Endorse Collect

Project Report

Version 1.0

Industrial Training (ECS591)

Degree

BACHELOR OF TECHNOLOGY (CSE)

PROJECT GUIDE:

Mr. Divyanshu Saxena

SUBMITTED BY:

Jain Aman Ajay
TCA2209075

AUGUST – DECEMBER ,2024

FACULTY OF ENGINEERING & COMPUTING SCIENCES

TEERTHANKER MAHAVEER UNIVERSITY, MORADABAD

DECLARATION

I hereby declare that this Project Report titled **Endorse Collect** submitted by me and approved by our project guide, Faculty of Engineering & Computing Sciences. Teerthanker Mahaveer University, Moradabad, is a Bonafide work undertaken by us and it is not submitted to any other University or Institution for the award of any degree diploma / certificate or published any time before.

Project ID : Project Id

Student Name: Jain Aman Ajay Signature

Project Guide : Mr. Divyanshu Saxena Signature

Table of Contents

1 PROJECT TITLE 4

2 PROBLEM STATEMENT 4

3 PROJECT DESCRIPTION 4

3.1 SCOPE OF THE WORK 5
3.2 PROJECT MODULES 6
3.3 CONTEXT DIAGRAM (HIGH LEVEL) 10

4 IMPLEMENTATION METHODOLOGY 11

5 TECHNOLOGIES TO BE USED 14

5.1 SOFTWARE PLATFORM 14
5.2 HARDWARE PLATFORM 14
5.3 TOOLS, IF ANY 14

6 ADVANTAGES OF THIS PROJECT 14

7 ASSUMPTIONS, IF ANY 17

8 FUTURE SCOPE AND FURTHER ENHANCEMENT OF THE PROJECT 17

9 PROJECT REPOSITORY LOCATION 18

10 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS 18

11 CONCLUSION 19

12 REFERENCES 20

Appendix

A: Data Flow Diagram (DFD)

B: Entity Relationship Diagram (ERD)

C: Use Case Diagram (UCD)

D: Data Dictionary (DD)

E: Screen Shots

1 Project Title

Endorse Collect

2 Problem Statement

Collecting and managing client testimonials can be a tough process, especially for businesses that rely on positive feedback to build trust and credibility. Traditional methods of gathering testimonials are often time-consuming and disorganized, leading to missed opportunities for leveraging client feedback effectively. The need for a streamlined, scalable, and user-friendly solution led to the development of Endorse Collect. This platform aims to solve the problem by providing an efficient way to collect, manage, and display testimonials, thus helping businesses enhance their online reputation.

Client testimonials are essential for businesses to establish trust and credibility among their audience. However, traditional methods of collecting and managing testimonials are inefficient, prone to errors, and lack scalability. With the growing emphasis on digital presence, businesses require a system that integrates functionality, usability, and reliability to meet their testimonial management needs.

3 Project Description

Endorse Collect is a web-based platform developed to simplify the collection, organization, and display of client testimonials. With its responsive design and intuitive user interface, the platform offers businesses a reliable solution for managing client feedback. The application focuses on delivering a seamless experience for users while maintaining high scalability and security standards.

Client testimonials are essential for businesses to establish trust and credibility among their audience. However, traditional methods of collecting and managing testimonials are inefficient, prone to errors, and lack scalability. With the growing emphasis on digital presence, businesses require a system that integrates functionality, usability, and reliability to meet their testimonial management needs.

Endorse Collect bridges this gap by providing:

1. Easy collection of client feedback.
2. Efficient management of testimonials using CRUD operations.
3. Enhanced credibility through streamlined display on digital platforms.

3.1 Scope of the Work

What Will Be Done:

User Interface Development:
A responsive and user-friendly interface will be designed and implemented using **Next.js**. The interface will prioritize ease of use, ensuring users can create, edit, view, and manage testimonials effortlessly. Key features will include accessible navigation, interactive feedback forms, and a clean layout optimized for both desktop and mobile devices.

Backend Development:
A robust and secure backend will be developed using **Node.js** and **Express.js**, ensuring smooth handling of all API requests and data transactions. The backend will be designed to support scalability, accommodate multiple users simultaneously, and implement best practices for security, such as input validation and data encryption. **MongoDB** will be integrated for efficient data management and retrieval.

Database Management:
The database schema will be carefully designed using **MongoDB**, catering to CRUD operations (Create, Read, Update, Delete). Data integrity and optimization will be emphasized to handle varying volumes of testimonial data. Relationships and indexing will be implemented to support faster query performance and seamless user experiences.

Deployment:
The entire application will be deployed on **Vercel**, leveraging its capabilities for continuous integration and deployment (CI/CD). Vercel's global edge network will ensure fast page load times, high availability, and robust scalability to support a growing user base.

Testing:
The platform will undergo comprehensive testing at every stage of development. This includes:

Unit Testing: Verifying the functionality of individual components and backend endpoints.

Integration Testing: Ensuring smooth communication between the frontend, backend, and database layers.

End-to-End Testing: Simulating real-world user scenarios to confirm the application performs as expected and is free of critical issues before launch.

What Will Not Be Done:

Advanced Analytics:
While the application will support basic display and management of testimonials, it will not include advanced analytics or reporting features, such as tracking user engagement, sentiment analysis, or detailed metrics**.**

Third-Party Integrations:
The scope excludes any integrations with external platforms or services. Features like social media sharing, CRM synchronization, or email marketing tools will not be implemented in this project.

Multi-Language Support:
The application will only support English, with no additional features for multi-language functionality or translation options. Expanding to support other languages may be considered for future phases.

Custom Theming:
The design will focus on a clean and simple aesthetic without extensive customization options. Users will not have access to advanced theming or branding tools, such as color palette adjustments or custom logos, in this version.

3.2 Project Modules

1. User Authentication Module:

- **Description:**
  This module is responsible for securely managing user registration, login, and authentication processes. It ensures that only authorized users can access the platform, safeguarding sensitive data and protecting against unauthorized access. Key functionalities include:
    - **User Registration:** A simple and secure process for new users to create accounts with validated email addresses and strong password enforcement.
    - **Login System:** Users can log in using their credentials, with features like error handling for incorrect inputs and account lockout after multiple failed attempts.
    - **Authentication and Authorization:** Secure session management to verify user identity on every request. Role-based access control (RBAC) will ensure users can only access functionalities based on their assigned roles (e.g., admin vs. regular user).

- **Secure Password Storage:** Passwords will be hashed using strong algorithms like bcrypt to prevent direct access to sensitive information.

- **Technology:**
  - **Next.js** for the frontend interface of authentication forms.
  - **Node.js** for backend services to handle API requests and manage authentication logic.
  - **JWT (JSON Web Tokens):** Used for stateless and secure token-based authentication, ensuring scalability and improved user experience.
  - **Session Storage:** Secure cookies or local storage will manage token-based sessions with expiration mechanisms.

2. Testimonial Management Module:

- **Description:**
  This core module allows users to create, view, edit, and delete testimonials. It's designed with ease of use, robust validation, and security in mind. Main features include:
  - **Create Testimonials:** Users can submit testimonials through a guided form, which includes input validation and sanitization to ensure data quality and prevent injection attacks.
  - **Edit and Delete Testimonials:** Authenticated users can update or remove their testimonials, with proper feedback provided to confirm their actions.
  - **Categorization and Filtering:** Testimonials can be organized and filtered by criteria like date, rating, or relevance, allowing users to quickly find specific feedback.
  - **Pagination and Search:** Optimized for a smooth user experience even with large datasets.
  - **Responsive Design:** The module is designed to work seamlessly on both desktop and mobile devices.

- **Technology:**
  - **Frontend: Next.js** to provide a dynamic, responsive, and intuitive user interface.
  - **Backend: Node.js/Express.js** to handle API logic.

- **Database: MongoDB** to store, retrieve, and manage testimonial data with schema validation.

3. Admin Dashboard Module:

- **Description:**
  This module offers an advanced interface for administrators to manage the platform effectively. Its purpose is to ensure smooth operation and provide oversight of user activities. Features include:
  - **User Management:** Admins can view and manage user profiles, deactivate accounts, and monitor activity logs.
  - **Testimonial Oversight:** Administrators can approve, edit, or delete testimonials as needed.
  - **Platform Analytics:** Provides insights into usage patterns, testimonial performance, and user engagement through visual reports like graphs and charts.
  - **Data Backup and Maintenance:** Tools for regular database backups, ensuring data integrity and disaster recovery capabilities.
  - **Role-Based Controls:** Different access levels for administrators and moderators, enhancing platform security.
- **Technology:**
  - **Frontend: Next.js** for a dynamic and visually appealing admin interface.
  - **Backend: Node.js/Express.js** to implement admin APIs and handle secure operations.
  - **Database: MongoDB** to support data storage and efficient querying for large datasets.
  - **Visualization:** Libraries like **Chart.js** or **D3.js** to generate analytics and reports.

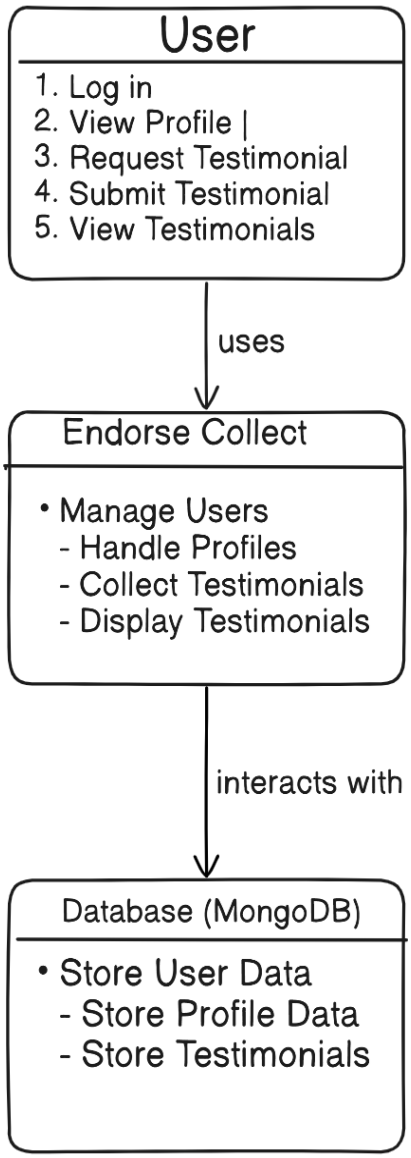
4. Testimonial Collecting Module:

- **Description:**
  This module allows customers to easily submit feedback or reviews for users. It provides a structured form where they can fill in specific details, ensuring consistent and useful data collection. The workflow includes:

- **Feedback Form:** Customers can input their details, including:
  - Name, position, and company (optional).
  - Written review or feedback with character count validation.
  - Rating system (e.g., 1–5 stars).
  - Links to their online presence (e.g., LinkedIn, portfolio).
- **Form Submission:** Once submitted, the testimonial is validated and securely stored in the database. Optional confirmation messages and previews allow users to review their input before final submission.
- **Data Management:** Submitted testimonials are automatically categorized for easy management and display in the Testimonial Management Module.

- **Technology:**
  - **Frontend: Next.js** to create an interactive and accessible form interface, designed for responsive usage across devices.
  - **Backend: Node.js/Express.js** for handling submission logic, data validation, and storage.
  - **Database: MongoDB** to store collected testimonials with schema definitions for proper structuring.

3.3 Context Diagram (High Level)



4 Implementation Methodology

1. Requirements Gathering:

This phase involved understanding and documenting the key functionalities, user expectations, and technical specifications essential for building the testimonial management platform. The main tasks included:

- Identifying user roles (e.g., regular users and administrators) and their specific needs.
- Defining the platform's core features, such as testimonial creation, editing, filtering, and administrative oversight.
- Outlining non-functional requirements, such as security, scalability, performance, and responsive design.
- Preparing a detailed Software Requirements Specification (SRS) document to guide development.

2. Technology Selection:

Frontend:

Next.js: Leveraged for its server-side rendering (SSR) capabilities to improve performance and SEO while providing a responsive and interactive user interface.

Integrated dynamic routing for seamless navigation between pages.

Backend:

Node.js and Express.js: Selected for fast and efficient server-side API development, offering excellent scalability and compatibility with JavaScript across the stack.

Database:

MongoDB: Used for its flexibility in handling document-based data structures, enabling efficient storage and retrieval of testimonials. Schema design supported relationships between testimonials, users, and categories.

- **Styling Framework:**
  - **Tailwind CSS:** Employed to create a modern, responsive, and visually appealing user interface with minimal development time. Its utility-first approach allowed for highly customizable and maintainable styles.
- **Authentication Technology:**

- **JWT (JSON Web Tokens):** Chosen for secure, stateless authentication, ensuring a scalable and seamless user login experience.

3. Module Development:

The platform was divided into distinct modules, each focusing on specific functionalities to streamline development and testing.

- **User Authentication Module:**
  - Designed and implemented a secure system for user registration and login.
  - Passwords were hashed using bcrypt to ensure secure storage.
  - JWT was integrated for stateless authentication, managing user sessions with role-based access control (RBAC).
- **Testimonial Management Module:**
  - Developed CRUD (Create, Read, Update, Delete) functionalities for testimonials.
  - Incorporated real-time validation for form inputs to prevent invalid or harmful data.
  - Designed filtering options to allow users to sort testimonials by date, rating, or relevance.
  - Optimized pagination to enhance user experience while handling large datasets.
- **Admin Dashboard Module:**
  - Built tools for administrators to monitor and manage testimonials, user accounts, and platform activity.
  - Integrated analytics to display insights such as the number of active users, testimonial trends, and overall platform engagement.
  - Provided database backup functionality and options for user account management.
- **Feedback Collection Module:**
  - Developed an interactive form for customers to submit testimonials.
  - Included fields for optional information, such as position, company, rating, and online presence.
  - Ensured smooth submission flow with user feedback upon successful form completion.

4. Deployment:

The entire application was deployed to a production-ready environment, ensuring high availability, scalability, and ease of access:

- **Vercel:** Chosen as the hosting platform due to its seamless integration with Next.js, automatic scaling, and global CDN (Content Delivery Network) for fast load times.
- Integrated CI/CD (Continuous Integration/Continuous Deployment) pipelines to enable automated testing and deployment for future updates.
- Set up environment variables to securely manage API keys, database credentials, and other sensitive information.

5.Testing:

Rigorous testing was conducted throughout the development lifecycle to ensure the platform's functionality, performance, and security:

- **Unit Testing:**
  - Focused on testing individual components and backend APIs to verify isolated functionality.
  - Tools like Jest were used to automate the testing process.
- **Integration Testing:**
  - Verified the seamless interaction between the frontend, backend, and database layers.
  - Simulated real-world scenarios to check data flow and functionality under various conditions.
- **End-to-End (E2E) Testing:**
  - Conducted comprehensive testing of user workflows, such as registration, testimonial submission, and dashboard usage.
  - Tools like Cypress were employed for automated browser testing.
- **Performance Testing:**
  - Measured response times, page load speeds, and database query performance under varying loads.
  - Ensured the platform could handle concurrent users without degradation in performance.

5 Technologies to be used

6 Advantages of this Project

Streamlined Testimonial Management

Centralized Platform:

The platform consolidates all client testimonials in one place, reducing the need for scattered storage methods like spreadsheets, emails, or standalone databases.

This centralization ensures easy access and management of all testimonials, enabling users to categorize, search, and update entries efficiently.

Seamless Workflow:

The platform simplifies the process of collecting, organizing, and updating testimonials through an intuitive interface and automated backend processes.

Features like CRUD (Create, Read, Update, Delete) operations and filtering allow users to manage testimonials without requiring technical expertise.

Enhanced Control:

Administrators can approve or reject testimonials before publishing, ensuring only high-quality and relevant feedback is displayed.

Options for exporting testimonials into different formats (e.g., JSON, CSV) for integration with other tools or platforms.

User-Friendly Interface

Intuitive Navigation:

Designed with usability in mind, the interface ensures even non-technical users can navigate the platform effortlessly.

Clear and logical menu structures guide users to key functions like adding new testimonials or viewing analytics.

Interactive Features:

Live previews allow users to see how testimonials will appear on their website before publishing.

Drag-and-drop functionality for rearranging testimonial order or grouping related entries.

Customizable Views:

Users can toggle between grid and list views for better visualization of testimonials, adapting to their preferred way of managing content.

Improved Credibility

Authenticity Displayed:

Showcasing real client feedback boosts a business's reputation by adding a layer of trust and transparency.

Optional fields for client photos, company names, and ratings add authenticity to testimonials, making them more relatable and trustworthy.

Impactful Presentation:

Testimonials can be presented dynamically using sliders or grids, enhancing the visual appeal of websites and marketing materials.

Highlighting top-rated or featured testimonials helps businesses showcase their strongest endorsements.

Social Proof:

By integrating testimonials into their marketing strategy, businesses can influence potential customers, turning positive feedback into a powerful conversion tool.

Responsive Design

Cross-Device Compatibility:

The platform is optimized for mobile, tablet, and desktop devices, ensuring a seamless experience regardless of the screen size.

Features like responsive grids, scalable typography, and touch-friendly controls enhance usability on smaller screens.

Adaptive Performance:

The design dynamically adjusts to varying internet speeds, providing a smooth experience even on slower connections.

Ensures accessibility for a broader audience, including users on low-end devices or those in remote areas.

Consistency Across Platforms:

Whether accessed via a browser or integrated into a native app, the platform maintains consistent functionality and appearance.

Scalable and Secure

Scalability:

The platform is built with Next.js and MongoDB, both known for their ability to handle high traffic and large datasets.

Horizontal scaling capabilities ensure that the platform can accommodate increasing user demands without performance degradation.

Data Security:

Secure authentication mechanisms, such as JWT (JSON Web Tokens), protect user accounts and restrict unauthorized access.

End-to-end encryption ensures sensitive data, like client details, remains private and protected during transmission.

Regular Updates:

The backend architecture supports smooth updates and new feature rollouts without disrupting ongoing operations.

Implements database indexing to maintain fast query performance even as data grows.

Time-Efficient

Automated Processes:

The testimonial collection process is automated, reducing manual effort for businesses and allowing them to focus on core operations.

Automated email reminders or notifications prompt clients to submit testimonials, further streamlining the collection process.

Bulk Management:

Features like bulk import/export, batch editing, and multi-selection actions allow users to manage multiple testimonials simultaneously, saving time.

Intelligent suggestions and autocomplete functions simplify data entry for repeated fields like company names or designations.

Integration Capabilities:

The platform can be integrated with existing systems (e.g., CRMs, websites) to automatically collect and display testimonials, reducing duplicate effort.

Scheduled publishing ensures testimonials are posted at optimal times for maximum visibility.

7 Assumptions, if any

User Accessibility: It is assumed that users accessing the platform will have a stable internet connection and a modern web browser for compatibility with the platform's features.
User Data: Users will provide accurate and genuine information when submitting testimonials or registering on the platform.
Hardware Requirements: It is assumed that the system used for development and deployment meets the specified hardware requirements (e.g., 8 GB RAM, 256 GB SSD).
Platform Usage: It is assumed that the platform will initially cater to a moderate user base, and any significant scaling requirements will be handled in future iterations.
Data Security Compliance: It is assumed that the platform's data storage and processing comply with standard security regulations like GDPR or relevant regional policies.

8 Future Scope and further enhancement of the Project

The future scope of the Endorse Collect project includes integrating advanced analytics for detailed reporting on testimonial performance, adding multi-language support to cater to a global audience, and developing a mobile application for iOS and Android to enhance accessibility. Future enhancements also involve integrating with social media and CRM systems for automated testimonial management, strengthening security with features like multi-factor authentication, and offering customizable templates and themes for personalized user experience. Additionally, incorporating AI for automated moderation, filtering inappropriate content, and implementing a feedback and review system to gather user insights will further improve the platform's functionality and user engagement.

While the platform is fully functional and meets its stated objectives, there are numerous opportunities for further development:

- **AI-Based Testimonial Moderation:** Implementing artificial intelligence to automate the review of testimonials for relevance and appropriateness can save time and ensure quality.
- **Multi-Language Support:** Expanding the platform’s usability to a global audience by incorporating multiple languages will significantly enhance its reach and appeal.
  Mobile Application Development: Developing native mobile apps for iOS and Android will make the platform even more accessible to businesses and their clients

9 Project Repository Location

S#	Project Artifacts (softcopy)	Location (Mention Lab-ID, Server ID, Folder Name etc.)	Verified by Project Guide	Verified by Lab In-Charge
1.	Project Synopsis Report (Final Version)	LAB 3201, PC 23 , TCA2209075	Name and Signature	Name and Signature
2.	Project Progress updates	LAB 3201 , PC 23 , TCA2209075	Name and Signature	Name and Signature
3.	Project Requirement specifications	LAB 3201 , PC 23 , TCA2209075	Name and Signature	Name and Signature
4.	Project Report (Final Version)	LAB 3201 , PC 23 , TCA2209075	Name and Signature	Name and Signature
5.	Test Repository	LAB 3201 , PC 23 , TCA2209075	Name and Signature	Name and Signature
6.	Project Source Code (final version) with executable	LAB 3201 , PC 23 , TCA2209075	Name and Signature	Name and Signature
7.	Any other document	LAB 3201 , PC 23 , TCA2209075	Name and Signature	Name and Signature

10 Definitions, Acronyms, and Abbreviations

Abbreviation	Description
SRS	Software Requirements Specification
UI	User Interface
CRUD	Create, Read, Update,Delete
JWT	JSON Web Token
API	Application Programming Interface
DB	Database
SSD	Solid State D
RAM	Random Access Memory
CSS	Cascading Style Sheet

11 Conclusion

The **Endorse Collect** project stands as a testament to the effectiveness of modern web development technologies in solving real-world problems. It successfully fulfills its primary objective of providing a scalable, efficient, and user-friendly platform for managing client testimonials, addressing a critical need for businesses striving to enhance their digital presence and build credibility.

The platform offers a range of features that streamline testimonial collection, management, and display. Its key strengths include:

- **Responsive and User-Friendly Interface:** The platform's intuitive design, built using **Next.js** and styled with **Tailwind CSS**, ensures that users across all devices—mobile, tablet, and desktop—can easily navigate and interact with the application.
- **Secure and Scalable Backend Operations:** By leveraging **Node.js**, **Express.js**, and **MongoDB**, the platform ensures secure data handling, robust performance, and the ability to scale effortlessly as user demand grows.
- **Enhanced Business Credibility:** The ability to efficiently manage and showcase authentic client feedback directly contributes to improved trust and reputation for businesses utilizing the platform.

Achievements of the Project

Throughout its development, **Endorse Collect** achieved several milestones:

1. **Simplified Testimonial Management:** The platform makes it easy for users to create, edit, categorize, and display testimonials, saving time and effort.
2. **Comprehensive Features:** Modules like secure user authentication, CRUD operations, and an admin dashboard provide a holistic solution to testimonial management.
3. **High-Performance Architecture:** The combination of server-side rendering, efficient database queries, and lightweight front-end design ensures the application performs reliably under heavy traffic.
4. **Ease of Deployment:** Hosting on **Vercel** provides high availability, seamless updates, and a global reach for users.

12 References

- **Next.js Documentation**:
https://nextjs.org/docs

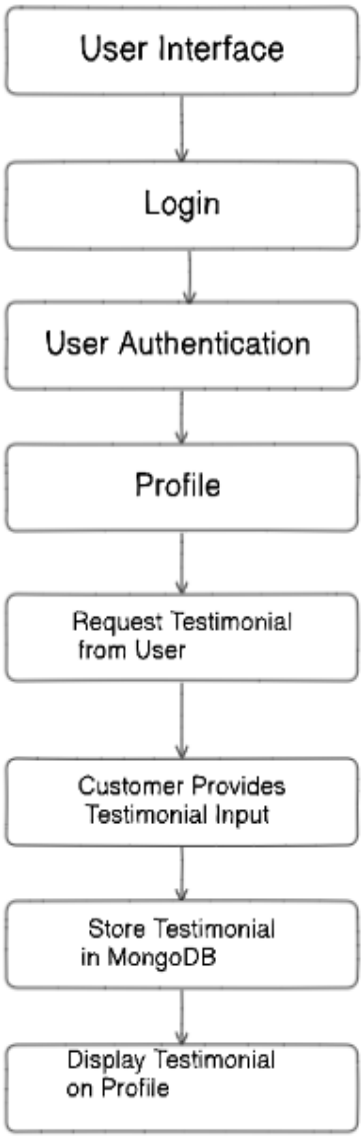
- **MongoDB Documentation**:
https://docs.mongodb.com/

- **Node.js Documentation**:
https://nodejs.org/en/docs/

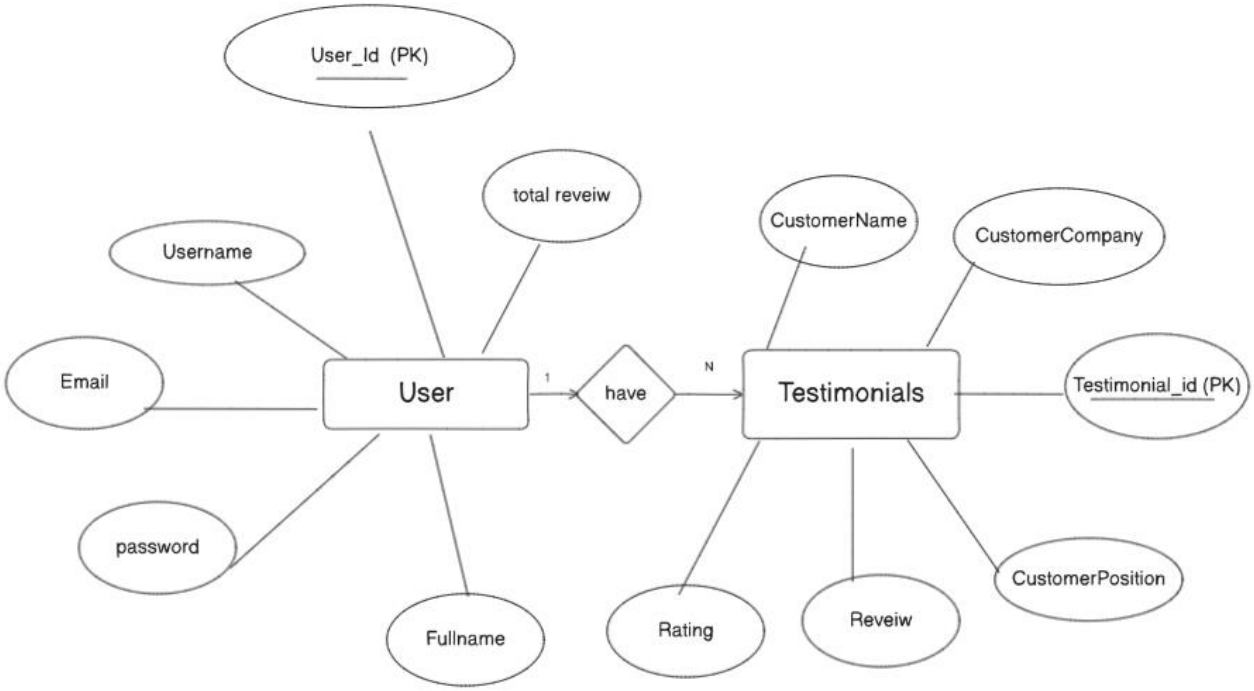
- **Vercel Deployment Guide**:
https://vercel.com/docs

S#	Reference Details	Owner	Version	Date
1.	**Next.js Documentation**:	Vercel	Latest	12/11/24
2.	**MongoDB Documentation**:	MongoDB	Latest	12/11/24
3.	**Node.js Documentation**:	Node.js	Latest	12/11/24
4.	**Vercel Deployment Guide**:	Vercel	Latest	26/11/24

Annexure A
Data Flow Diagram (DFD)

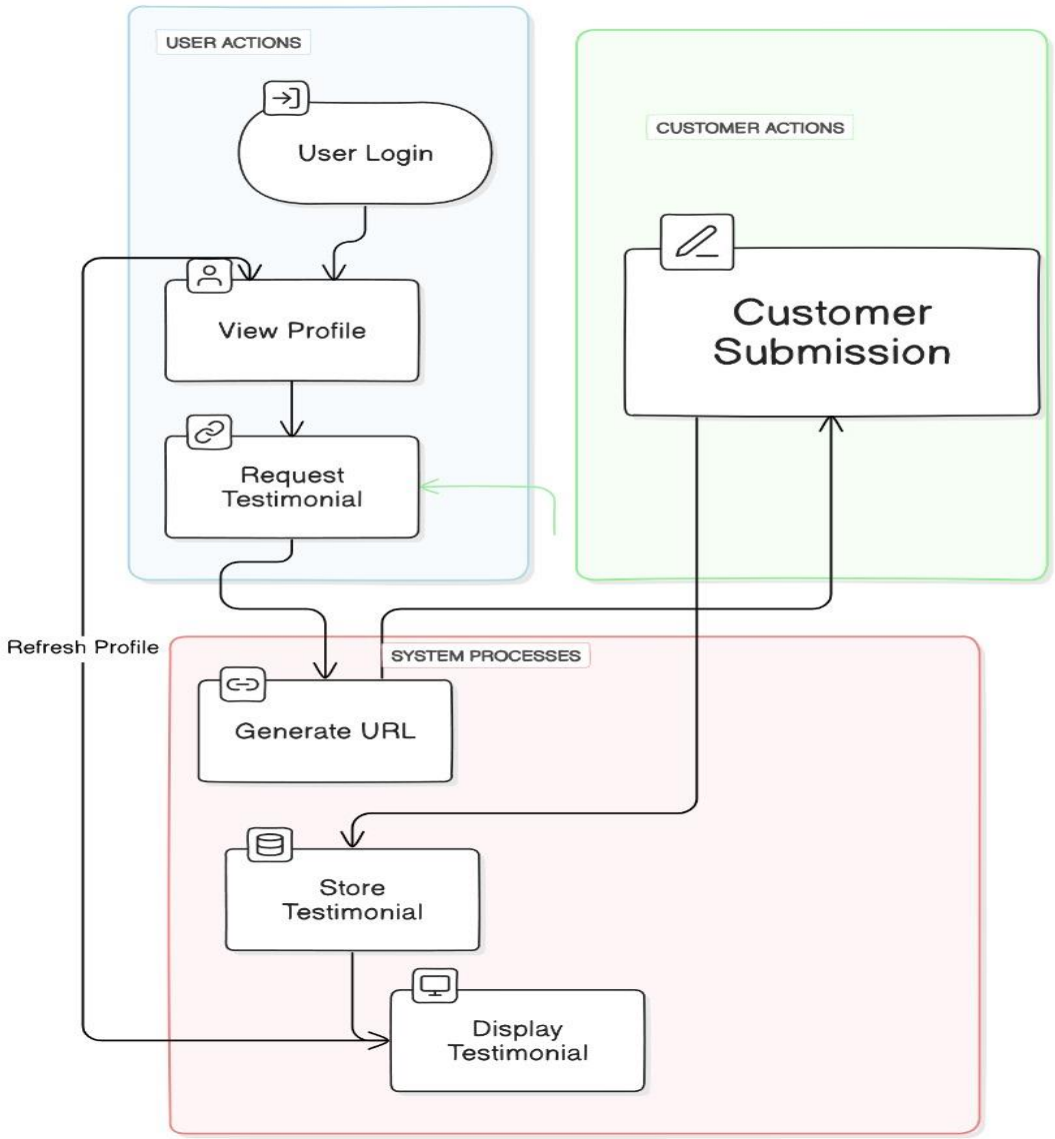


Annexure B
Entity-Relationship Diagram (ERD)



Annexure C
Use-Case Diagram (UCD)

Endorse Collect Testimonial Collection Flowchart



Annexure D
Data Dictionary (DD)
(Mandatory)

Example:

User Table (USR)

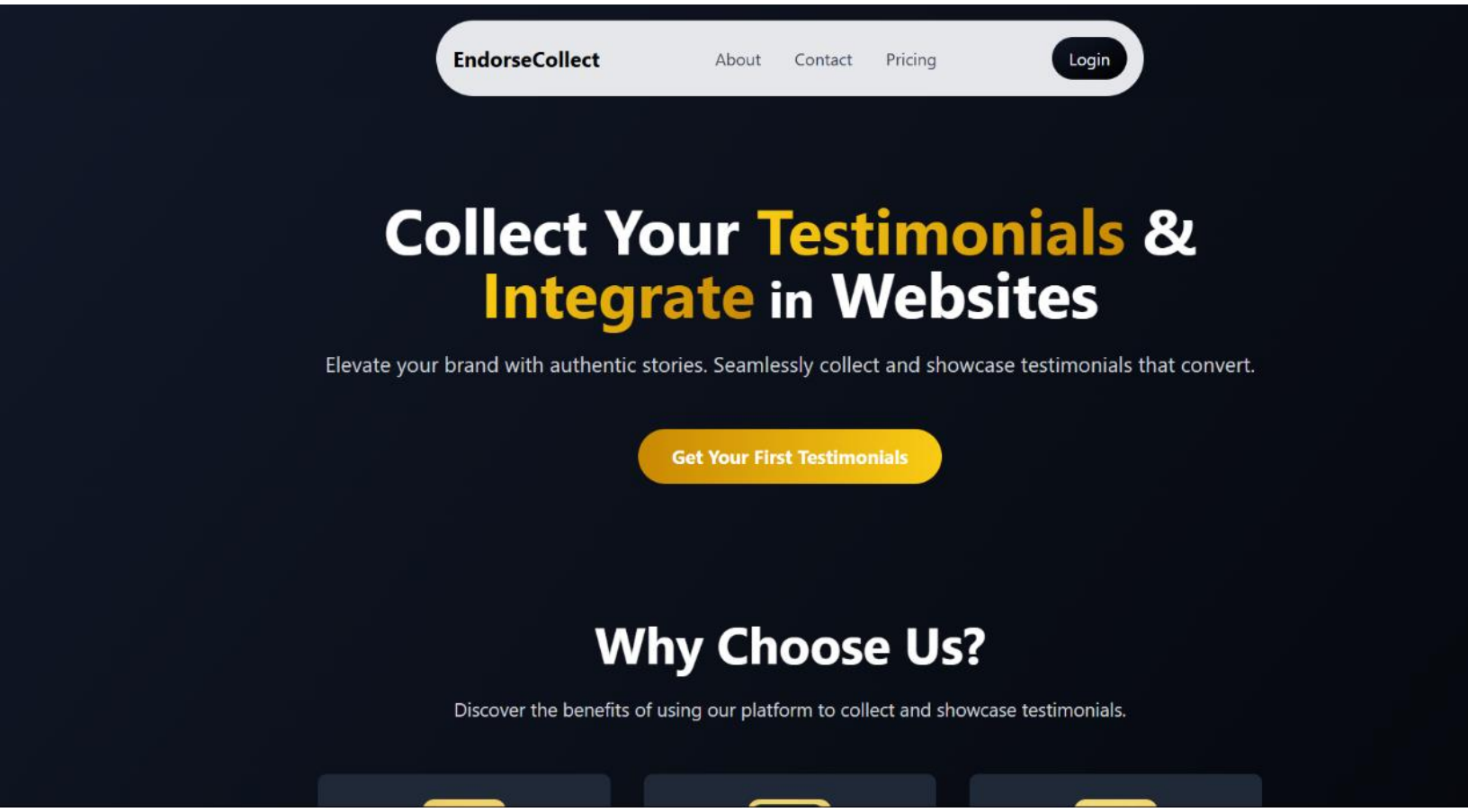
Fields	Data type	Description
USR-ID	Number	Unique identifier for the user
USR-Name	Text	User's full name
USR-Email	Text	User's email address
USR-Password	Text	Encrypted password

Testimonial Table(TEST)

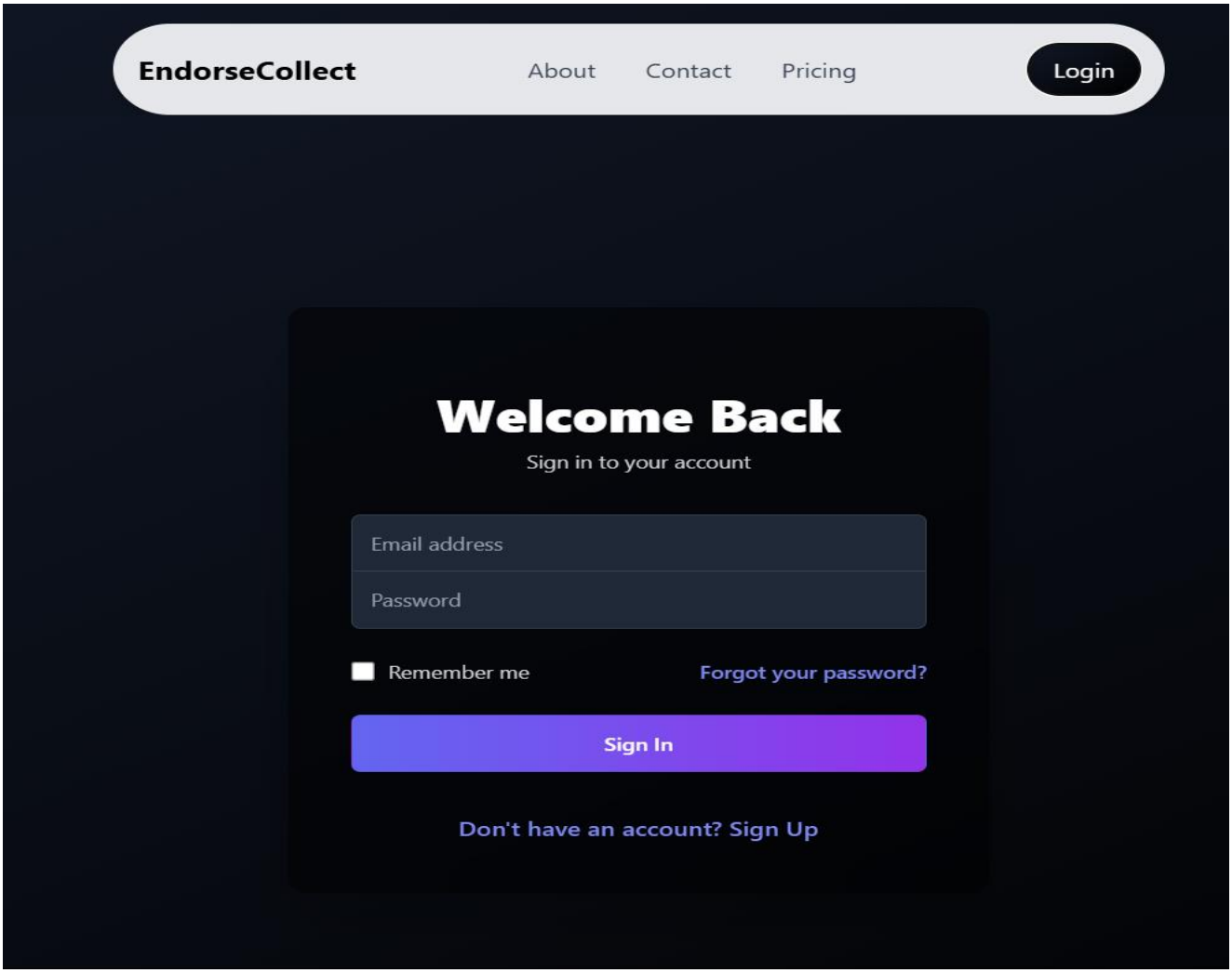
Fields	Data type	Description
TEST-ID	Number	Unique identifier for the testimonial
TEST-User-ID	Number	User ID linked to the testimonial
TEST-Content	Text	The actual testimonial content
TEST-Rating	Number	Rating Given Out of 5
TEST-Date	Date	Date when the testimonial was submitted

Annexure E
Screen Shots

Home Page:

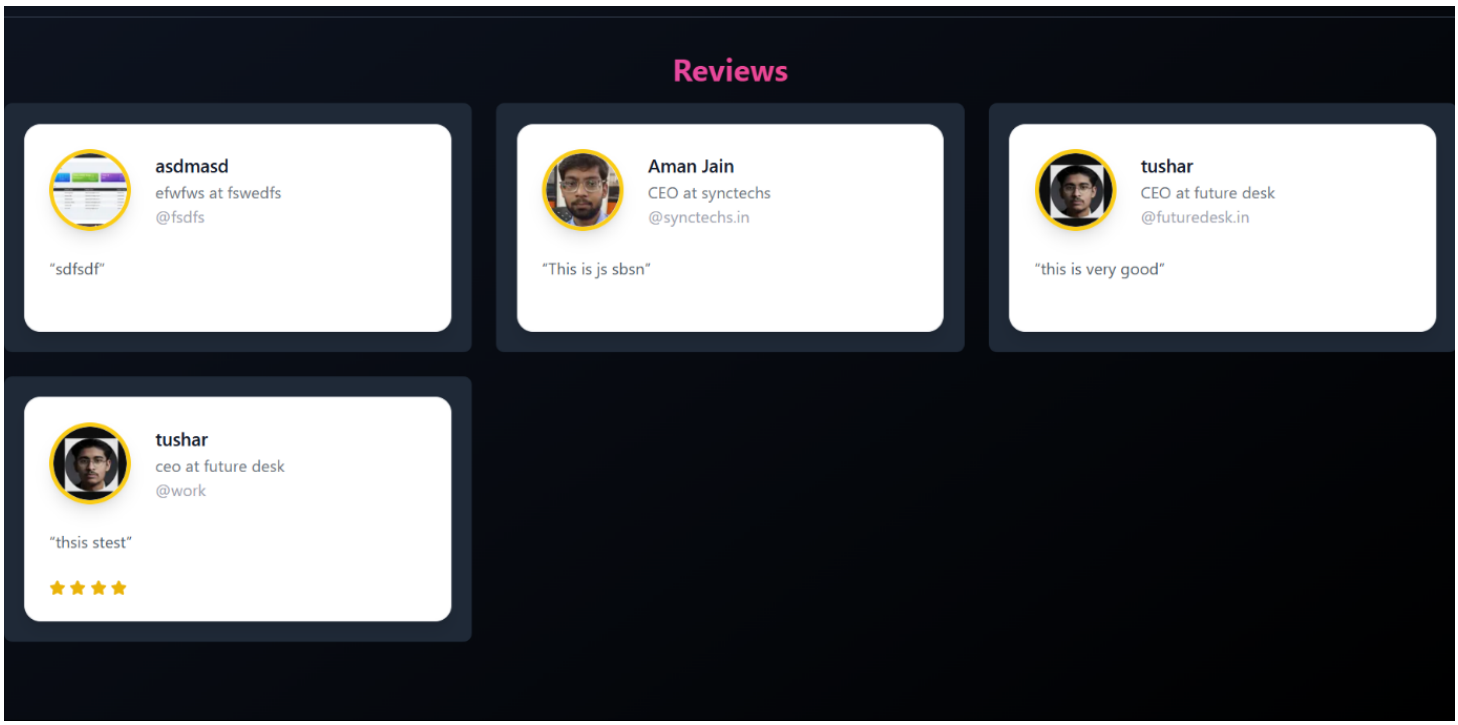


Login Page

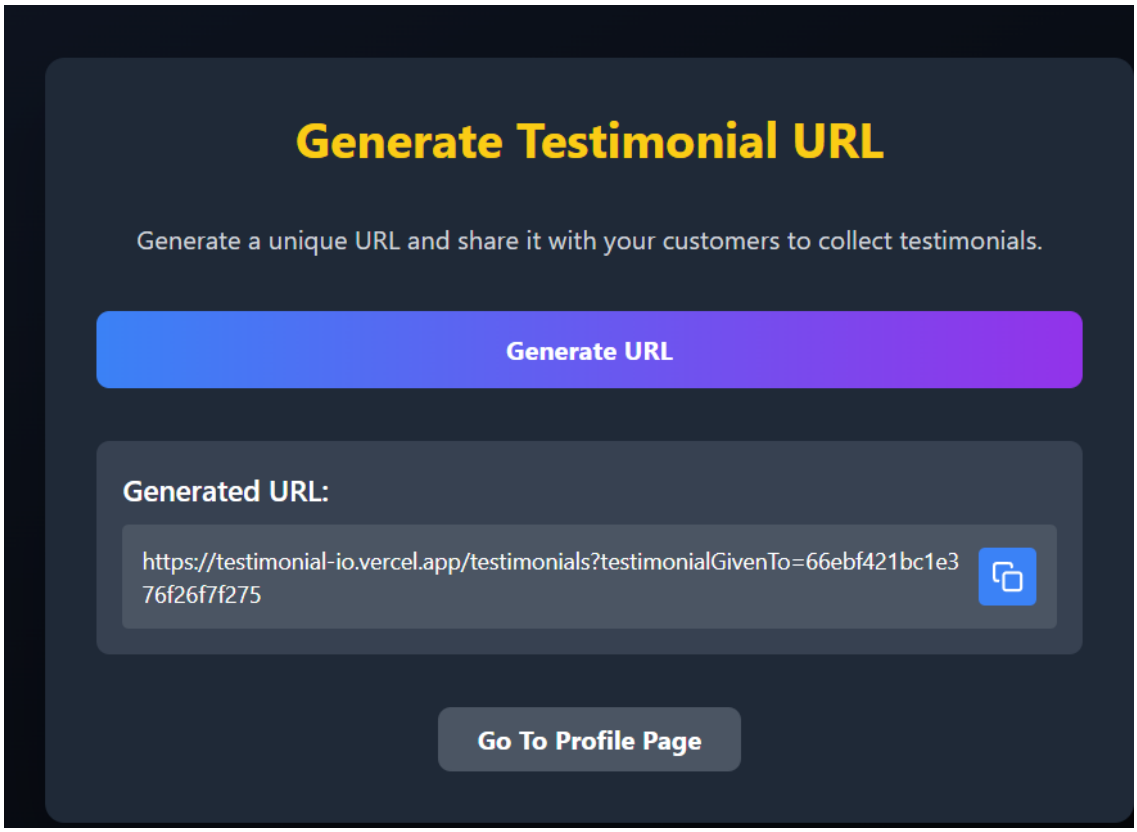

EndorseCollect
About
Contact
Pricing
Login
Welcome Back
Sign in to your account
Email address
Password
Remember me
Forgot your password?
Sign In
Don't have an account? Sign Up

Testimonial card

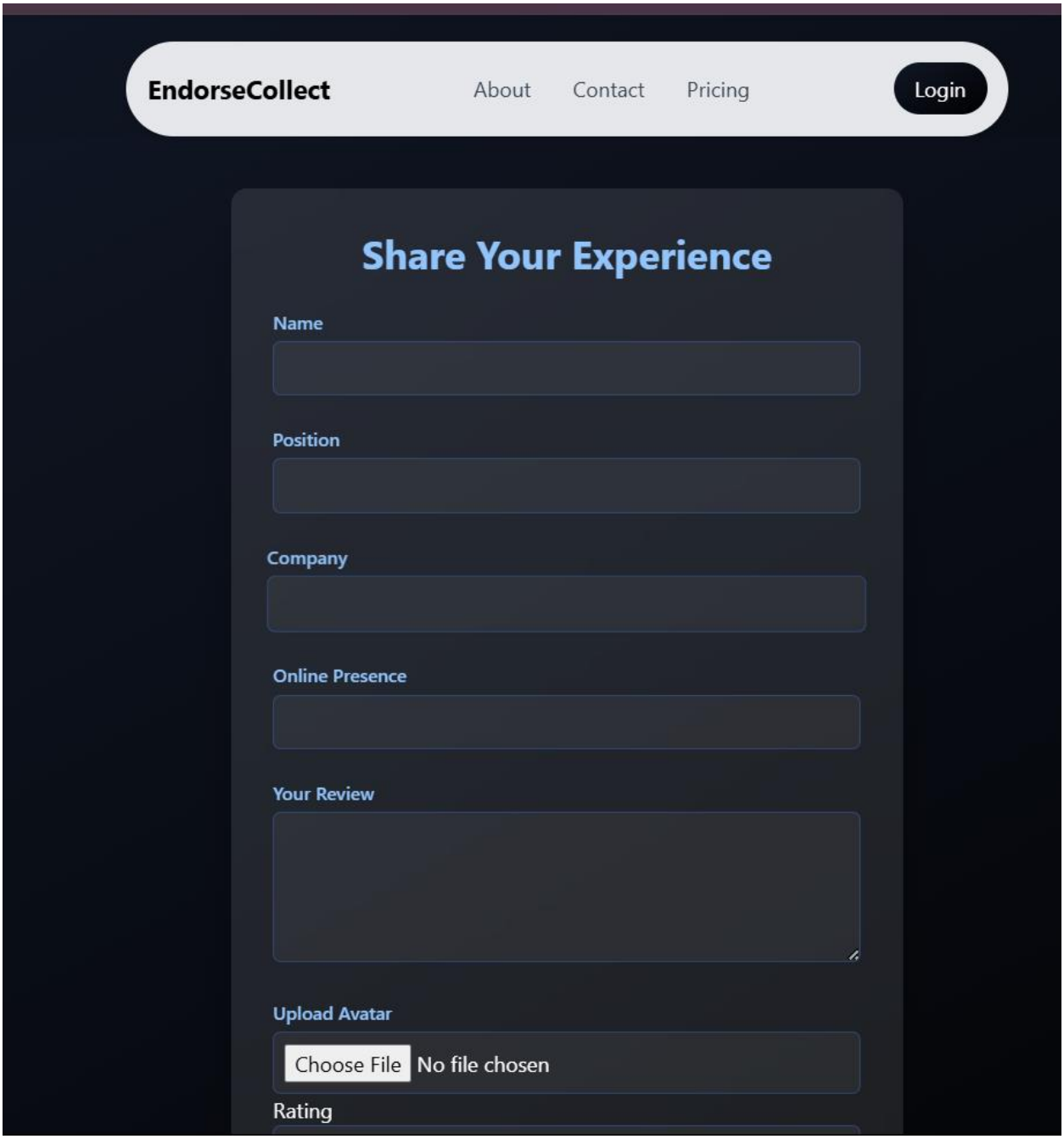
All the Reviews Given to User



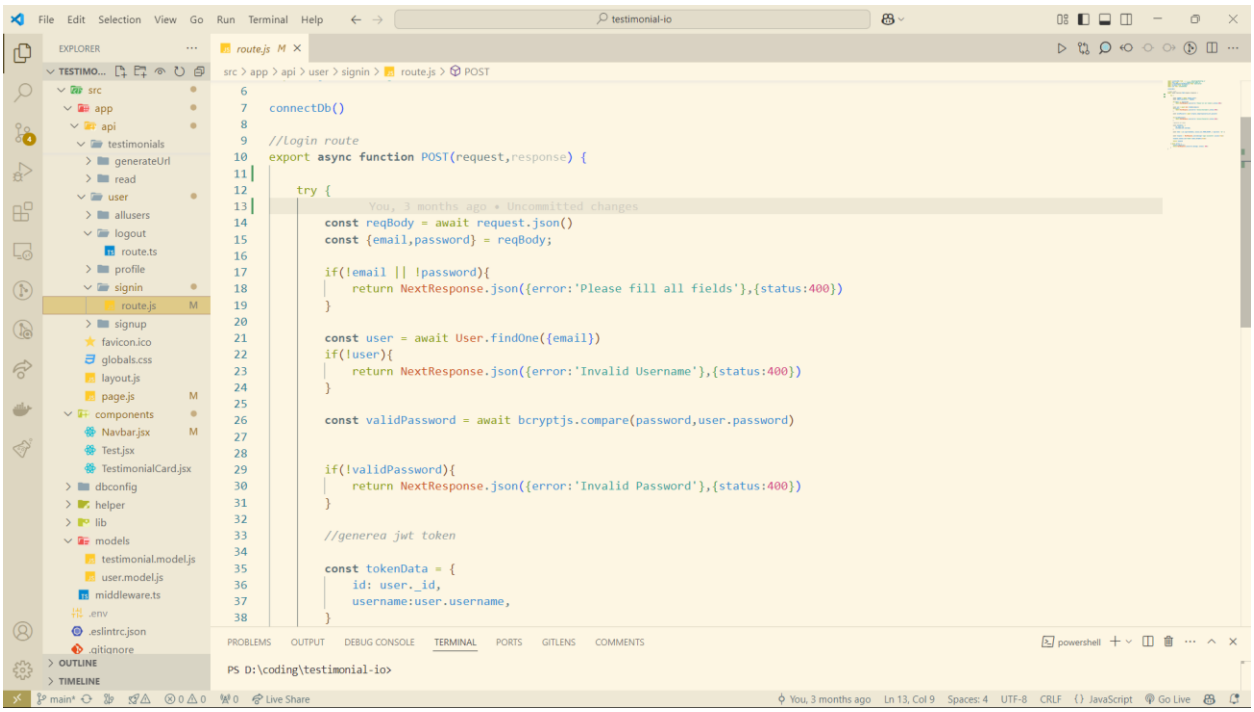
Testimonial Link Generation Page



Testimonial Form



Code :

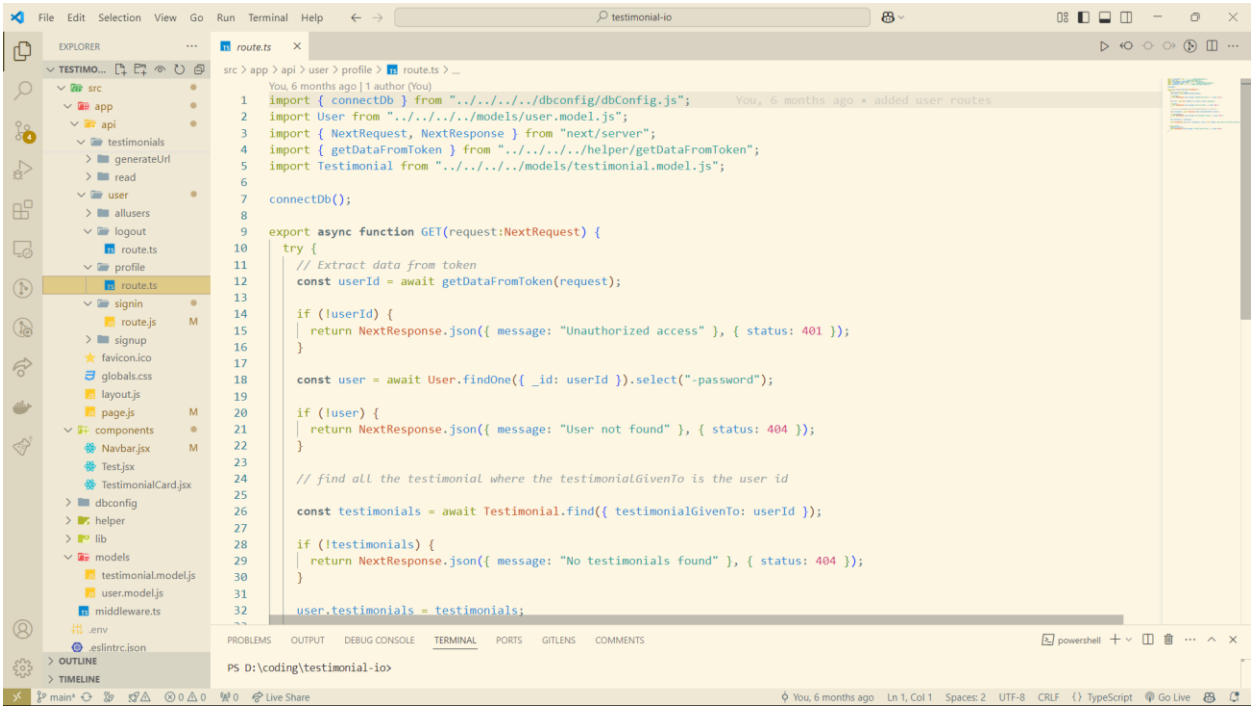


Sign In Route

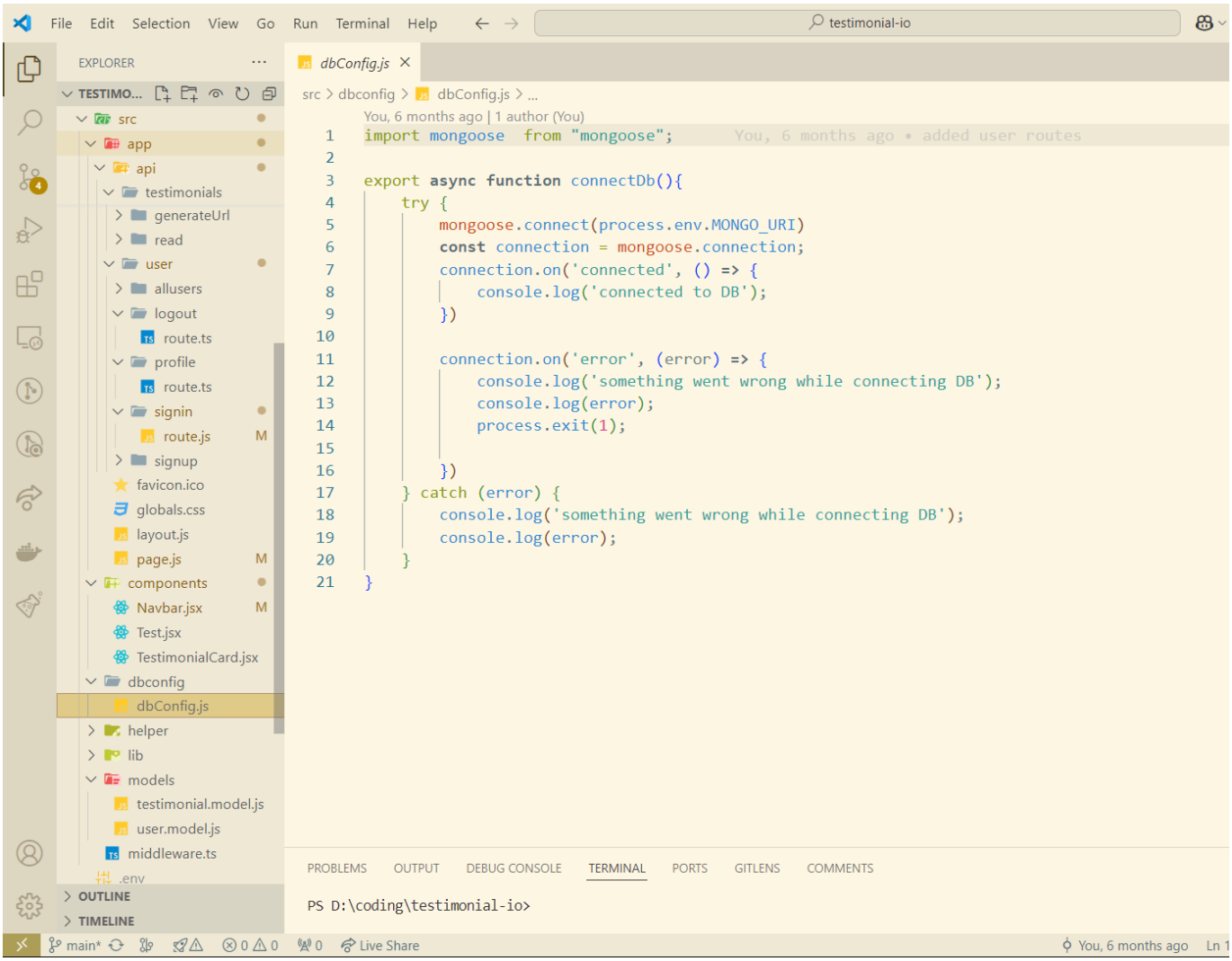
Utility Function For Retrieving User from Token

```
You, 6 months ago | 1 author (You)
import { NextRequest } from "next/server";
import jwt from 'jsonwebtoken'


export const getDataFromToken = (request:NextRequest) => {
    try {
        const token = request.cookies.get("token")?.value || "";
        const decodeToken:any = jwt.verify(token,process.env.TOKEN_SECRET!);

        return decodeToken.id

    } catch (error) {

        console.log(error.message);
    }
}       You, 6 months ago • chore: Update npm dependencies for @types/jsonw…
```

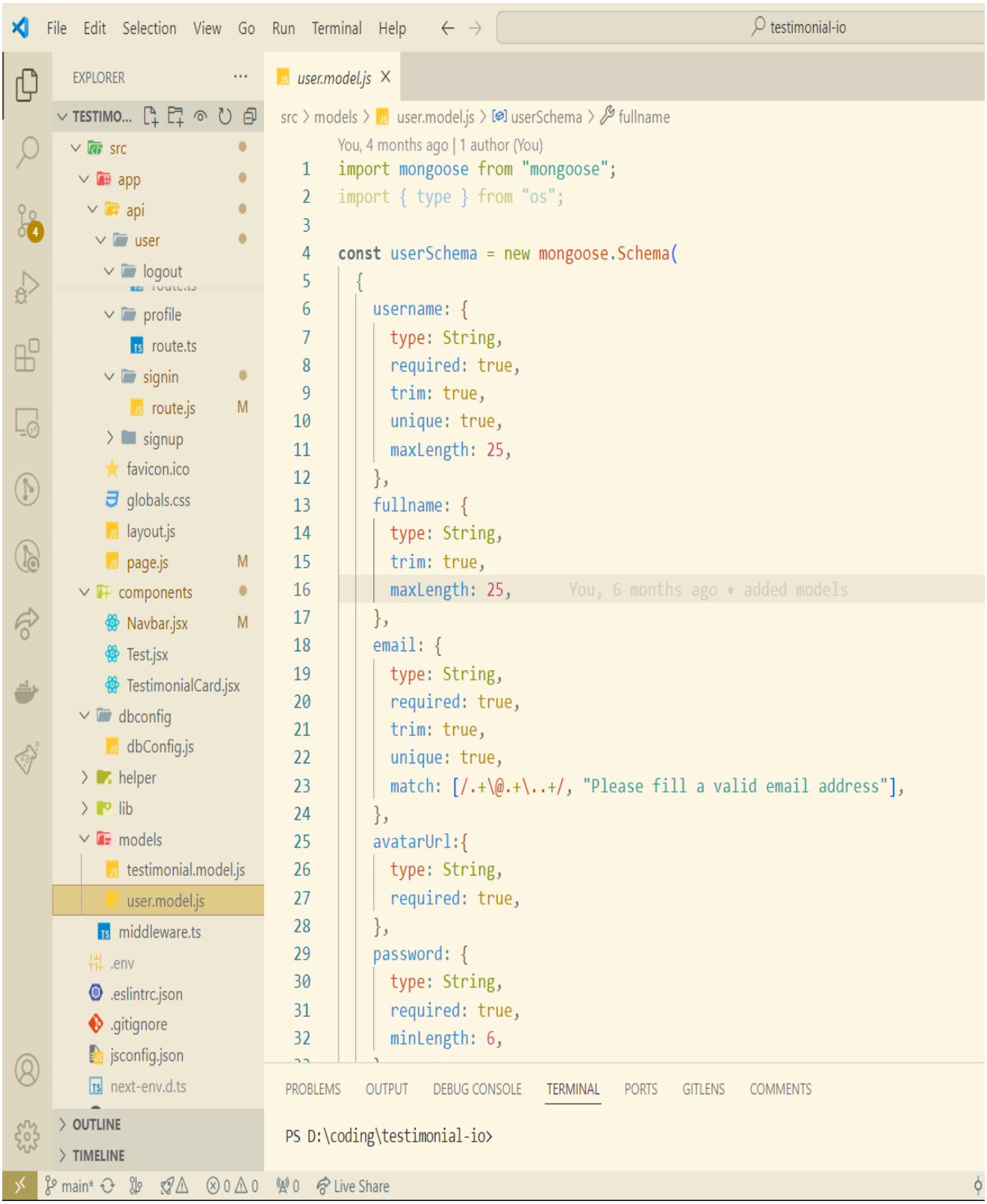
Profile Route :



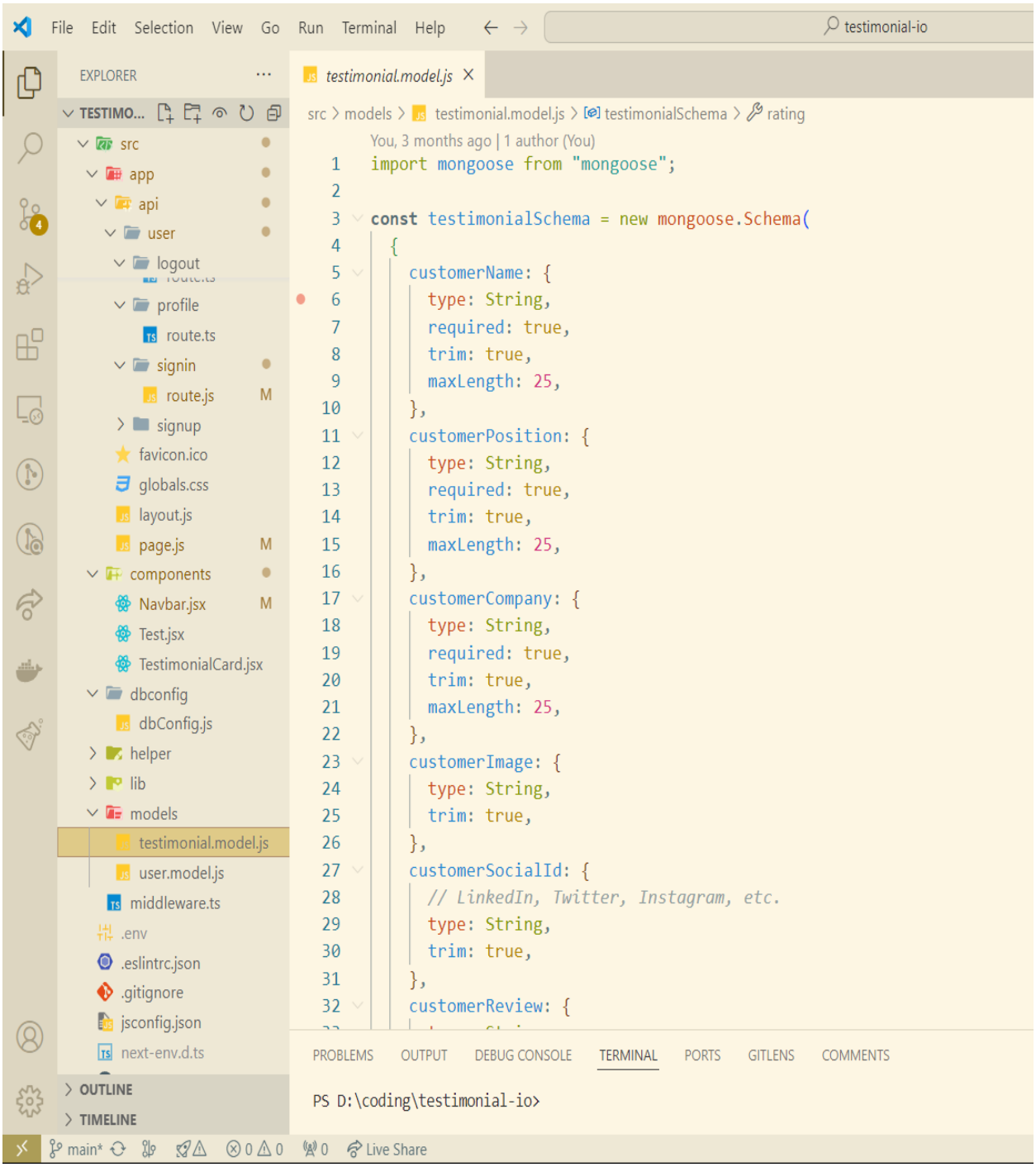
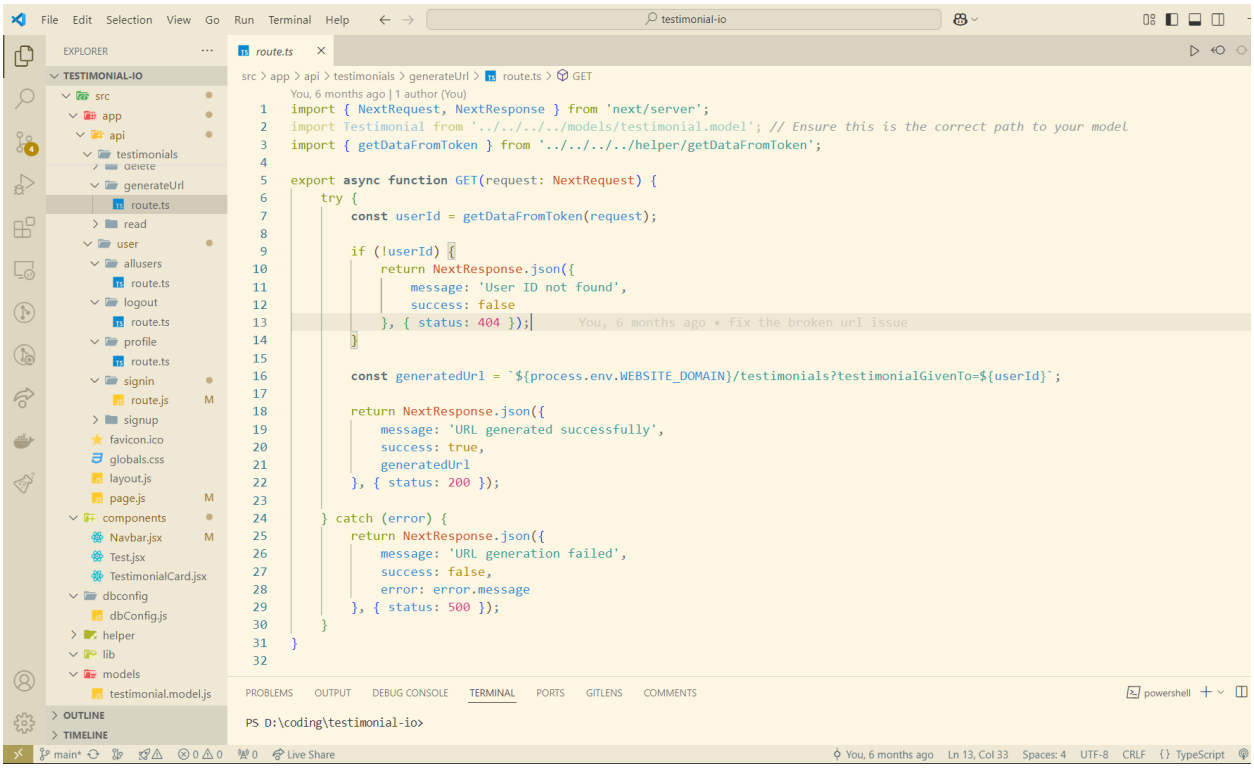
Database Connectivity Module



User Model


File Edit Selection View Go Run Terminal Help
testimonial-io
EXPLORER
user.model.js
src > models > user.model.js > userSchema > fullname
You, 4 months ago | 1 author (You)
import mongoose from "mongoose";
import { type } from "os";
const userSchema = new mongoose.Schema(
{
username: {
type: String,
required: true,
trim: true,
unique: true,
maxLength: 25,
},
fullname: {
type: String,
trim: true,
maxLength: 25,
You, 6 months ago • added models
},
email: {
type: String,
required: true,
trim: true,
unique: true,
match: [/.+\@.+\..+/, "Please fill a valid email address"],
},
avatarUrl:{
type: String,
required: true,
},
password: {
type: String,
required: true,
minLength: 6,
src
app
api
user
logout
profile
route.ts
signin
route.js
signup
favicon.ico
globals.css
layout.js
page.js
components
Navbar.jsx
Test.jsx
TestimonialCard.jsx
dbconfig
dbConfig.js
helper
lib
models
testimonial.model.js
user.model.js
middleware.ts
.env
.eslintrc.json
.gitignore
jsconfig.json
next-env.d.ts
OUTLINE
TIMELINE
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS COMMENTS
PS D:\coding\testimonial-io>
main* Live Share

Testimonial Model:



URL Generation Route