

Tutor Connect A Real Time Web Tutoring App

Dr.N. Baggyalakshmi^{1*}, J. Kavipriya², Dr.R. Revathi³

^{1*} Assistant Professor, Department of Computer Science, PSGR Krishnammal College for Women, Coimbatore.

E-mail: baggyanethra@gmail.com

² Student, B.sc Computer Science, PSGR Krishnammal College for Women, Coimbatore. E-mail: kavipriya2006@gmail.com

³ Assistant Professor, Department of Computer Science, Karpagam Academy of Higher Education, Coimbatore.

E-mail: ravathilakshay@gmail.com

Abstract--- The main objective of this internship was website design and development. Several programming languages that are in use to develop a web-based application or software. Some of them are only used for the UI and the frontend of the application, some are used for the backend design of the software. For Example- Django, Html etc. are some programming language to develop frontend of an application. java etc. are used at the backend. Nowadays there are also some framework' s that use vastly. Frameworks are basically structured programming by using Model, View, and Controller. If we develop web-based application that is very useful for us because we can access it from anywhere of the world. It is very helpful for our daily life. That is why I choose subject of my report is "Website design and development". Working in EzerBase added huge experiences in my upcoming career. This report takes us through all the details of Website design and development knowledge and experience gathered during this internship period. Beyond technical skills, interns will engage in user research and usability testing, working closely with the design team to refine user interfaces and enhance the overall user experience. They will learn to balance functionality, performance, and aesthetics to develop applications that align with modern design trends and best practices. In conclusion, this web application development internship offers a unique opportunity for individuals passionate about creating seamless user experiences through innovative web applications. By combining technical expertise with a user-centric approach, interns will gain practical skills, professional growth, and a chance to contribute to cuttingedge projects in the realm of web development.

Keywords--- Tutor, User Interface, Time Web.

I. INTRODUCTION

The field of education has undergone a significant transformation in recent years, with online learning becoming increasingly popular and accessible. As a result, the demand for efficient and user-friendly online tutoring platforms has soared [1]. This internship opportunity in web application development focuses on creating an innovative and interactive online tutor web application that brings together students and tutors in a virtual learning environment. The goal of this internship is to provide aspiring web developers with a hands-on experience in developing a web application specifically designed for online tutoring. Participants will have the chance to contribute to a project that addresses the growing need for remote educational solutions, offering students the opportunity to access quality tutoring services from anywhere, at any time.

Throughout the internship, participants will work closely with a team of experienced developers and designers to develop the web application from conception to deployment [2]. They will engage in various stages of the development lifecycle, including requirement gathering,

system design, database development, front-end and back-end implementation, testing, and integration. Interns will have the opportunity to explore and utilize cutting-edge web technologies and frameworks, such as DJANGO, HTML, CSS, JavaScript, and relevant libraries and frameworks like React or Vue.js. They will also gain exposure to server-side scripting languages like Python or Node.js, and database management systems to ensure efficient data storage and retrieval. In addition to technical aspects, interns will collaborate with the design team to create an intuitive and visually appealing user interface [3]. They will focus on developing features that facilitate seamless communication between tutors and students, such as live chat, virtual whiteboards, file sharing, and scheduling functionalities.

Throughout the internship, participants will be encouraged to propose innovative solutions and contribute their ideas to enhance the application's usability and performance. They will actively participate in team discussions, receive guidance and feedback from experienced mentors, and have the opportunity to refine their skills in a collaborative and supportive environment [4]. Upon completion of the internship, participants will have gained valuable experience in web application development,

specifically in the context of online tutoring platforms [5]. They will have a comprehensive understanding of the development lifecycle and the skills necessary to build interactive and responsive web applications.

The paper will provide interns with a strong foundation for a career in web development, with a focus on educational technology and remote learning solutions. In summary, this internship opportunity in online tutor web application development offers an exciting chance to contribute to the rapidly evolving field of online education. By developing a comprehensive and user-friendly web application, interns will play a pivotal role in bridging the gap between students and tutors in a virtual learning environment. Through hands-on experience and collaboration with industry professionals, participants will acquire valuable skills and knowledge that will pave the way for a successful career in web development and educational technology.

II. LITERATURE REVIEW

Alfaro et al [6] An overview of intelligent tutorial systems (ITS) is provided, with an emphasis on how well they work with contemporary learning paradigms. In order to gather information about ITS and e-Learning systems, a bibliographical survey of classic works in printed and digital literature was employed, along with database searches for these and other works from universities and digital repositories. The primary research shortcoming stems from the fact that the search was restricted to documents written in Portuguese, Spanish, and English.

Saiyad et al [7] established competency-based curricula and promoted e-learning as a vital resource for independent study. Adopting excellent online teaching strategies is necessary for online learning to be successful. These include synchronous and asynchronous teacher-student contact, online teaching and learning that is in line with curriculum and objectives, and supporting students' active learning, self-directed learning, and development of higher order thinking skills. Good online teaching strategies should also include a feedback mechanism, respect for different learning styles and abilities, efficient time management, and ongoing student observation and mentoring.

Pratama et al [8] Artificial intelligence (AI) has rapidly advanced, creating new opportunities across many industries, including education. AI technology is enabling individualized learning experiences that are gradually replacing traditional one-size-fits-all teaching approaches. This article examines how AI is transforming education by enhancing overall learning outcomes, raising engagement, and customizing the learning process to each student's needs. This study employed a qualitative descriptive research design, collecting data through the use of questionnaires, interviews, observations, and documentation.

Chung et al [9] In an effort to expand and improve education, colleges are using e-learning platforms more and more. There hasn't been much research done on how electronic peer tutoring affects students' learning, according

to the literature review. As a result, there aren't many technological affordances available to support remote appointments and online peer tutoring sessions. In order to enable in-person and remote peer tutoring, a new smartphone app called Mobile Education Networked Tutoring On Request (MENTOR) was developed. This paper details its development. By utilizing smartphone location data, the MENTOR app coordinates in-person tutoring sessions and allows for online remote tutoring sessions. It also uses tutor-tutee matching to predict students' tutoring needs.

Seo et al [10] Artificial intelligence (AI) technologies provide efficient assistance for remote learning and teaching. These systems can personalize the learning process for students, automate repetitive work for instructors, and facilitate adaptive evaluations. Even though artificial intelligence (AI) presents many benefits, it is yet unclear how AI systems will affect student-teacher interactions, norms, and culture. The happiness and learning results of students in online learning are significantly influenced by the learner-instructor relationship, which includes communication, presence, support, and interaction. To find any holes, difficulties, or obstacles stopping AI systems from reaching their full potential and jeopardizing the security of these interactions, it is crucial to find out how instructors and students view the influence of these systems on their interactions.

III. PROPOSED METHODOLOGY

3.1. Modules in the System

User Authentication and Registration: This module allows users to create accounts, login, and manage their profiles. It includes features such as email verification, password reset, and profile customization.

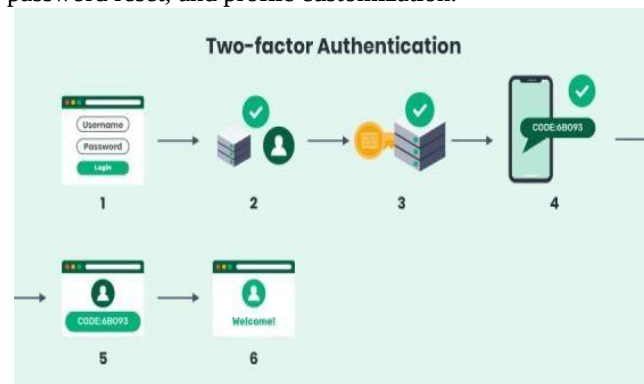


Figure 1 - Login Method

Tutor and Student Dashboard: The dashboard serves as the main interface for tutors and students. Tutors can manage their availability, create tutoring sessions, view student progress, and access resources. Students can search for tutors, schedule sessions, track their progress, and access learning materials.

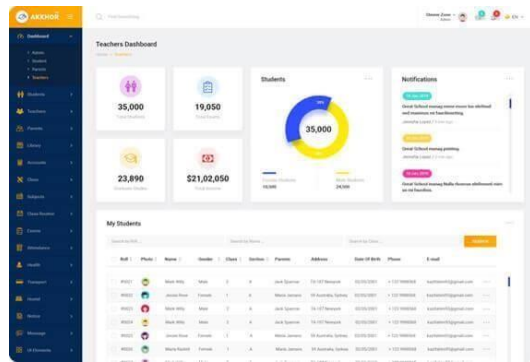


Figure 2 - Dashboard

User Management: This module handles user registration, login, and authentication. It also manages user profiles, including personal information, preferences, and settings.

Tutor Profile Management: This module allows tutors to create and manage their profiles. It may include features such as adding qualifications, expertise, availability, and setting tutoring rates.

Profile Management: Similar to the tutor profile management module, this module enables students to create and manage their profiles. It may include features like academic information, learning goals, and preferences.

Course Catalog: This module provides a comprehensive list of courses or subjects available for tutoring. It includes search and filtering capabilities, allowing students to find specific courses of interest.

Scheduling and Booking: This module facilitates the scheduling and booking of tutoring sessions. It allows students to view tutor availability, select suitable time slots, and book sessions accordingly. It may also handle reminders and notifications.

Virtual Classroom: This module offers a virtual learning environment where tutors and students can interact in real-time. It typically includes video conferencing, chat functionality, a whiteboard for collaborative note-taking, and file sharing capabilities.

Feedback and Rating: This module enables students to provide feedback and ratings for tutors and vice versa. It helps maintain quality standards and assists future students in selecting suitable tutors.

Content Management: This module allows administrators to manage the website's content, including adding or updating course materials, resources, and other educational content.



Figure 3 - Steps of Tutoring

These modules provide a foundation for developing an online tutor web application. Depending on your specific requirements, you can customize and extend these modules or add additional ones to suit your application's needs.

Progress Tracking: The system enables tutors and students to track progress overtime. It may include features like session summaries, progress reports, and milestone achievements to monitor the learning journey.

Communication and Notifications: The platform allows tutors and students to communicate via messaging or chat features. It also sends notifications for session reminders, updates, and announcements.

Content Sharing: Tutors can share educational resources, documents, and study materials with their students through the platform. This facilitates seamless access to supplementary materials.

3.2. Mechanics

User Registration and Authentication: Implement a user registration and authentication system to allow tutors and students to create accounts and log in securely. This ensures that only authorized individuals can access the platform and its features.

Tutor Profile Creation: Provide a mechanism for tutors to create their profiles, including information such as their qualifications, areas of expertise, teaching experience, and availability. This profile information helps students make informed decisions when selecting a tutor.

Student Search and Selection: Develop a search functionality that allows students to find tutors based on specific criteria such as subject, level, availability, and ratings. Additionally, provide a mechanism for students to select and book tutors based on their preferences.

Scheduling and Booking: Implement a calendar system that enables tutors to manage their availability and allows students to book tutoring sessions based on the available time slots. Consider integrating notifications or reminders to ensure both parties are aware of upcoming sessions.

Live Video Communication: Integrate a video conferencing tool that enables real-time communication between tutors and students. This can be accomplished using technologies such as WebRTC or third-party video conferencing APIs. Ensure that the video communication system is reliable and provides good audio and video quality.

Sharing and Collaboration: Enable tutors and students to share learning materials, assignments, and resources during the tutoring sessions. This can be achieved through features such as file uploads, document sharing, and collaborative whiteboards.

Payment Processing: Implement a secure payment gateway to facilitate financial transactions between students and tutors. This could involve integrating popular payment processors or building a custom payment system that handles transactions securely.

Ratings and Reviews: Include a rating and review system that allows students to provide feedback on the

tutoring sessions and the overall experience. This helps build trust and credibility within the platform and assists other students in selecting the most suitable tutors.

Administrative Dashboard: Develop an administrative dashboard for platform administrators to manage user accounts, resolve disputes, monitor activity, and perform other administrative tasks. This dashboard should provide insights into the platform's usage and allow for effective management.

IV. SYSTEM STUDY

Online tutor web applications can be a valuable resource for studying web development. Whether you're a beginner looking to start learning web development or an experienced developer seeking to enhance your skills, online tutoring can provide personalized guidance and support. Here are some ways online tutors can assist with studying web development:

Fundamentals of web development: Online tutors can help you grasp the foundational concepts of web development, including HTML, CSS, and JavaScript. They can explain the core principles, syntax, and best practices, ensuring you have a strong understanding of the basics.

Project-based learning: Tutors can guide you through project-based learning, where you work on real-world web development projects to apply your knowledge. They can provide step-by-step instructions, code reviews, and feedback to help you develop practical skills and build a portfolio.

Frameworks and libraries: If you're interested in learning specific web development frameworks or libraries such as React, Angular, or Vue.js, online tutors with expertise in these areas can teach you the fundamentals and guide you through building projects using these technologies.

Troubleshooting and problem-solving: Web development often involves encountering errors and debugging code. Online tutors can assist you in troubleshooting issues, understanding error messages, and developing effective problem-solving strategies.

Industry trends and best practices: Web development is a dynamic field, and staying updated with the latest trends and best practices is crucial. Tutors can help you explore modern web development techniques, responsive design, accessibility standards, and other relevant topics.

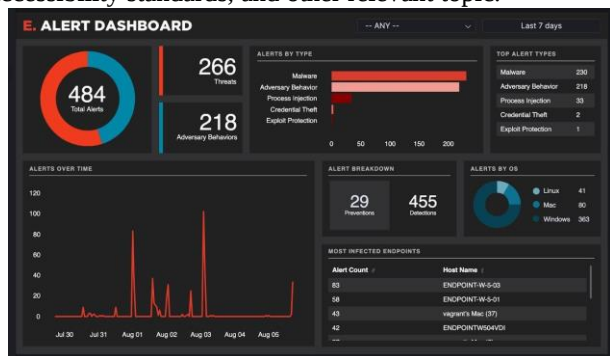


Figure 4 - Security Dashboard

V. SOFTWARE SPECIFICATION

System study in the context of online tutor web development refers to the process of analyzing and understanding the requirements, objectives, and constraints of the system being developed. It involves gathering information about the current system (if any), identifying user needs, and defining the scope of the online tutor web application. Here are some key aspects of system study in online tutor web development

5.1. Front-End- Django 2. Back – End - Html

Background Study

Programming Languages: Familiarize yourself with popular web programming languages such as HTML, CSS, JavaScript, and server-side languages like Python, Ruby, PHP, or JavaScript-based frameworks like Node.js.

Front-End Development: Learn front-end development concepts and technologies such as responsive web design, user interface (UI) and user experience (UX) design, client-side scripting, and frameworks like React, Angular, or Vue.js.

Back-End Development: Gain knowledge of back-end development, including server-side scripting, database management systems (e.g., MySQL, PostgreSQL), API development, server deployment, and frameworks like Django, Ruby on Rails, or Express.js. **Web Development Frameworks:** Explore popular web development frameworks and libraries that simplify the development process and provide ready-to-use components and tools. Examples include Laravel, ASP.NET, Flask, or Ruby on Rails.

Version Control: Learn how to use version control systems like Git, which enables tracking changes, collaborating with other developers, and managing code repositories.

Web Security: Understand web security principles, including best practices for handling user authentication, data encryption, secure communication, and protection against common web vulnerabilities like cross-site scripting (XSS) and SQL injection.

Performance Optimization: Study techniques for optimizing web application performance, such as minimizing file sizes, leveraging caching mechanisms, optimizing database queries, and employing content delivery networks (CDNs).

MAIN MODULE	SUB MODULE	MS DYNAMICS (PROFESSIONAL)	VTIGER (COMMUNITY)	VTIGER (PROFESSIONAL)	SUGAR CRM (COMMUNITY)	SUGAR CRM (PREMIUM)	SUITECRM	ICView	Salesforce	Zoho
SALES	OPPORTUNITIES	✓	✓	✓	✓	✓	✓	✓	✓	✓
	LEADS	✓	✓	✓	✓	✓	✓	✓	✓	✓
	COMPETITORS	✓	✓	✓	✓	✓	✓	✓	✓	✓
	ACCOUNTS	✓	✓	✓	✓	✓	✓	✓	✓	✓
	PRODUCTS	✓	✓	✓	✓	✓	✓	✓	✓	✓
	ACTIVITIES	✓	✓	✓	✓	✓	✓	✓	✓	✓
INVOICING AND INVENTORY MANAGEMENT	ACTIVITY TRACKING	✓	✓	✓	✓	✓	✓	✓	✓	✓
	CONTRACTS	✓	✓	✓	✓	✓	✓	✓	✓	✓
	QUOTES	✓	✓	✓	✓	✓	✓	✓	✓	✓
	PURCHASE ORDERS	✓	✓	✓	✓	✓	✓	✓	✓	✓
	ORDERS	✓	✓	✓	✓	✓	✓	✓	✓	✓
	INVOICE	✓	✓	✓	✓	✓	✓	✓	✓	✓
SERVICES	PRICE BOOKS	✓	✓	✓	✓	✓	✓	✓	✓	✓
	ASSETS	✓	✓	✓	✓	✓	✓	✓	✓	✓
	SALES ORDER	✓	✓	✓	✓	✓	✓	✓	✓	✓
	PRODUCTS	✓	✓	✓	✓	✓	✓	✓	✓	✓
	SALES LITERATURE	✓	✓	✓	✓	✓	✓	✓	✓	✓
	VISITS	✓	✓	✓	✓	✓	✓	✓	✓	✓
SERVICES	CASES	✓	✓	✓	✓	✓	✓	✓	✓	✓
	SOLUTIONS	✓	✓	✓	✓	✓	✓	✓	✓	✓
	FORECAST	✓	✓	✓	✓	✓	✓	✓	✓	✓

Figure 5 - Performance Optimization

Web Standards and Accessibility: Familiarize yourself with web standards and guidelines, including HTML and CSS specifications, accessibility requirements, and ensuring compatibility across different browsers and devices.

APIs and Integration: Learn about integrating web applications with external APIs, such as payment gateways, social media platforms, mapping services, or third-party libraries.

Testing and Debugging: Understand testing methodologies and tools for web applications, including unit testing, integration testing, and end-to-end testing. Also, learn debugging techniques to identify and resolve issues in the code.

Continuous Integration and Deployment: Explore practices and tools for automating the build, testing, and deployment processes, such as continuous integration (CI) tools like Jenkins or Travis CI, and deployment tools like Docker or Heroku.

Web Performance Analysis: Gain knowledge of tools and techniques for analyzing web performance metrics, identifying bottlenecks, and optimizing website speed and responsiveness.

Requirement Gathering: The first step is to gather requirements by interacting with stakeholders, including potential users (tutors, students, administrators), educational institutions, and tutoring companies. This involves conducting interviews, surveys, and workshops to understand their needs, expectations, and pain points.

User Analysis: It is essential to analyze the characteristics and preferences of different user roles, such as tutors and students. This includes understanding their technical proficiency, preferred communication methods, scheduling preferences, and specific features they may require.

Existing System Evaluation: If there is an existing system or processes in place, it is important to evaluate and analyze them. This helps identify strengths, weaknesses, and areas for improvement.

Use Case Definition: Use cases describe the interactions and relationships between users and the system. They help define the system's functionality, features, and workflows. Use cases should cover common scenarios, such as user registration, profile management, scheduling, and virtual classroom interactions.

System Scope and Objectives: The system study phase clarifies the scope of the online tutor web application. It defines the goals, objectives, and expected outcomes of the system. It also helps identify any constraints, such as budget limitations, technical considerations, or time constraints.

Feasibility Study: A feasibility study assesses the viability and practicality of implementing the online tutor web application. It considers factors such as technical feasibility (can the required technology be implemented?), economic feasibility (is the project financially viable?), and operational feasibility (can the system be easily integrated into existing workflows?).

Technology Assessment: This involves evaluating different technologies, frameworks, and tools that can be used to develop the online tutor web application. It includes assessing factors such as scalability, security, compatibility, and ease of maintenance.

Risk Analysis: Risk analysis involves identifying potential risks and challenges that may arise during the development and implementation of the online tutor web application. This allows for proactive planning and mitigation strategies to address these risks.

User Interface Design: During the system study phase, it is important to consider the user interface (UI) design. This includes defining the layout, navigation, and visual elements of the application to ensure a user-friendly and intuitive experience.

Documentation: Throughout the system study phase, it is crucial to document all findings, requirements, use cases, and decisions made.

Industry Standards and Best Practices: Familiarize yourself with the industry standards and best practices relevant to your software domain. This may include software development methodologies (e.g., Agile, Waterfall), coding standards, quality assurance practices, and data security guidelines.

Technology Stack: Research and evaluate the various technologies, programming languages, frameworks, and tools available for developing software in your specific domain. Consider factors such as scalability, performance, security, and compatibility when selecting the appropriate technology stack.

VI. RESULTS AND DISCUSSION

Input Screen

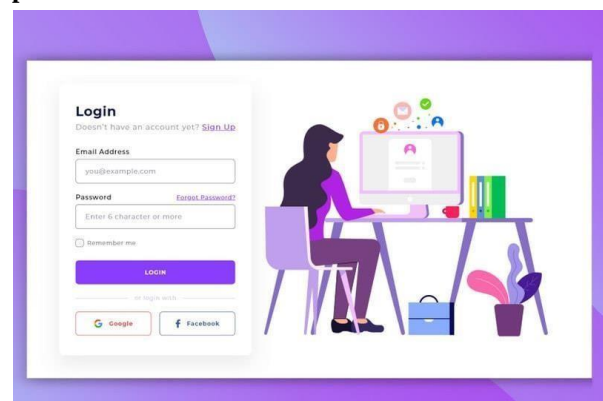


Figure 6 - Login

Design

Designing an online tutor web application involves considering various aspects, including the architecture, user experience, functionality, and scalability. Here's a high-level overview of the system design considerations for an online tutor web application:

Architecture: Client-Server Architecture: Adopt a client-server architecture where the client (web browser or mobile app) interacts with the server for data retrieval,

processing, and storage.

Cloud Infrastructure: Utilize cloud-based infrastructure (e.g., AWS, Azure, or GCP) for scalability, reliability, and easy deployment.

Management: User Authentication: Implement user authentication mechanisms, including registration, login, and password management, to ensure secure access to the application.

User Profiles: Allow users to create and manage their profiles, including personal information, contact details, and preferences.

Tutor and Student Interaction: Real-time Communication: Enable real-time communication between tutors and students through features like video conferencing, chat, and screen sharing.

Scheduling and Notifications: Implement scheduling functionality to allow tutors and students to book and manage sessions. Send notifications for upcoming sessions and reminders.

Content Sharing: Provide features for tutors to share study materials, resources, assignments, and feedback with students.

Learning Management System (LMS): Course Management: Implement a system to create, manage, and organize courses, lessons, and learning materials. Include features for curriculum planning and progress tracking.

Assessments and Grading: Incorporate assessment tools, such as quizzes, assignments, and exams, and implement a grading system to evaluate student performance.

Progress Tracking: Enable students and tutors to track progress, view completed lessons, and monitor performance through visual representations and analytics.

Payment and Billing: Payment Gateway Integration: Integrate a secure payment gateway to facilitate transactions between students and tutors.

Table Design

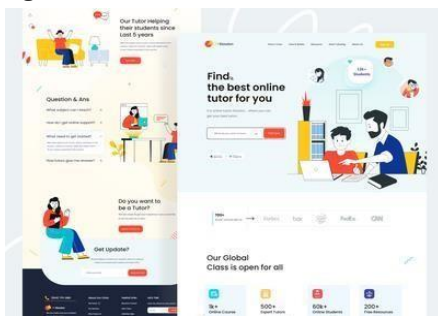


Figure 7 - Example Table

When designing the database for an online tutor web application, you'll need to consider the tables and their relationships to efficiently store and manage the application's data.

Here's a suggestion for the table design in an online tutor web application:

User Table: user_id (Primary Key) username password (encrypted/hashed) email role (e.g., student, tutor, admin) additional user information (e.g., name, contact details)

Tutor Table: tutor_id (Primary Key) user_id (Foreign Key referencing the User Table) bio (biography or introduction) specialization (e.g., math, science, languages)

Student Table: student_id (Primary Key) user_id (Foreign Key referencing the User Table) Course Table: course_id (Primary Key) tutor_id (Foreign Key referencing the Tutor Table) title description creation_date Lesson

Table: lesson_id (Primary Key) course_id (Foreign Key referencing the Course Table) title content video_url (optional) lesson_order (to maintain the order of lessons within a course)

Enrollment Table: enrollment_id (Primary Key) student_id (Foreign Key referencing the Student Table) course_id (Foreign Key referencing the Course Table) enrollment_date

Session Table: session_id (Primary Key) tutor_id (Foreign Key referencing the Tutor Table) student_id (Foreign Key referencing the Student Table) course_id (Foreign Key referencing the Course Table) start_time end_time session_notes (optional)

Payment Table: payment_id (Primary Key) student_id (Foreign Key referencing the Student Table) tutor_id (Foreign Key referencing the Tutor Table) amount payment_date

Output Screen

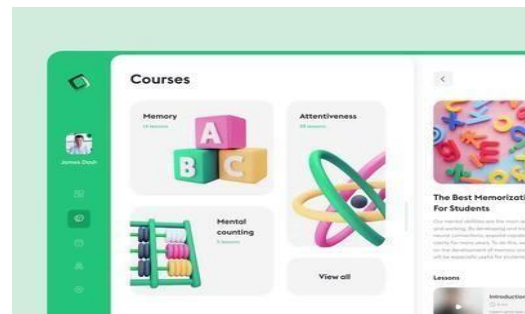


Figure 8 - Final Screen

The output screen or user interface (UI) design of an online tutor web application for web development should provide a user-friendly and intuitive experience. Here are some key components and considerations for the output screen in an online tutor web application:

Dashboard: Welcome message and user information (name, profile picture) Navigation menu or sidebar for easy access to different sections of the application (e.g., courses, sessions, profile) Overview of enrolled courses, upcoming sessions, and notifications

Course List: Display a list of available courses with relevant information such as course title, tutor name, and brief description Filter and search functionality to help users find specific courses Enrollment status indicator for each course (enrolled, in progress, completed)

Course Details: Detailed information about a selected course, including the course description, tutor information, and course curriculum Display lessons within the course, highlighting the lesson title, progress status, and completion status Option to start or resume a course, or mark it as completed

Lesson View: Present lesson content, including text, images, videos, or interactive elements. Clear navigation options to move between lessons or sections within a lesson. Progress tracking to indicate the user's completion status within the lesson and the overall course.

Communication and Collaboration: Provide real-time communication channels for students and tutors, such as chat or messaging features, to facilitate discussions and support. Integration with video conferencing tools for live tutoring sessions. Discussion forums or community features for peer-to-peer interaction and knowledge sharing.

Personal Profile: Allow users to manage their personal information, profile picture, and contact details. View and update their enrollment status, course progress, and session history. Provide options for account settings, including password management and email preferences.

Responsive Design: Ensure that the web application is responsive and adaptable to different screen sizes and devices, such as desktops, tablets, and mobile devices. Optimize the layout and components to provide a seamless experience across various screen resolutions and orientations.

Remember to incorporate clear and intuitive navigation, visually appealing design elements, and responsive interactions to enhance the user experience. Conduct user testing and gather feedback to iterate and improve the output screen design based on the specific needs and preferences of your target audience.

```
#output-container { width:
```

```
300px; height: 200px; background-color: #f1f1f1; padding:
```

```
10px;
```

```
font-family: Arial, sans-serif; font-size: 14px;
```

```
}
```



Figure 9 - Output of Online Tutor Website

Editing

Navigate to the Individual that need to edit. Edit the fields that need to be changed. Click Save to save a draft or Save and Post to finish editing.

VII. CONCLUSION

In conclusion, developing an online tutor website involves various elements and considerations to create a successful and user-friendly platform. Here are some key

points to keep in mind: User Experience, Tutor Profiles, Scheduling and Booking, Communication Tools, Payment Integration, Responsive Design, Security and Privacy, Testing and Feedback, Scalability, developing an online tutor website requires a combination of web development skills, user experience design, and knowledge of educational requirements. By considering these factors and implementing them effectively, you can create a successful online tutor platform that meets the needs of both students and tutors.

REFERENCE

- [1] Cavalcanti, A. P., Barbosa, A., Carvalho, R., Freitas, F., Tsai, Y. S., Gašević, D., & Mello, R. F. (2021). Automatic feedback in online learning environments: A systematic literature review. *Computers and Education: Artificial Intelligence*, 2.
- [2] Akobe, D., Popoola, S. I., Atayero, A. A., Oseni, O. F., & Misra, S. (2019). A web framework for online peer tutoring application in a smart campus. In *Computational Science and Its Applications-ICCSA: 19th International Conference, Saint Petersburg, Russia, July 1-4, 2019, Proceedings, Part V* 19, 316-326. Springer International Publishing.
- [3] Joshi, P., & Dewangan, D. S. (2021). Impact and Development of online education (E-learning) in India. *Journal of Contemporary Issues in Business and Government*, 27(1), 3450-3458.
- [4] Hrastinski, S., Stenbom, S., Benjaminsson, S., & Jansson, M. (2021). Identifying and exploring the effects of different types of tutor questions in individual online synchronous tutoring in mathematics. *Interactive Learning Environments*, 29(3), 510-522.
- [5] Wang, H., Tlili, A., Huang, R., Cai, Z., Li, M., Cheng, Z., ... & Fei, C. (2023). Examining the applications of intelligent tutoring systems in real educational contexts: A systematic literature review from the social experiment perspective. *Education and Information Technologies*, 1-36.
- [6] Alfaro, L., Rivera, C., Castaneda, E., Zuniga-Cueva, J., Rivera-Chavez, M., & Fialho, F. (2020). A review of intelligent tutorial systems in computer and web-based education. *International Journal of Advanced Computer Science and Applications*, 11(2).
- [7] Saiyad, S., Virk, A., Mahajan, R., & Singh, T. (2020). Online teaching in medical training: Establishing good online teaching practices from cumulative experience. *International Journal Of Applied And Basic Medical Research*, 10(3), 149-155.
- [8] Pratama, M. P., Sampelolo, R., & Lura, H. (2023). Revolutionizing Education: Harnessing the Power of Artificial Intelligence for Personalized Learning. *Klasikal: Journal of Education, Language Teaching and Science*, 5(2), 350-357.
- [9] Chung, S. H., & Tan, S. C. (2019). MENTOR-Intelligent Mobile Online Peer Tutoring Application for Face-to-Face and Remote Peer Tutoring. *ASCILITE Publications*, 386-391.
- [10] Seo, K., Tang, J., Roll, I., Fels, S., & Yoon, D. (2021). The impact of artificial intelligence on learner-instructor interaction in online learning. *International journal of educational technology in higher education*, 18(1), 1-23.