**Project Proposal for DNACollaborator**

**Title**: DNACollaborator

**Introduction:**

* **Overview**: DNACollaborator is a web-based platform for genealogy enthusiasts. It enables users to collaborate in identifying DNA segments from ancestors, reconstruct ancestral DNA, and extend family trees.
* **Objective**: To provide a tool for genealogy enthusiasts to analyze DNA data, collaborate on family tree reconstruction, and integrate genetic information with traditional genealogical research.
* **Technologies to Be Used:** MERN stack, GraphQL, JWT, D3, WikiTree API, React-table.

**Project Description:**

1. **Idea Conceptualization:**

* Address the challenge of tracing DNA segments to specific ancestors.
* Facilitate collaboration among genealogy researchers.

2. **User Interface (React):**

* Interface for uploading DNA test files.
* Dashboard with user and tester information, family tree visualization, and chromosome graphics.
* React table for listing DNA segments with search, sort, and filter capabilities.

**3. Backend Services (Express.js, Node.js):**

* Backend to handle file uploads, DNA data parsing, and user management.
* Integration with WikiTree API for family tree data.

**4. Database (MongoDB):**

* Schema to store user profiles, DNA segment data, and family tree information.
* Ensure scalability and efficient data retrieval.

**5. API Integration (GraphQL):**

* Use GraphQL for flexible and efficient data queries.
* Manage data exchange between frontend, backend, and WikiTree API.

**6. Authentication (JWT):**

* Secure user authentication using JWT.
* Implement access control for paid and unpaid users.

**7. Data Visualization (D3):**

* Use D3 for chromosome visualization.
* Implement interactive elements for user engagement.

**Project Roadmap:**

**Week 1:**

* Day 1-2: Project setup, basic frontend and backend structure, and set up JWT authentication.
* Day 3-4: Frontend development for file uploading and user dashboard.
* Day 5-6: Backend development for file processing and database schema.
* Day 7: Initial integration of frontend and backend.

**Week 2:**

* Day 8-9: Implement D3 visualizations for family tree and chromosomes.
* Day 10-11: Integrate WikiTree API.
* Day 12: Testing and debugging of all components.
* Day 13: Finalize user access control and data sharing features.
* Day 14: Final review, adjustments, and deployment.

**Initial Task List for Project Board**

1. **Project Setup:**

* Initialize project repository and MERN stack environment.
* Set up basic frontend and backend skeleton.

2. **File Upload Interface:**

* Develop UI for DNA file upload.
* Implement file parsing logic on the backend.

3. **User Dashboard Development:**

* Create user information display and tester selection features.
* Integrate React Table for DNA segment display.

4. **D3 Visualizations:**

* Implement family tree visualization using D3.
* Develop chromosome graphic representation with D3.

5. **WikiTree API Integration:**

* Integrate and test WikiTree API for family tree data.

6. **Authentication System:**

* Set up JWT authentication.
* Implement access controls for different user types.

7. **Testing and Deployment:**

* Conduct comprehensive testing.
* Deploy the application to a hosting service.

A screenshot of a computer

Description automatically generated

Note that you can get all fields. When you pull this in javascript, you see the fields necessary for:

A screenshot of a computer code

Description automatically generated

curl 'https://api.wikitree.com/api.php?action=getProfile&key=Clemens-1&fields=Id,PageId,Name,FirstName,LastNameAtBirth,BirthDate,DeathDate'

or

curl 'https://api.wikitree.com/api.php?action=getProfile&key=7146&fields=Id,PageId,Name,FirstName,LastNameAtBirth,BirthDate,DeathDate'

[

{

"page\_name": "Clemens-1",

"profile": {

"Id": 5185,

"PageId": 7146,

"Name": "Clemens-1",

"FirstName": "Samuel",

"LastNameAtBirth": "Clemens",

"BirthDate": "1835-11-30",

"DeathDate": "1910-04-21"

},

"status": 0

}

]