**1. API**

**Implementation**

For the API, I used Node.js with MongoDB.

**Choice Justification**

The scope of this assisment is relatively small, meaning there are no significant technical factors favoring a relational or non-relational database. Both options could satisfy the requirements effectively.

MongoDB was chosen primarily for the following reasons:

1. **Ease of Setup:** MongoDB is quick to set up, whether locally or using a cloud service like MongoDB Atlas.
2. **Fast Deployment:** It supports rapid development and deployment.
3. **Collaboration:** MongoDB Atlas facilitates real-time collaboration by allowing multiple users to work on the same database.

**Suggestions for Improvement**

To enhance the API's functionality and security:

1. **CORS Policy & CSRF Protocole:**
   * Implement a CORS policy to allow only authorized clients to communicate with the API.
   * Enforce CSRF protection to block unauthorized requests from tools like Postman or bots, safeguarding the API against denial-of-service (DoS) attacks.
2. **Logging Middleware:**
   * Add a middleware to log all incoming and outgoing requests.
   * Store these logs in a dedicated document to track errors and monitor API usage effectively.
3. **Hashing Protocol:**
   * Use hashing for all incoming and outgoing requests to ensure secure data transmission between the client and the API.

**Steps to Test Email Functionality:**

1. **Create a Mailgun Account:**
   * Sign up for a Mailgun account to enable email sending functionality.
2. **Set Up Environment Variables:**
   * Add the sandbox test email provided by Mailgun to your .env file.
   * Generate an API key from Mailgun and include it in the .env file.
3. **Authorize Email Recipients:**
   * Authorize the sandbox test email addresses to allow email sending within the test environment.
4. **Important Security Reminder:**
   * **Do not push your .env file** containing the Mailgun API key to a public GitHub repository. Doing so may result in your account being blocked by Mailgun for unauthorized access.

**Testing Credentials:**

You can use the following test user credentials to interact with all endpoints:

* **Email:** testfares@yopmail.com
* **Password:** 123456

**Documentation link:**

<http://localhost>:4200/api-docs

### ****2. Frontend****

#### ****Implementation****

For the frontend, I used React, as specified in the assignment.

#### ****Clarification****

Although I followed the requirement to use React, my expertise lies more in Angular and Ionic frameworks. To complete the frontend, I leveraged assistance from ChatGPT.

#### ****Additional Details****

I can provide screenshots of my conversations with ChatGPT during the development process if required for evaluation purposes.