

**Report**

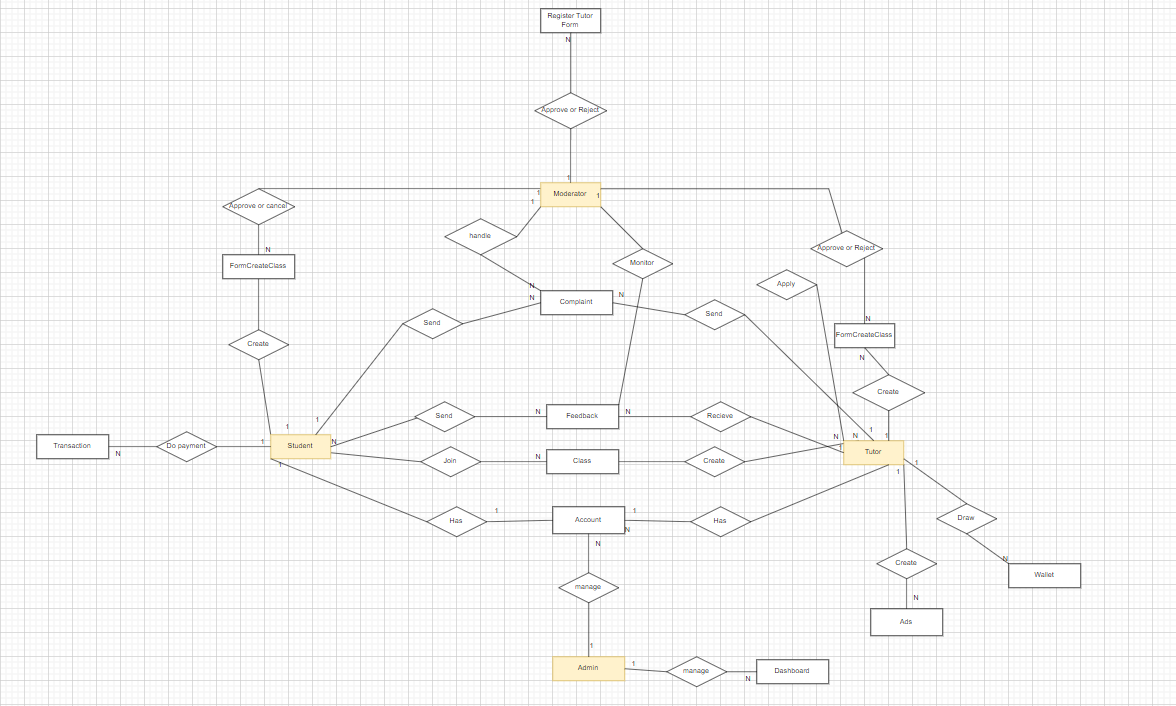
**On demand tutor**

– Ho Chi Minh, June 2024 –

**Table of Contents**

1. ERD
2. Function Requirement
3. Usecase List
4. Functional
5. Bussines rule
6. Screen flow
7. Non-functional
8. UI
9. Technical Requirement

# ERD

****

# II. Function Requirement

## 1. User Requirements

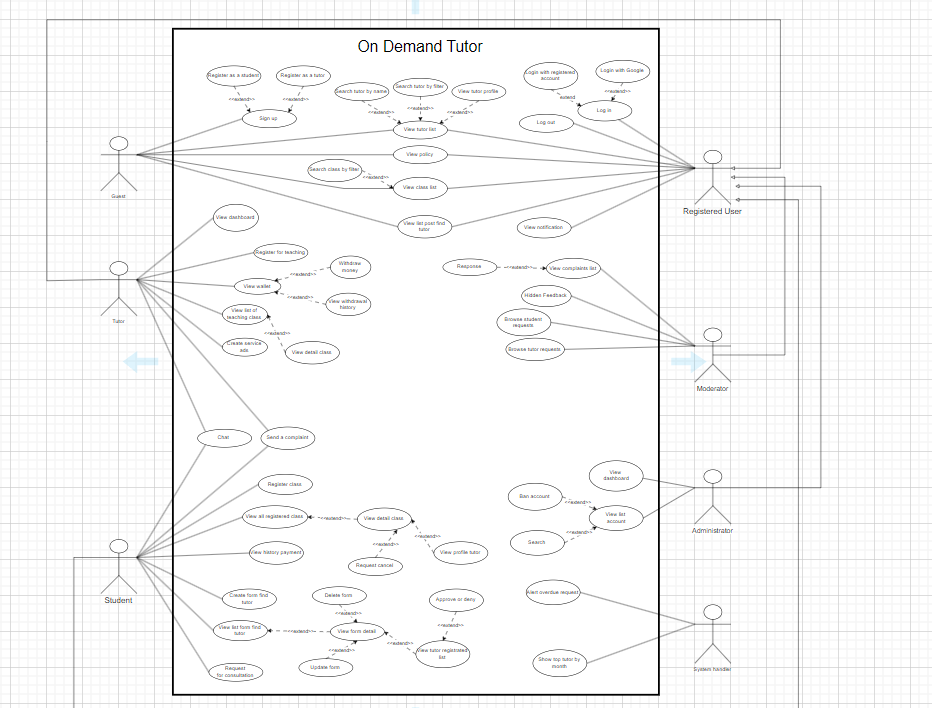
### 1.1 Actors

|  |  |  |
| --- | --- | --- |
| **#** | **Actor** | **Description** |
| 1 | **Tutors** | * They can indeed be individuals from various backgrounds, but our primary target is graduates who are looking to earn money and support themselves. These individuals must possess **credentials** to demonstrate their expertise, including educational qualifications, specialised skills and personal identification information to ensure trust and credibility. * When it comes to “students”, the main reason why we do this web app is to help them feel easier in studying with reasonable prices and immediately. |
| 2 | **Guests** | * Limited access to system features (web app). |
| 3 | **Student** | * The person who uses the application to rent a tutor to help them with their study problem. They can search for tutor by name, subject, chat with tutor, leave rating and feedback for tutor after rent (web applications) |
| 4 | **Tutor** | * User who uses the app to create their service to support students about the subject. They can create a rental service by hour, upload academic videos to advertisements for their service. (web app) |
| 5 | **Moderator** | * A person is responsible for content moderation and complaints on the website. |
| 6 | **Administrator** | * A privileged user who can manage the entire system. |

### 1.2 Use Cases

* Sign up as tutor
* Sign up as students
* Login with Google
* Login with registered account
* Forgot password
* Change password
* Logout
* Edit profile
* See announcement
* See tutor list
* Search tutor by character
* Search for tutors using filters
* View tutor profile
* See feedback tutor
* Request a tutor to teach
* Register to find a tutor
* See the list of tutors currently studying
* Cancel Tutor tuition
* View payment account information
* Pay
* View payment history
* Feedback
* Register as a tutor
* Refund
* View chatbox list
* View chatbox content
* Send Message
* See policies and regulations
* Submit a complaint
* See wallet
* Withdraw money
* View withdrawal history
* View the list of students currently teaching
* Create service ads
* See teaching schedule
* Create teaching schedule notes
* Delete lesson schedule notes
* View monthly statistics
* See complaints
* Hide feedback
* Censoring requests for tutors
* See list of requests for tutoring
* Send email to request an interview
* Show list of top lecturers of the month
* Notification
* View dashboard
* View account list
* Ban accounts
* Search user accounts

#### 1.2.1 Diagram(s)



#### 2.2.2 Descriptions

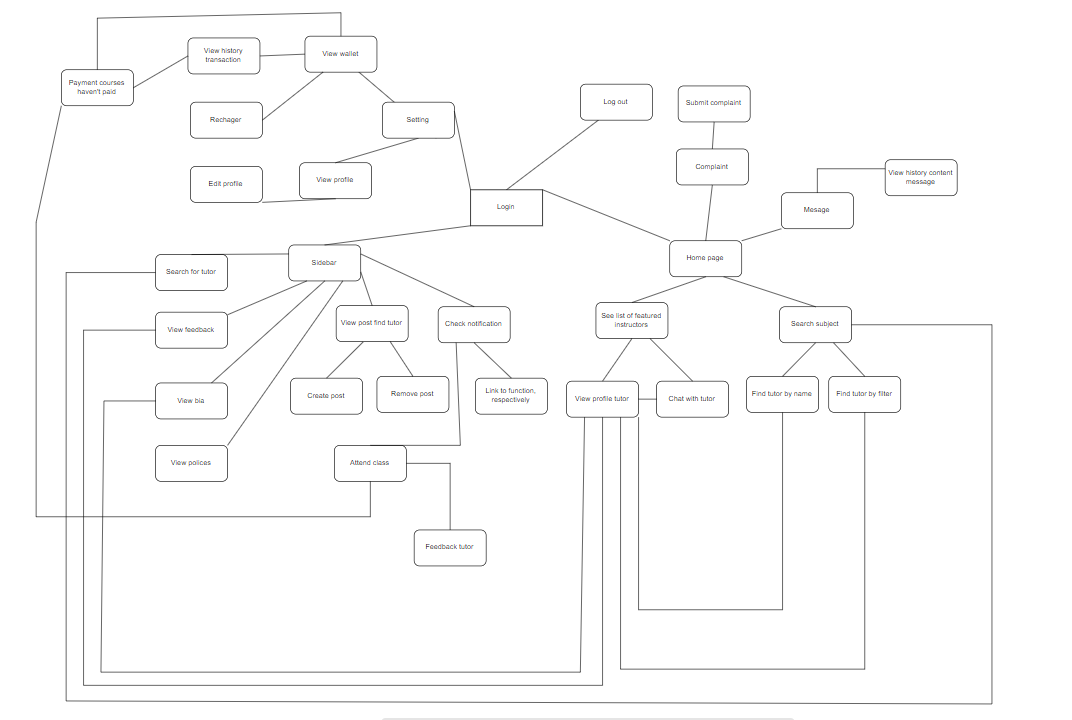
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Use Case** | **Actors** | **Use Case Description** | **Business Rule** |
| 01 | **Sign up as tutor** | Tutor, guest | Users can register as a tutor to provide tutoring services on the platform. | - A user's profile must be verified and approved by an administrator before they can become an instructor.  - Each instructor account can only be associated with a single user. |
| 02 | Sign up as students | Student, guest | Users can register as students to request and receive tutoring services. | - Each student account can only be associated with a single user.  - Students must verify their email or phone number to activate their account. |
| 03 | Login by Google | Tutor, Studen,  guest | Users can log in using their Google account. | - Users must have a valid Google account.  - If the user does not have an account on the platform, a new account will be created automatically.  - Google account information will be used to fill in the user's basic information. |
| 04 | Login by registered account | All | Users can log in using an account they have registered on the platform. | - The user must enter the correct registered username and password.  - Passwords must comply with complexity requirements (at least 8 characters, including uppercase letters, lowercase letters, numbers and special characters).  - After 3 unsuccessful login attempts, the account will be locked and the user must go through the forgotten password process. |
| 05 | Forgotten password | Tutor, Student | Users can reset a forgotten password | - Users must provide registered email or phone number.  - The system will send a confirmation code to the registered email or phone number.  - The user must enter the correct confirmation code to be allowed to reset the password. |
| 06 | Change password | Tutor, Student | Registered users can change their password. | - Users must enter the correct current password before setting a new password.  - The new password must comply with complexity requirements (at least 8 characters, including uppercase letters, lowercase letters, numbers and special characters).  - The system will store password history and will not allow users to use old passwords. |
| 07 | Logout | Member | Users can log out of their account. | - When a user logs out, the system will end the current session and will not allow unauthorized access.  - If the user is inactive for a certain period of time, the system will automatically log out. |
| 08 | Edit profile | Member | Users can edit and update their profile information | - Users can only edit personal information such as full name, email, phone number.  - For lecturer accounts, they can also update information about qualifications and teaching experience.  - The system will save the history of changes to personal information. |
| 09 | View notification | Member | Users can view notifications related to their activities on the platform. | - Users can only edit personal information such as full name, email, phone number.  - For lecturer accounts, they can also update information about qualifications and teaching experience.  - The system will save the history of changes to personal information. |
| 10 | View tutor list | member | Everyone can view a list of available tutors. | - The list of lecturers must be classified and arranged based on criteria such as subjects, price, reviews, and experience.  - Users can apply filters to narrow down the list of instructors.  - Instructor List only shows accounts that have been verified and approved by administrators. |
| 11 | Find tutor by name | Member | Everyone can search for tutors by entering keywords. | - Users can search for tutors by entering the tutor's full name or partial name.  - The search results should display the tutor's name, subject expertise, rating, and availability.  - Users can click on a tutor's name to view their detailed profile. |
| 12 | Find tutor by filter | Member | Everyone can filter tutor search results using various criteria. | - Users can apply filters to search for tutors, such as:   * Subject expertise (e.g., math, English, science) * Location (e.g., city, state, zip code) * Availability (e.g., weekdays, weekends, specific time slots) * Rating (e.g., 4 stars and above) * Hourly rate (e.g., $20-$50 per hour)   - The search results should display the filtered tutors' information, including name, subject expertise, rating, and availability.  Users can sort the search results by various criteria, such as rating, hourly rate, or availability.  - Users can select a tutor from the search results to view their detailed profile. |
| 13 | View tutor profile | Member | Users can view detailed profiles of individual tutors. | **- When a user clicks on a tutor's name, they should be taken to the tutor's detailed profile page.**  **- The tutor's profile should include the following information:**   * **Full name** * **Subject expertise** * **Educational background** * **Teaching experience** * **Hourly rate** * **Schedule availability** * **User reviews and ratings** * **Profile picture (if available)**   **- Users should be able to request a booking or schedule a consultation with the tutor directly from the profile page.**  **- The profile page should also display any additional information the tutor has chosen to provide, such as teaching methodologies, student testimonials, or teaching certifications.** |
| 14 | View feedback about tutor | All | Users can view feedback and ratings provided for tutors. | - Users can view detailed information about instructors, including personal information, qualifications, experience, and service prices.  - Users can view reviews and feedback about instructors from people who have used the service. |
| 15 | Indicate tutor | Student | Students can request a specific tutor to provide tutoring services. | **- Students must select a specific instructor and send a request to that instructor.**  **- The request must include information about the subject, time, location and other requirements.**  **- Instructors can accept or decline teaching requests.** |
| 16 | Request tutor | Student | Students can register to receive tutoring services | **- Students must provide information about subjects, time, budget and other requirements.**  **The system will search and suggest instructors that match the student's requirements.**  **- Students can choose and register with one or more instructors.** |
| 17 | View all tutor | Student | Students can view a list of tutors they are currently working with. | **- Users should be able to view a list of all available tutors in the system.**  **- The list should display the tutor's name, subject expertise, rating, availability, and hourly rate.**  **- Users can apply filters and sort the list of tutors similar to the "Find Tutor by Filter" functionality.**  **- From the list, users can click on a tutor's name to view their detailed profile..** |
| 18 | Cancel tutor | Student | Students can cancel their enrollment with a tutor. | **- Users should be able to cancel a scheduled tutoring session.**  **- The cancellation should be subject to the platform's cancellation policy, which may include fees or restrictions based on the timing of the cancellation.**  **- Users should be able to view the cancellation policy and any applicable fees before confirming the cancellation.**  **- The system should notify the tutor about the cancellation and update their availability accordingly.** |
| 19 | View information about account payment | Student | Users can view their payment account information. | **- Users should be able to view their account payment information, including:**   * **Payment method (e.g., credit card, PayPal)** * **Billing address** * **Payment history**   **- Users should be able to update their payment information as needed.** |
| 20 | Payment | Student | Users can make payments for tutoring services. | **- Users should be able to make payments for tutoring sessions or other platform services.**  **- The payment process should be secure and integrated with a payment gateway.**  **- Users should receive a confirmation of their payment, including a receipt or transaction ID.** |
| 21 | History payment | Student | Users can view their payment history | **- Users should be able to view their payment history, including:**   * **Date of payment** * **Amount paid** * **Reason for payment (e.g., tutoring session, platform fee)**   **- Payment method used**  **- Users should be able to filter and sort the payment history based on various criteria, such as date range or payment type.** |
| 22 | Feedback tutor | Student | Users can provide feedback on their tutoring experiences. | Automatically log out after 30 minutes |
| 23 | Become tutor | Guest | Apply to become a tutor on the platform | - The platform should review the application and either approve or reject the user's request to become a tutor.  - Approved tutors should be able to create and manage their profiles, update their availability, and receive booking requests from users. |
| 24 | Back money | Student | Users can request refunds for tutoring services | - Users can request a refund for tutoring services used.  - Refund requests are subject to review and approval based on the platform's refund policies.  - The refund amount will be sent back to the user's account within 7 business days. |
| 25 | History object chat | Student, tutor | Users can view their chat history with other users. | - Users can view conversation history with other users on the platform.  - Conversation history will be stored for 6 months. |
| 26 | View content chat | Student, tutor | Users can view the content of their chat conversations. | - Users can view the content of conversations with other users.  - The content of the conversation will be confidential and only the participating parties will have access. |
| 27 | Send message | Student, tutor | Users can send messages to other users on the platform. | - Users can send messages to other users on the platform.  - The message will be delivered to the recipient within 5 minutes.  - Users cannot send messages more than 5 times in 1 minute. |
| 28 | View polices and regulations | All | Users can view the platform's policies and regulations. | - Users can view the platform's policies and regulations.  - Policies and regulations will be updated periodically and users will be notified of changes. |
| 29 | Submit complaint | Student, Tutor | Submit complaints or grievances. | - Users can submit complaints related to services or activities on the platform.  - Complaints will be processed within 5 business days.  - Users will receive feedback on the complaint processing status. |
| 30 | View balance | Student, Tutor | Users can view their account balance. | - Account balance will be updated in real time. |
| 31 | Withdraw | Student, tutor | Users can withdraw funds from their account. | - Users can withdraw money from their account.  - Minimum withdrawal amount is 50,000 VND.  - Maximum withdrawal amount is 5,000,000 VND per time.  - Withdrawal request processing time is 2 business days. |
| 32 | History of withdraw | Student, tutor | Users can view their withdrawal history. | - Withdrawal history will be stored for 12 months. |
| 33 | View list student is currently studying in class | Tutor, Admin, Moderator | View a list of students they are currently teaching. | - This list will be updated in real time. |
| 34 | Bia | Tutor | Tutors can create advertisements for their tutoring services. | - Ads must comply with the platform's ad content policies and guidelines.  - Ads will be approved before being displayed. |
| 35 | View schedule teach | Tutor | Tutors can view their teaching schedules. | - The teaching schedule will be updated in real time. |
| 36 | Note schedule teach | Tutor | Tutors can create notes related to their teaching schedules. | - These notes are only visible to the tutor and do not affect the teaching schedule. |
| 37 | Remove schedule teach | Tutor | Tutors can delete their schedule notes. | - Deleting notes does not affect the teaching schedule. |
| 38 | View monthly statistic | Admin | View monthly platform usage statistics. | - The statistics should be displayed in a clear and organized manner, with the ability to filter and export the data as needed. |
| 39 | View complaint | Admin | View user complaints and grievances. | - The complaint management system should include:   * Ability to view the complaint details, including the user's name, contact information, and the nature of the complaint * Tools to investigate and respond to the complaint * Functionality to escalate or assign complaints to relevant team members * Reporting and analytics on complaint trends and resolutions |
| 40 | Hide feedback | Moderator | Hide user feedback. | - This functionality should be used in cases where the feedback is deemed inappropriate, false, or violates the platform's policies.  - When feedback is hidden, the platform should notify the tutor and provide a reason for the action.  - Administrators should maintain a record of all hidden feedback and the justification for the action. |
| 41 | Censoring requests for tutors | Moderator | Review and approve student requests for tutors. | - The censorship process should involve screening the applicant's information, such as educational background, teaching experience, and subject expertise, to ensure they meet the platform's requirements.  - Administrators should have the ability to reject or approve tutor applications based on the platform's policies and quality standards.  - Rejected applicants should be notified of the decision and provided with a reason for the rejection. |
| 42 | Censoring requests for become to tutors | Moderator | View a list of tutor applications. | - The censorship process should involve screening the applicant's information, such as educational background, teaching experience, and subject expertise, to ensure they meet the platform's requirements.  - Administrators should have the ability to reject or approve tutor applications based on the platform's policies and quality standards.  - Rejected applicants should be notified of the decision and provided with a reason for the rejection. |
| 43 | Send email to request an interview | Admin | Platform administrators can send interview requests to tutor applicants. | - The email should include information about the platform, the requirements for becoming a tutor, and instructions for scheduling the interview.  - The platform should maintain a record of all interview requests sent and the corresponding tutor applications or responses. |
| 44 | Show list of top lecturers of the month | Admin | Platform administrators can display a list of top-performing tutors for the month. | - The list should be based on metrics such as the number of sessions, student ratings, and revenue generated.  - Users should be able to access this list and view the top tutors' profiles, availability, and contact information. |
| 45 | View dashboard | Admin | Platform administrators can view a dashboard with platform usage statistics and analytics. | - The dashboard should include metrics such as total users, active tutors, bookings, revenue, and user engagement.  - Administrators should be able to customize the dashboard layout and the data displayed, as well as export the data for further analysis. |
| 46 | View account list | Admin | Platform administrators can view a list of all registered user accounts. | - The account list should display relevant information, such as the user's name, email, role (e.g., student, tutor), and account status (active, disabled).  - Administrators should have the ability to disable user accounts, either temporarily or permanently, based on platform policies or user violations.  - When an account is disabled, the user should be notified, and their access to the platform should be revoked. |
| 47 | Disable account | Moderator | Platform administrators can disable user accounts. | - The "Disable Account" functionality should only be accessible to platform administrators, not regular users.  - Administrators should have the necessary permissions to disable user accounts on the platform |
| 48 | Search user account | Admin | Platform administrators can search for and view individual user accounts. | - The search functionality should provide a filtered view of the user accounts, allowing administrators to quickly find and manage specific users.  - Administrators should be able to perform actions such as viewing account details, updating account information, or disabling accounts directly from the search results. |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

## 2. Functional Requirements

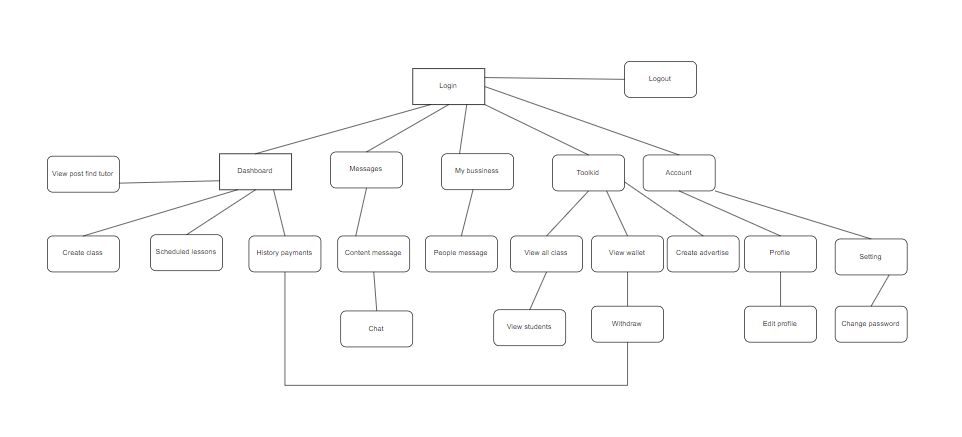
### 2.1 System Functional Overview

#### 2.1.1 Screens Flow

Student



Tutor



#### 3.1.2 Non-Screen Functions

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Feature** | **System Function** | **Description** |
| 1 | Notification | Log-in fail | Server overload, network failure, incorrect password when logging in, the screen displays a corresponding error message to the user. |
| 2 | Security | Request security | For new users who have not done much security, the screen displays a message after logging in asking the user to set up some account security to make it safer. |
| 3 | Convenient | New feature | The screen introduces features for premium accounts, performing updates to premium accounts. |
| 4 | Authenticity | Log-out | When making a transaction, the screen displays whether you are sure to perform this function or not. You can choose to continue or cancel. |
| 5 |  |  |  |

## 3. Non-Functional Requirements

### 3.1 External Interfaces

1. Performance:

The application must have a fast response time, no more than 3 seconds to display the list of suitable instructors.

The system must be able to handle a maximum of 1000 class booking requests simultaneously without affecting the user experience.

The application must be scalable to accommodate an increasing number of users in the future.

2. Usability:

The user interface must be easy to use, intuitive, and user-friendly.

The process of registration and searching for instructors must be simple, with no more than 5 steps.

The application must have detailed user guides to help new users quickly familiarize themselves.

3. Reliability:

The application must have a minimum availability of 99.9% during its operational time.

The system must be able to recover quickly after an incident, no more than 1 hour.

User data must be secured and backed up regularly to prevent data loss.

4. Security:

The application must comply with security standards such as HTTPS, data encryption, and two-factor authentication.

Users must be warned about the risks related to information security and privacy.

The system must be able to detect and prevent attacks such as DDoS, SQL Injection, and Cross-site Scripting.

5. Scalability:

The application must be able to scale its infrastructure to meet the increasing user demand in the future.

The system must be able to automatically scale up/down resources (such as CPU, RAM, storage) based on usage needs.

The application architecture must be designed to be flexible, with the ability to expand individual components (modular) when necessary.

6. Maintainability:

The source code must be written clearly, following coding standards, and have comprehensive technical documentation.

The application must be easy to update, fix bugs, and deploy new versions.

The system must have monitoring tools and automatic error reporting to support the maintenance process.

7. Reusability:

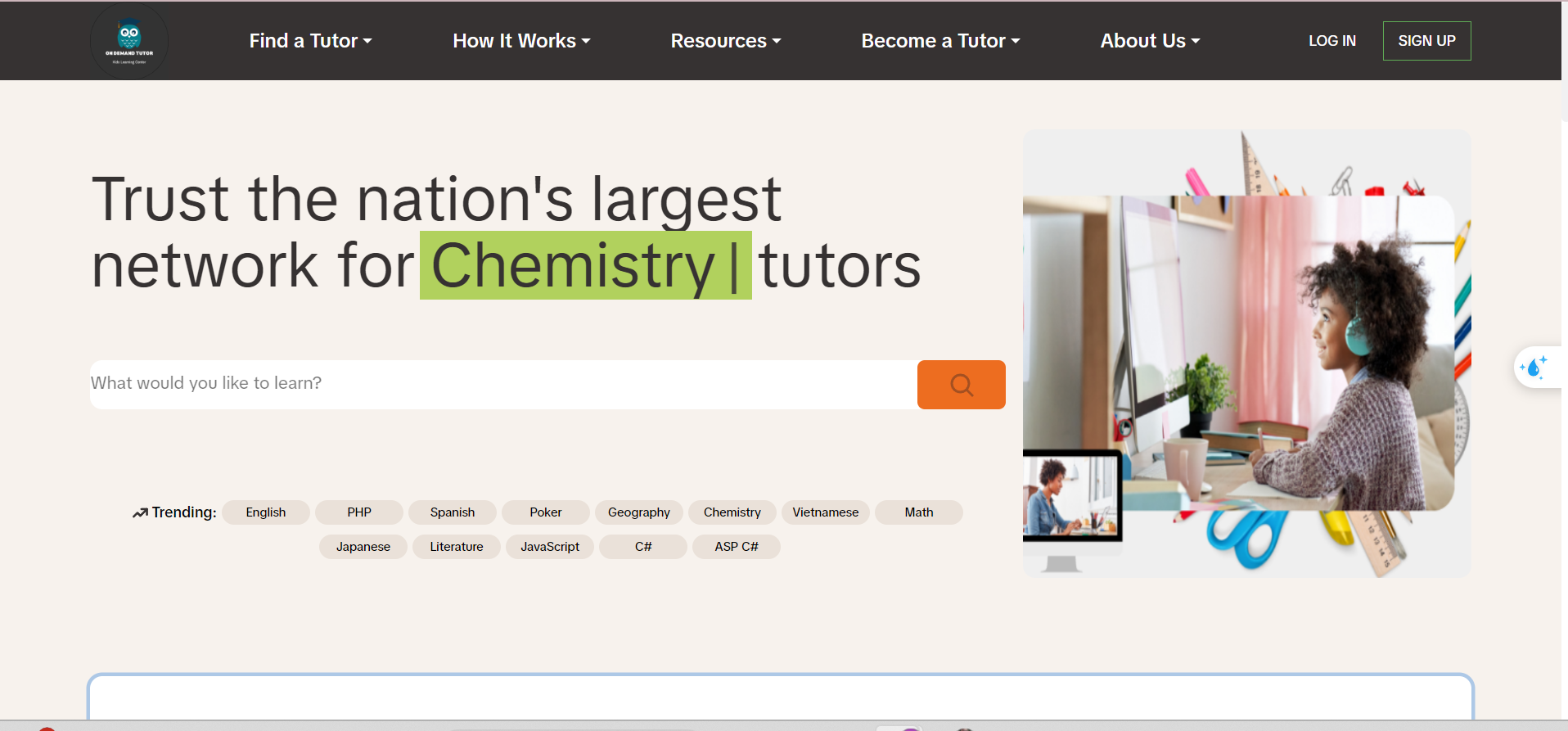
The components within the application must be designed to be reusable in other projects in the future.

The code structure and libraries used must adhere to object-oriented design principles.

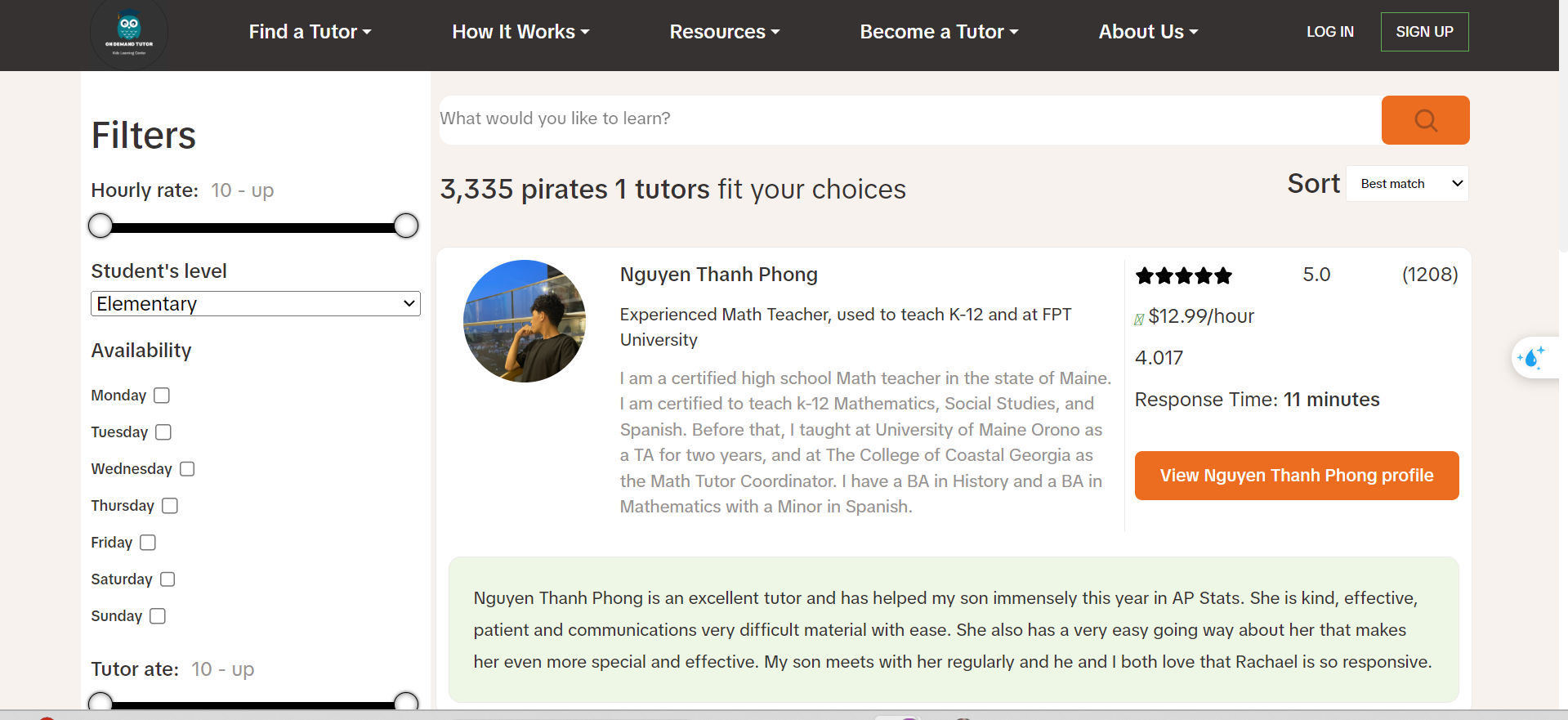
The application must be able to integrate easily with other systems through standard APIs.

III. UI

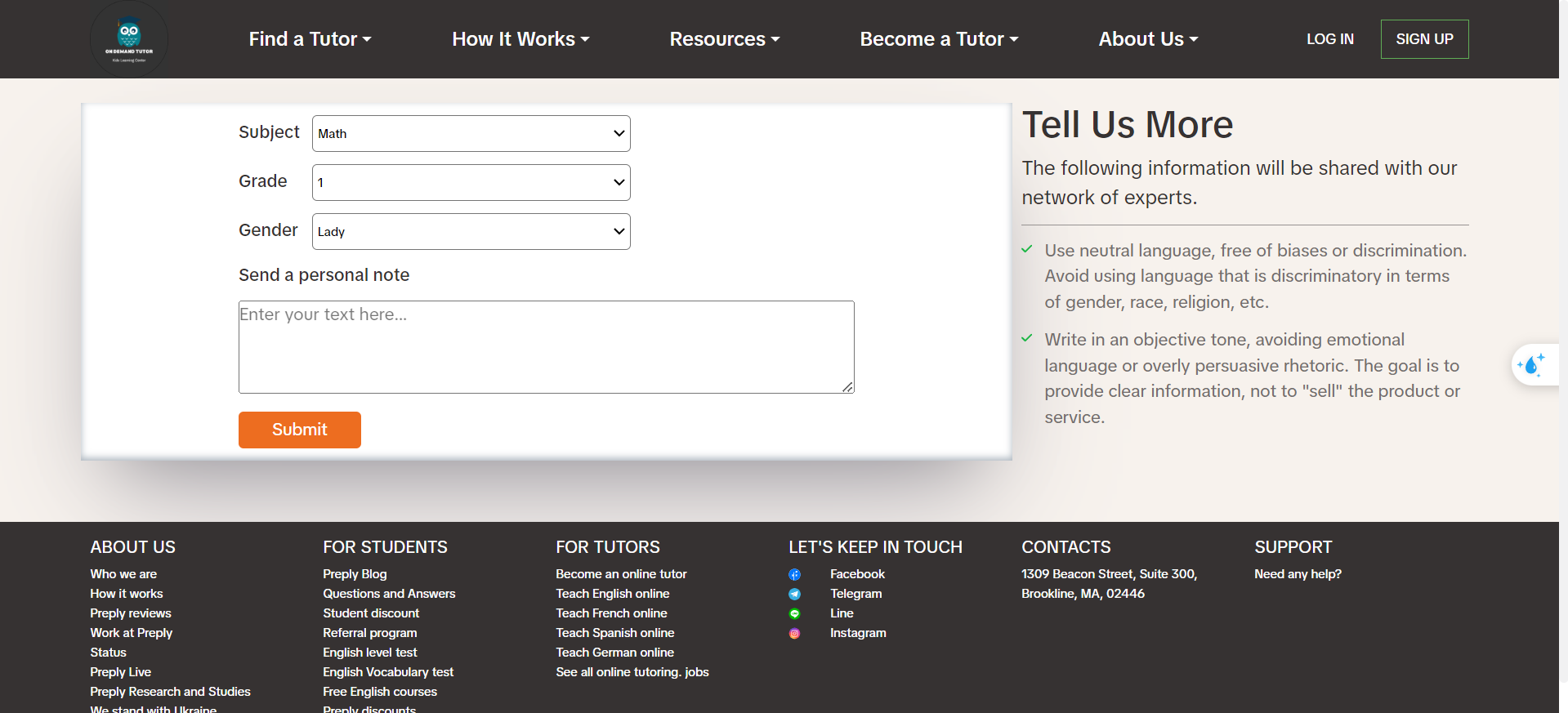
Home page



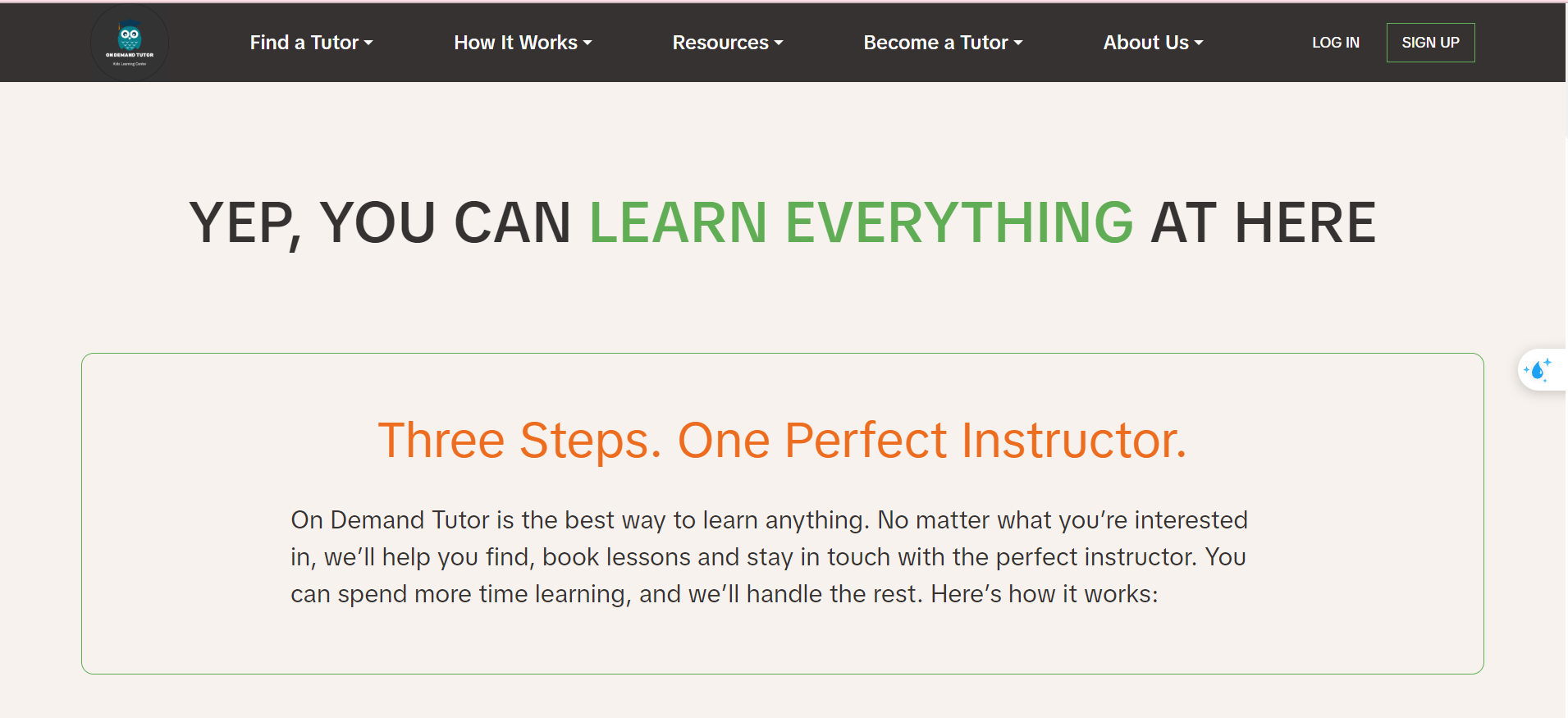
Find tutor page



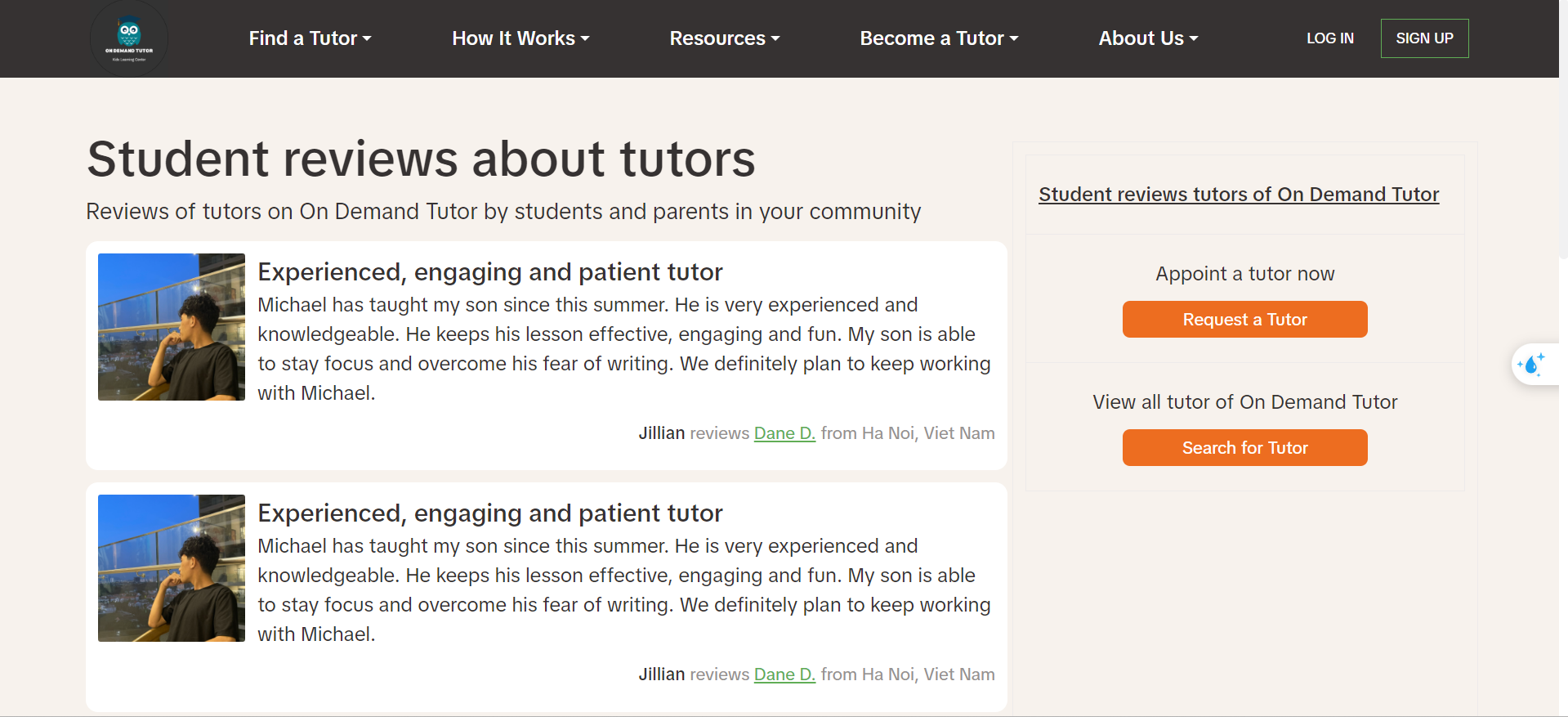
Request tutor page



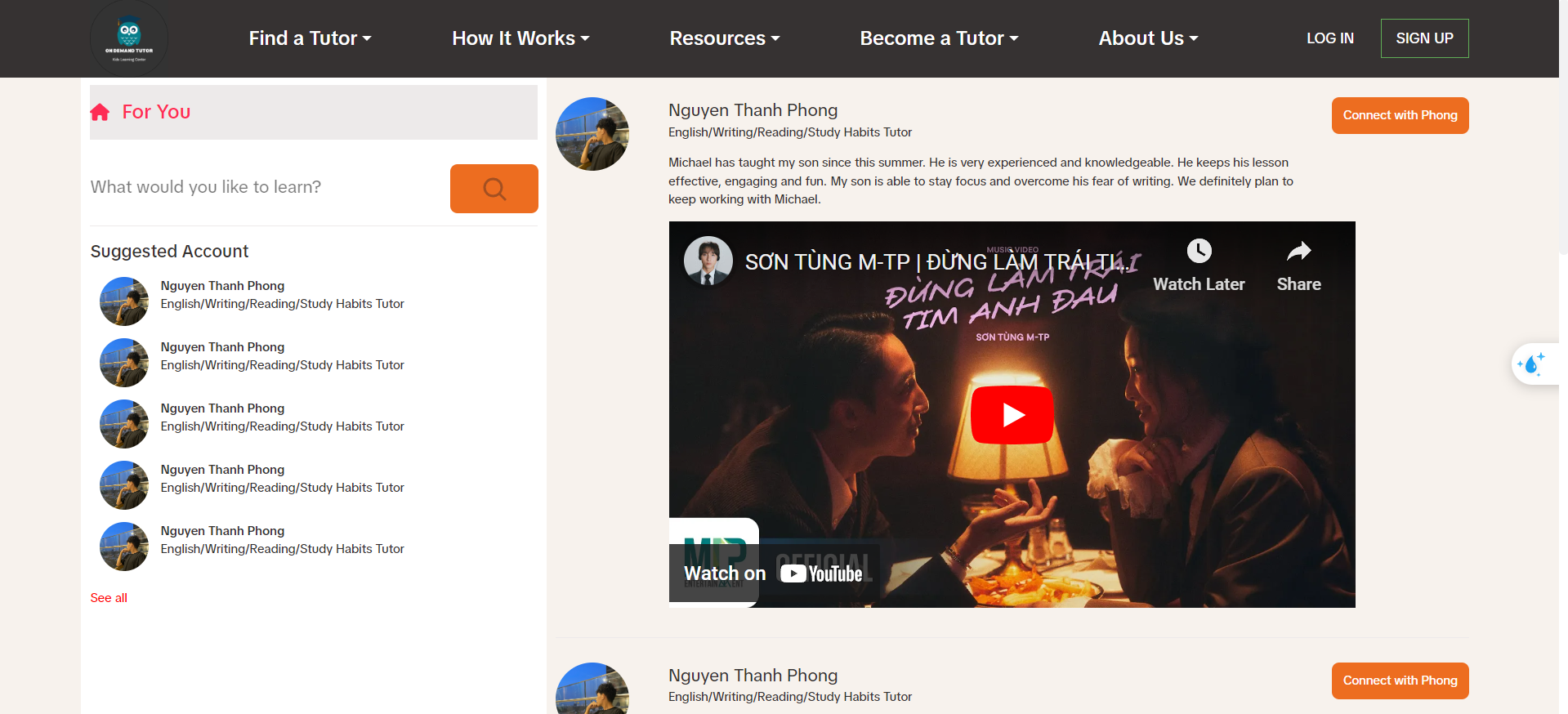
Introduc request tutor page



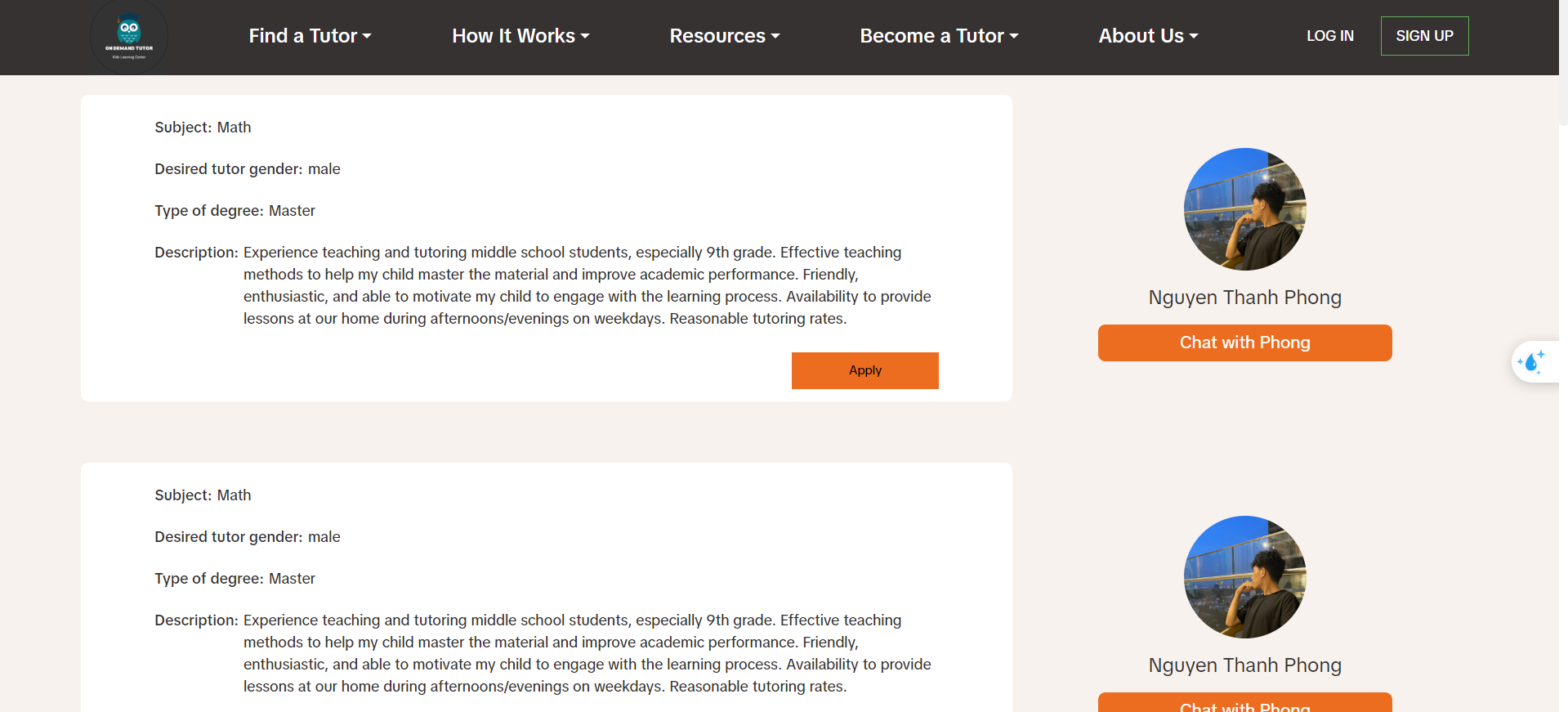
Feedback page



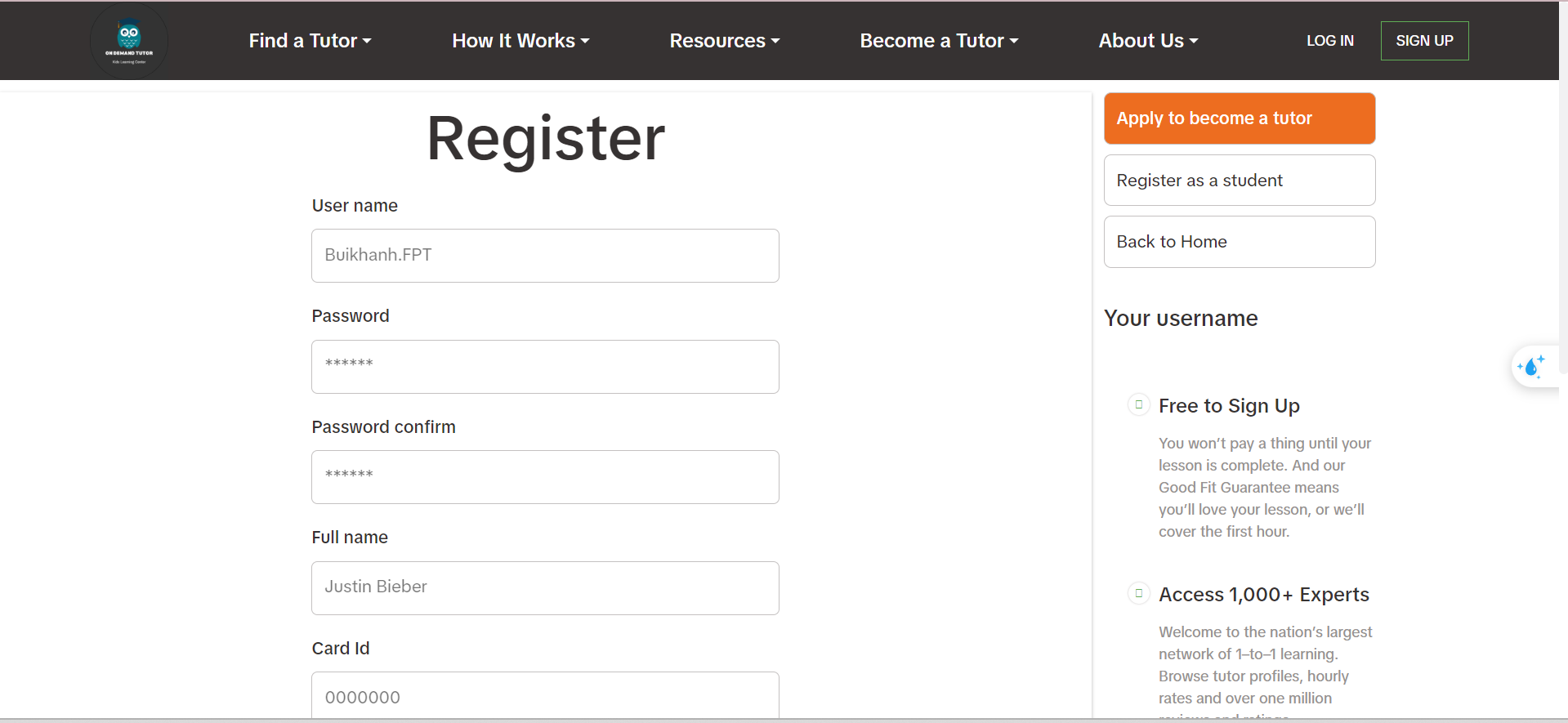
Bia page



Post page



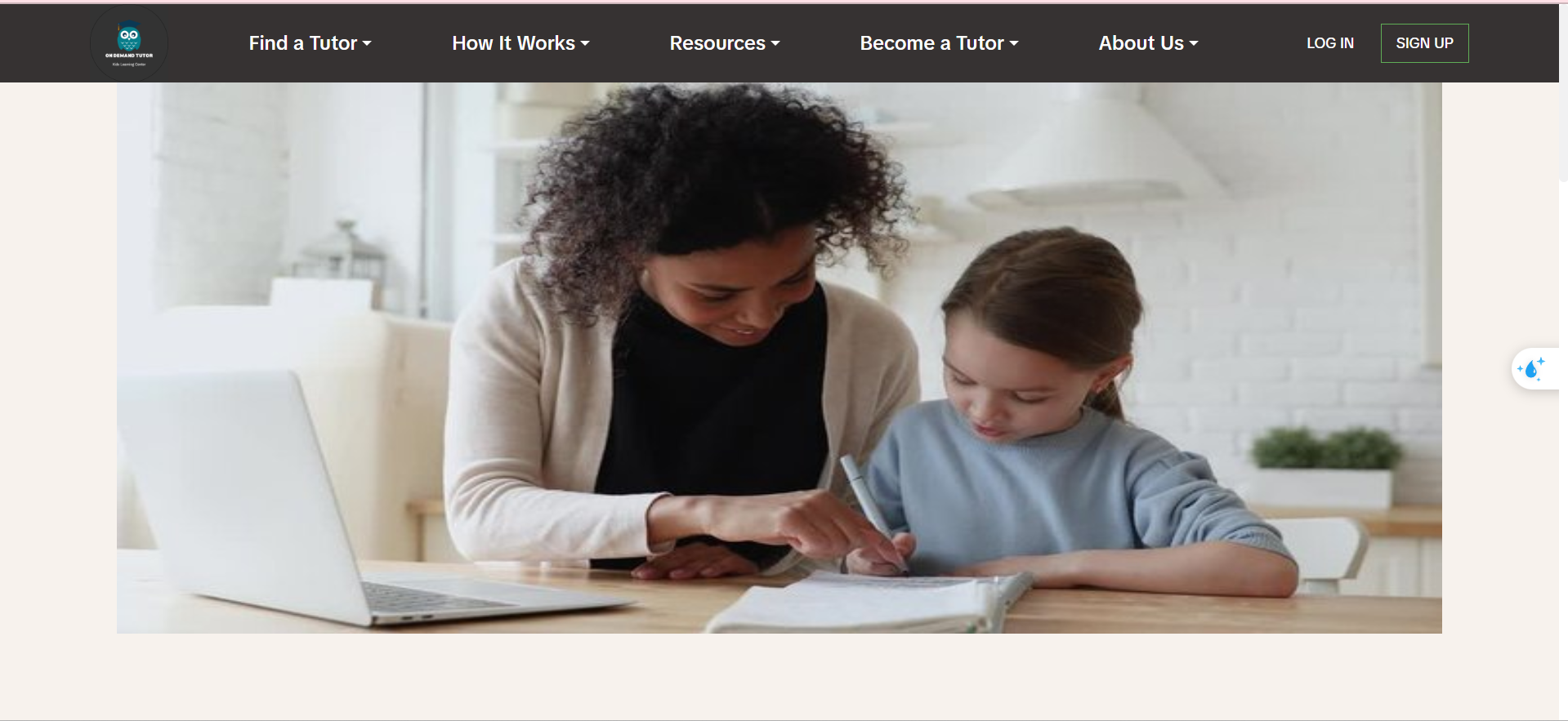
Become tutor page



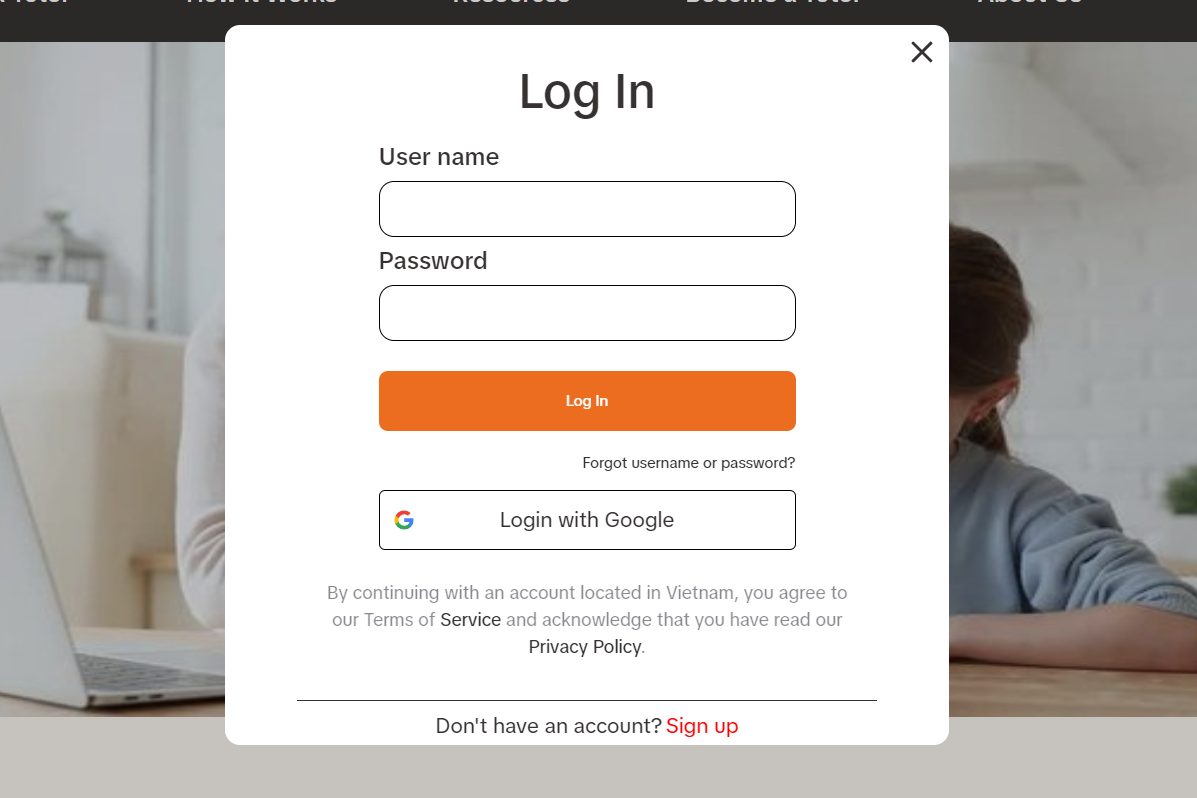
Introduce become tutor page



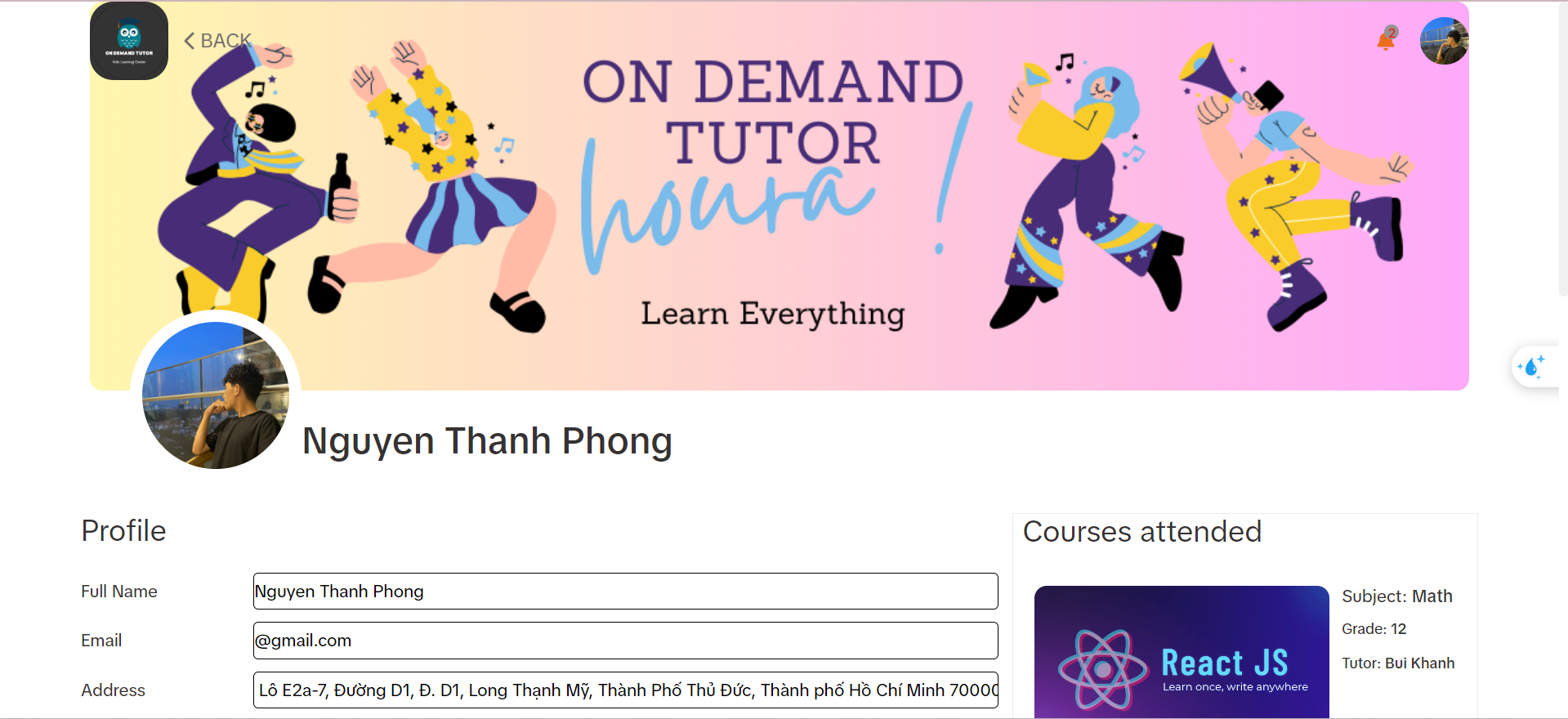
About us page



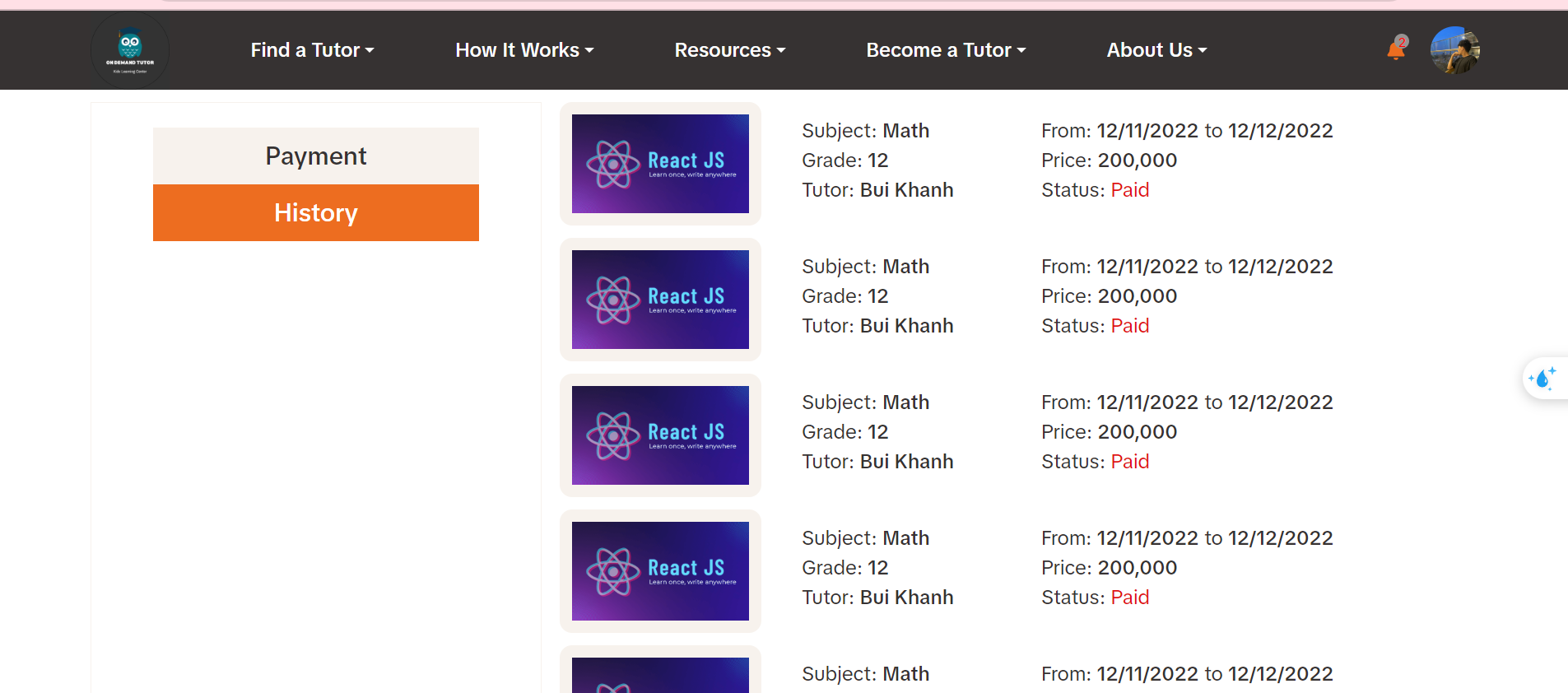
Modal login



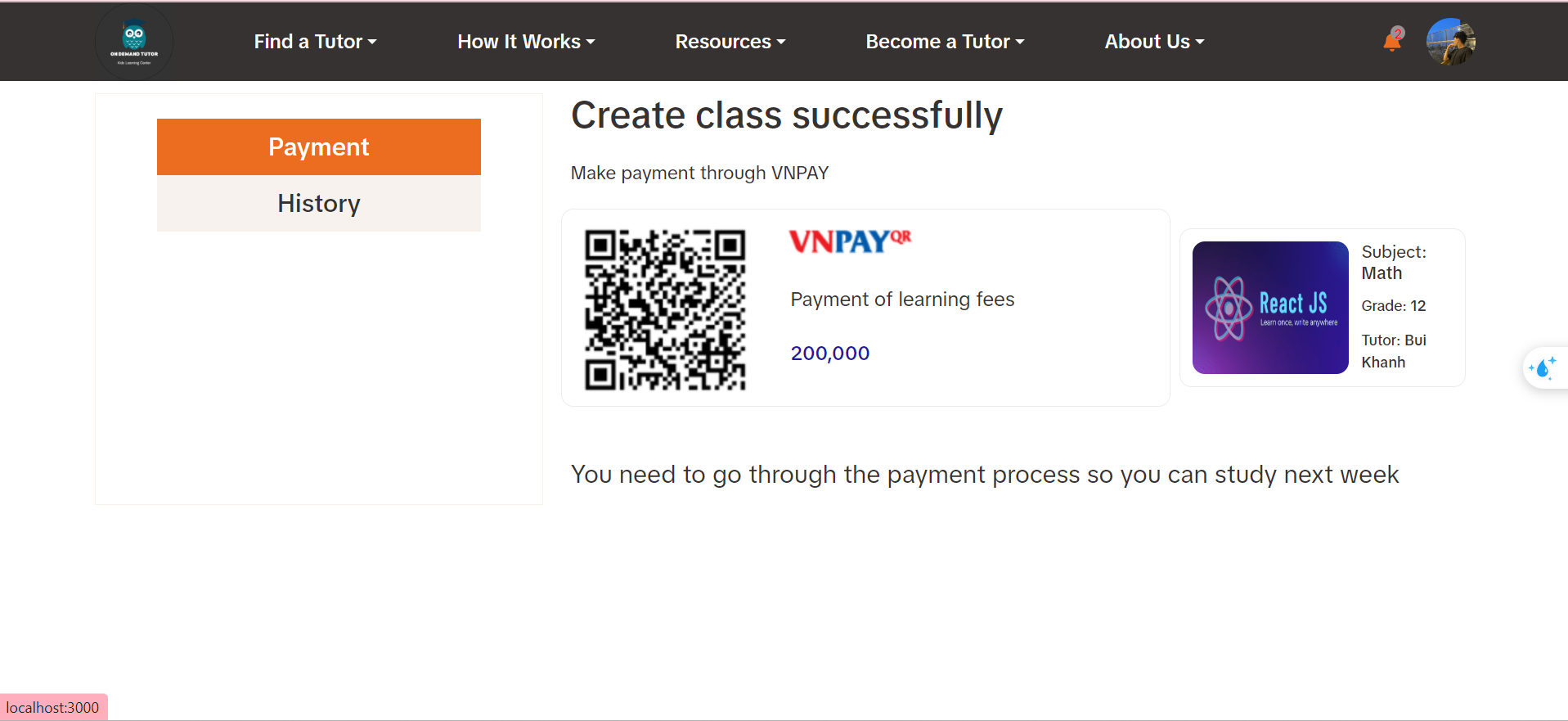
Profile studet page



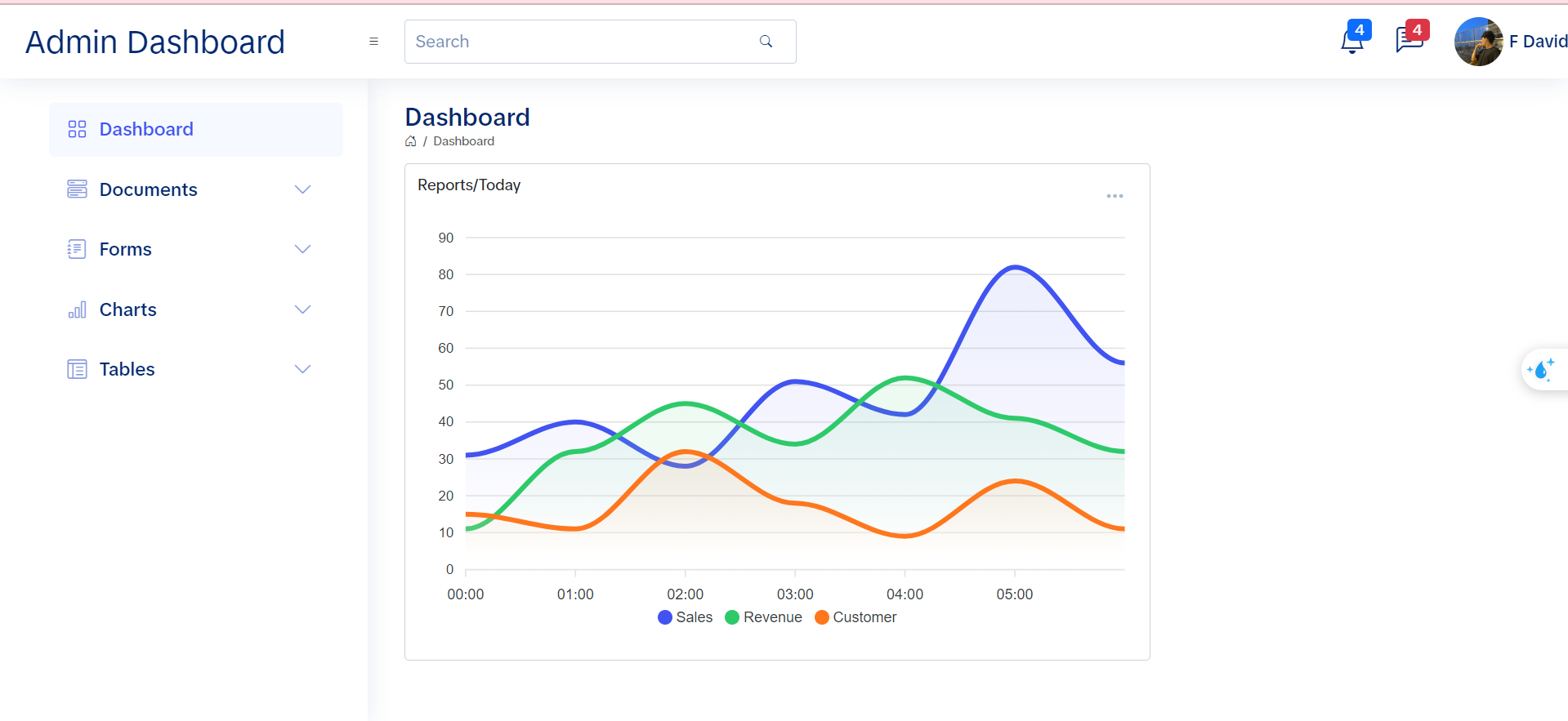
History transaction page



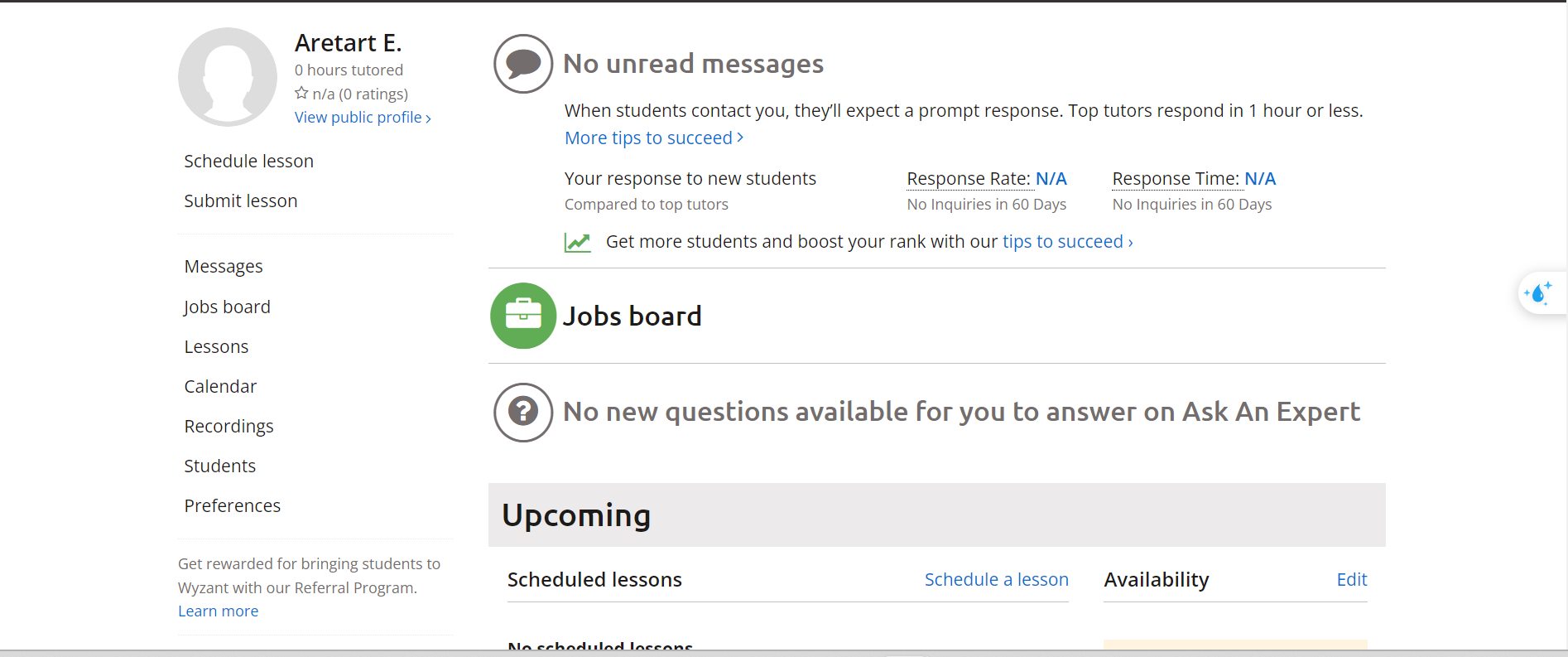
Payment page



Admin page



Dashboard tutor page



IV. TECHNICAL REQUIREMENT

1. Front-end with React:

Use React version 18 or newer to leverage the latest features and improvements.

Apply best practices for component structure, state management (e.g., Redux, MobX) and routing (e.g., React Router).

Integrate popular UI libraries like Material UI, Ant Design, or Chakra UI to accelerate development.

Implement responsive design to ensure the interface works well across various devices.

Use TypeScript instead of JavaScript to have typed code and reduce errors.

Perform unit and integration testing to ensure code quality.

2. Back-end with C#:

Use .NET Core 6.0 or newer to take advantage of the latest features.

Design RESTful APIs following the OpenAPI (Swagger) standard for easy integration and extensibility.

Utilize Entity Framework Core as the ORM to manage the database.

Apply Clean Architecture or CQRS patterns to increase scalability, maintainability, and testability.

Implement unit and integration tests for APIs and business logic classes.

3. Database:

Use SQL Server Database to store the data.

Design the database schema following normalization best practices to ensure data integrity and performance.

Optimize SQL queries and design appropriate indexes.

Establish a backup and recovery strategy to handle potential failures.

4. Deployment and Operations:

Use Docker to package the front-end and back-end as separate containers.

Deploy the application to a cloud platform like Azure or AWS using CI/CD pipelines.

Set up auto-scaling and load balancing mechanisms to ensure scalability.

Implement monitoring and logging tools to track the application's operational status.