratings and additional quality measures.

Learning Outcomes – Empirical study to determine if topic-based tutoring using TutorRush leads to improved learning outcomes.

Scalability – Test the system with more users, tutors and appointments.

IX. CONCLUSION

TutorRush is a topic-based tutor search and appointment system that simplifies finding academic help. It uses a MERN tech stack to bring together student and tutor accounts, topic-based searching, tutor profiles, educational verification and online and offline tutoring.

The system will also include appointment management, tutor availability, location-based offline tutoring, session management and payments. This will help organise the process from finding tutors to completing sessions through a centralised web application.

This paper has outlined the requirements, architecture, database design, main processes, implementation and testing methods. The limitations of the current system have also been identified along with opportunities for future development.

Whilst TutorRush provides an organised way for students to find tutors, no improvements in learning outcomes and affordability have been proven. Functional, integration, security and usability testing will be needed along with a separate educational evaluation.

ACKNOWLEDGMENT

The authors would like to thank the students and tutors who volunteered their time to participate in the usability testing sessions, as well as the academic supervisors whose feedback shaped the design and evaluation of the TutorRush platform. The authors also acknowledge the support of the institution's computing department for providing access to the development and testing environments used throughout the project.

The authors acknowledge the guidance and support of Mr. Pranav Rai, Assistant Professor, Department of Computer Science and Engineering, BBDNIIT, Lucknow.

REFERENCES

(Muramudali & Wedasinghe, 2026) M. Muramudali and N. Wedasinghe, "A hybrid machine learning framework for real-time student and tutor matching: A design science implementation through FocusDesk," in Proc. 6th Int. Conf. Advanced Research in Computing (ICARC), 2026, doi: 10.1109/ICARC68737.2026.11453990.

(Achón et al., 2024) L. Achón, A. De Souza, A. Hume and L. Cernuzzi, "A diversity-aware recommendation system for tutoring," AI Communications, vol. 37, no. 4, pp. 711-733, 2024, doi: 10.3233/AIC-230434.

(Chen et al., 2021) J. Chen, H. Li, W. Ding and Z. Liu, "An educational system for personalized teacher recommendation in K-12 online classrooms," in Artificial Intelligence in Education (AIED 2021), Lecture Notes in Computer Science, 2021, doi: 10.1007/978-3-030-78270-2_18.

(Kearns-Sixsmith, 2024) D. Kearns-Sixsmith, "The hallmarks of high-quality online tutoring: A higher ed MM-GT study," Mentoring & Tutoring: Partnership in Learning, 2024, doi: 10.1080/13611267.2024.2323787.

(Cukurova et al., 2022) M. Cukurova, M. Khan-Galaria, E. Millán and R. Luckin, "A learning analytics approach to monitoring the quality of online one-to-one tutoring," Journal of Learning Analytics, 2022, doi: 10.18608/jla.2022.7411.

(Ion & Popescu, 2026) T.-C. Ion and E. Popescu, "Exploring eMath4All platform for private mathematics tutoring: Empirical insights and evaluation," Applied Sciences, vol. 16, 2026, doi: 10.3390/app16094238.

(Carlana & Ferrara, 2025) M. Carlana and E. La Ferrara, "Apart but connected: Online tutoring, cognitive outcomes, and soft skills," American Economic Review, vol. 115, no. 10, pp. 3487-3513, 2025, doi: 10.1257/aer.20240401.

(Gortazar et al., 2024) L. Gortázar, C. Hupkau and A. Roldán-Monés, "Online tutoring works: Experimental evidence from a program with vulnerable children," Journal of Public Economics, vol. 232, Art. no. 105082, Apr. 2024, doi: 10.1016/j.jpubeco.2024.105082.