Geega Tech

Technical Task for Laravel - School GPT

Objective

Develop an AI-based platform tailored for **school students**. The system will:

1. Enable **registration** and **login** for users (students).
2. Provide a **chat interface** where students can ask questions, and GPT answers tailored to their class level.

Functional Requirements

1. User Registration - Sign Up API

Field Name	Data Type	Constraints
Name	String	Required
Class	Integer	Required, Range (1-12)
Phone Number	String	Required, Unique, Valid
Email ID	String	Required, Unique, Valid Email
Password	String	Required, Min Length 8

- Encrypt the password using Laravel's **bcrypt**.
- Use Laravel Sanctum for secure **token-based authentication**.

API Endpoint:

```
POST /api/signup
Request Body:
{
  "name": "John Doe",
  "class": 4,
  "number": "9876543210",
  "email": "john@example.com",
  "password": "securepassword"
}
Response:
{
  "message": "Registration Successful",
  "data": { "user_id": 123, "name": "John Doe" }
}
```

---

2. User Login API

Field Name	Data Type	Constraints
Email ID	String	Required, Valid Email
Password	String	Required

Functionality:

- Authenticate users using email and password.
- Issue a **Sanctum Token** for subsequent requests.

API Endpoint:

```
POST /api/login
Request Body:
{
```

```
  "email": "john@example.com",
  "password": "securepassword"
}
Response:
{
  "message": "Login Successful",
  "token": "generated_token_string"
}
```

---

3. Chat Interface for School GPT

The chat interface allows students to ask questions and receive tailored responses based on their class level.

- **Front-End**:
  - A simple chat UI using **Blade Templates** or Vue.js.
  - Display the conversation in a user-friendly format.
- **Backend Logic**:
  - Fetch the student's class using the authenticated user token.
  - Create a **custom GPT prompt** based on the class level (1-12).
  - Call the **OpenAI API** to generate responses.
  - Log all questions and answers in the database for history.

API Endpoint:

```
POST /api/chat
Headers:
Authorization: Bearer {token}

Request Body:
{
  "question": "What is gravity?"
}

Response:
{
```

```
  "answer": "Gravity is a force that pulls everything toward the ground. For example, when you drop a ball, it falls down because of gravity."
}
```

School GPT Prompt Example:

```
function generatePrompt($class, $question) {
    $levels = [
        1 => "Explain simply for a 1st grader: ",
        4 => "Answer clearly for a 4th-grade student: ",
        7 => "Provide a moderately detailed answer for a 7th grader: ",
        10 => "Provide a detailed and technical answer for a 10th-grade student: "
    ];

    return $levels[$class] . $question;
}
```

Database Structure

1. Users Table

Column	Type	Constraints
`id`	Integer	Primary Key, Auto-Increment
`name`	String	Not Null
`class`	Integer	Range (1-12)
`email`	String	Unique, Not Null
`password`	String	Encrypted
`created_at`	Timestamp	

2. Chat History Table

Column	Type	Constraints
`id`	Integer	Primary Key, Auto-Increment
`user_id`	Integer	Foreign Key (users)
`question`	Text	Not Null
`answer`	Text	Not Null
`created_at`	Timestamp	

Non-Functional Requirements

1. **Performance**: API response time under 3 seconds.
2. **Security**: Secure all endpoints with Laravel Sanctum.
3. **Scalability**: Design APIs to support multiple concurrent users.

Deliverables

1. **Sign-Up API** for user registration
2. **Login API** for user authentication
3. **Chat API** integrated with OpenAI to provide tailored responses
4. Simple **Chat Interface** for student interaction
5. Postman Collection for API Testing
6. Full Laravel Codebase with Documentation

Deadline: Friday, 20 Dec, 2024

Assign To: Kamani Anand

Geega Tech