Project Title: INMAT

Subtitle: A MERN Stack Application for Skill Development and Internship

Name: Shreeraj Parmar

Date: 29/08/2024

Contact Information: 9327400960, shreeraj.on7@gmail.com

Table of Contents

Abstract/Executive Summary

The INAMT project is a web application developed using the MERN stack. It generates customizable invoices with various color options that can be downloaded in PDF format. The application is designed to enhance user experience by offering flexible invoice creation and secure multi-factor authentication via email-based verification.

In addition to its core functionality, the project integrates Google Analytics to track and analyze user interactions within the application. This feature provides valuable insights into user behavior, helping to optimize the app's performance and usability. Combining these features makes INAMT a comprehensive tool for both learning and practical application, showcasing advanced web development skills in a real-world context.

This project is intended for skill development and to demonstrate proficiency in modern web technologies, with the ultimate goal of securing an internship in the field of web development.

Introduction

Purpose:

The motivation behind developing the INAMT project was to expand my knowledge and skills in web development by working on a slightly more complex and in-depth project. Having previously completed smaller MERN stack projects, I wanted to challenge myself by exploring new technologies and concepts that I had not yet mastered. This project was designed not only to solidify my understanding of the MERN stack but also to provide hands-on experience with features such as PDF generation, API security, and user analytics.

Objectives:

The primary objectives of the INAMT project were:

- **Learn and Implement JWT Token-Based Verification:** Securing APIs through JWT tokens was a key objective, as it is essential for building secure web applications.
- **Integrate Google Analytics 4:** I aimed to understand how to incorporate real-time user analytics into a web application, allowing me to track and analyze user interactions effectively.
- **Create PDFs with Puppeteer:** Generating downloadable PDFs from dynamic data was a new area for me, and I used Puppeteer, a library created by Google, to achieve this.

- **Implement Dynamic Input Fields in React:** Enhancing the user interface with dynamic input fields in React was another goal, improving the application's interactivity.
- **Improve Project Structure:** I also focused on organizing files and components more effectively, aiming to create a well-structured and maintainable codebase.
- **How to send email through Node.**js: I want to learn how to send mail through Node.js and how to implement it in my project as two-step verification.

Scope:

The scope of the INAMT project includes:

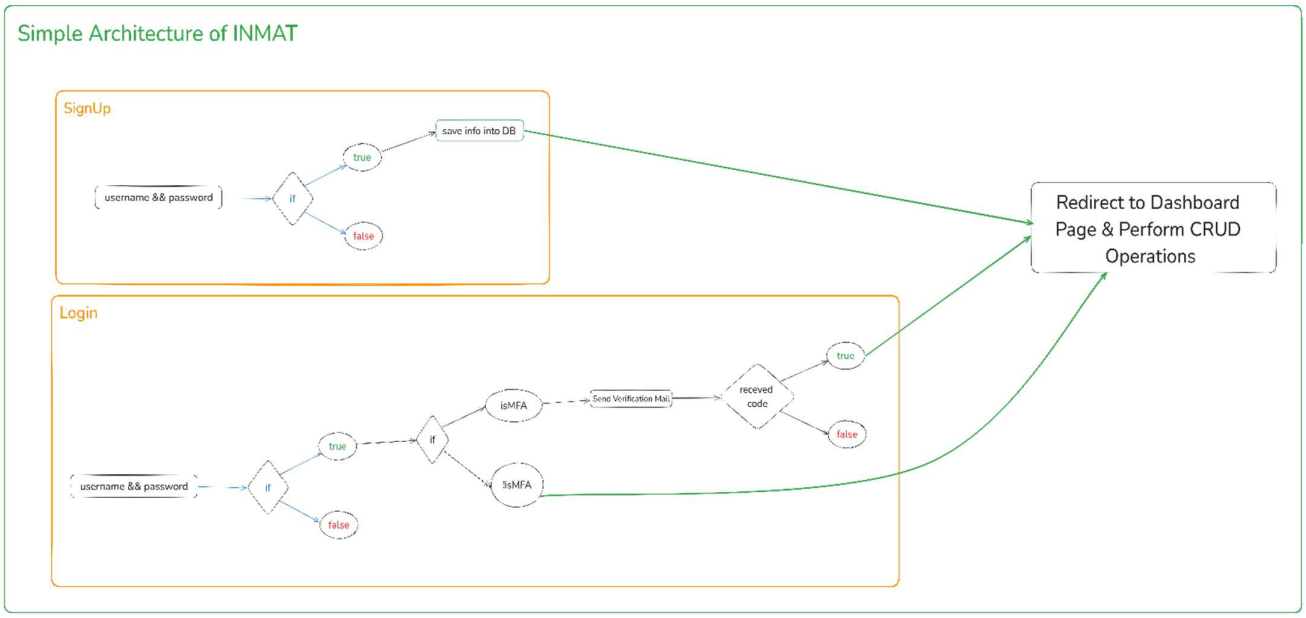
- **Covered:** The project successfully covers the implementation of JWT token-based API security, real-time user analytics through Google Analytics 4, PDF generation using Puppeteer, and dynamic input fields in React. The project is structured to ensure maintainability and scalability.
- **Not Covered:** While the project encompasses several intermediate-level features, it does not delve into very advanced topics such as server-side rendering, microservices architecture, or advanced database optimization techniques. The focus remains on learning and applying key web development concepts that are essential for building robust and secure web applications.

Technology Stack

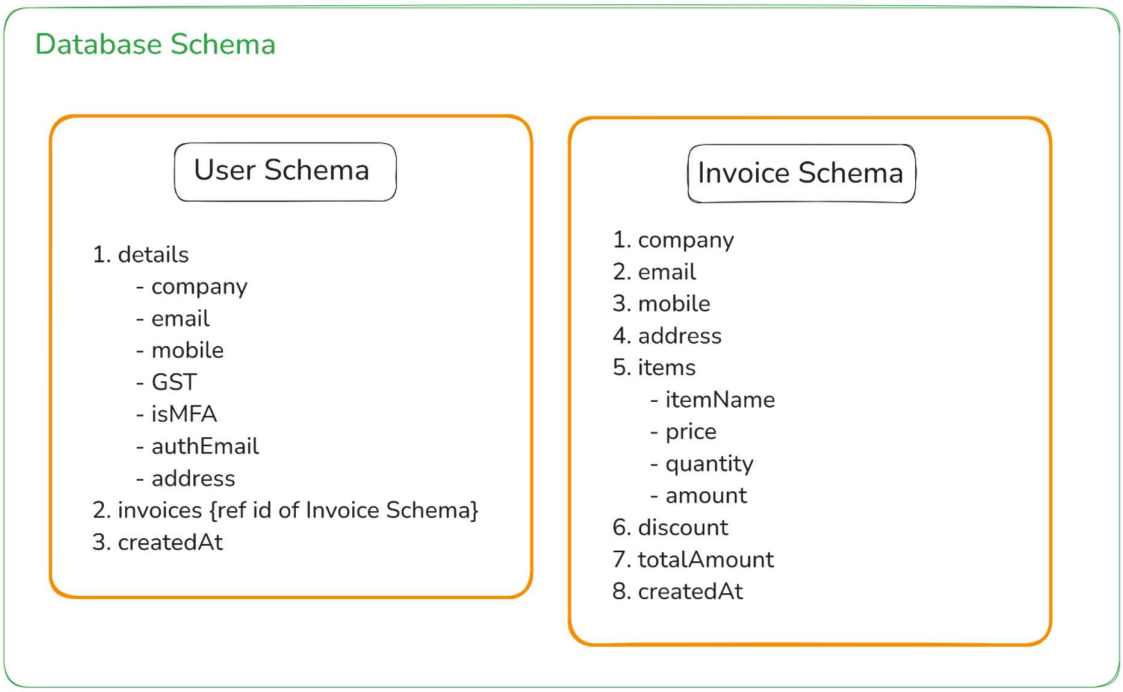
- **Frontend:**
  - React.js
  - Material-UI (MUI)
  - Tailwind CSS
- **Backend:**
  - Node.js
  - Express.js
  - MongoDB
- **Other Tools:**
  - JWT (JSON Web Tokens) for Authorization
  - Passport.js for Authentication
  - Axios for API Requests
  - Puppeteer for PDF Generation
  - Nodemailer for Email Verification
  - Google Analytics for User Interaction Tracking

Project Design and Architecture

- ## System Architecture:



- ## Database Schema:



- **File Structure:**

File Structure

client

- public
  - invoiceAssets
    - {some images assets}
- src
  - components
    - Dashboard
      - CreateInvoice.jsx
      - Menu.jsx
      - Drawer.jsx
      - Filter.jsx
      - GetStarted.jsx
      - InvoicePrintPreview.jsx
      - Loader.jsx
      - MultiFactor.jsx
      - Profile.jsx
      - RenderContent.jsx
      - Tostify.jsx
    - Hero.jsx
    - Login.jsx
    - SignUp.jsx
  - context
    - AooProvider.jsx
  - services
    - api.js
  - App.css
  - App.jsx
  - CheckInternet.jsx
  - index.css
  - main.jsx
- index.html
- package.json
- postcss.config.js
- tailwind.config.js
- vite.config.js

server

- controller
  - auth-controller.js
  - pdf-controller.js
  - mail-controller.js
  - user-controller.js
  - invoice-controller.js
- Database
  - db.js
- middleware
  - jwt-middleware.js
- model
  - user.js
  - invoice.js
- routes
  - route.js
- package.json
- index.js {main file}

also enviornmental variables included in file structure

Implementation

First I started with Figma Design, in this platform, I made a design for this project. Then I start developing the front-end, including creating components accordingly design & then I write logic code for those components.

Then I started to develop a backend and create a model, APIs & controller. I want to use MFA using the Google Authenticator app & speakeasy library, but the library not working properly so I decided to email-based verification.

ChatGPT is very helpful for creating these projects, some unknown errors & learning something new like how to implement dynamic input fields & Google Analytics & puppeteer library.

Mistakes :

1. Lack of Proper Planning:

The project started without a clear plan or deadlines, which led to inefficiencies and delays.

- **Lesson Learned:** In future projects, I will plan to set clear milestones and deadlines to ensure a more organized workflow.

2. Non-Responsive Design Initially:

React components were initially created without considering responsiveness. Responsiveness was only addressed after the project was near completion.

- **Lesson Learned:** In the next project, I will prioritize responsive design from the beginning to avoid rework.

3. Delayed Implementation of JWT:

JWT was implemented late in the development process after the APIs were already created, which complicated the integration.

- **Lesson Learned:** Security features like JWT should be integrated from the start to streamline development and avoid potential vulnerabilities.

4. AWS EC2 Deployment Challenges:

Lack of understanding of AWS EC2 led to deployment failures. Then I decided to deploy on another cloud service.

- **Lesson Learned:** I will plan to improve his knowledge of cloud deployment services to ensure smoother deployment in future projects.

5. Incomplete Project Summary:

The project summary was not written from the start, which made it difficult to document errors and solutions.

- **Lesson Learned:** Documenting the project's progress from the beginning will help in better understanding and referencing issues faced during development.

I Apply All these lesions in the next project & development process.

Deployment

- **Hosting:**
  I want to host my backend in AWS but with, a lack of knowledge in Network ACL & Security Groups I can not access public DNS or public IP to see my backend routes. I tried but did not resolve this issue, so pause this solve this problem & move to another hosting service like render.com or vercal.

  Then I deploy my backend into render. & frontend into vercel & database in mongoAtlas.

- **Deployment Process:**
  I push my code in GitHub then I deploy the backend. But here the problem is in the puppeteer library. The Puppeteer library works on a local server but not a hosted server. So I tried ChatGPT but did not solve this problem.

  Then I decide to change the pdf library, I search a lot of pdf libraries then I choose the react-pdf library. But in react-pdf library style is not working properly. Then I decided to use the previous library this is Puppeteer, I thought I solved this problem. then I searched on Google & YouTube, and finally, one video helped me to solve this problem. The solution is to use a Docker file to solve this PDF issue.

Conclusion

Summary:

The INAMT project was developed as a MERN stack application with the primary goal of enhancing web development skills and securing an internship. The project successfully met its objectives by incorporating key features such as invoice generation in multiple colors with PDF export, email-based verification for added security, and the integration of Google Analytics for real-time user interaction tracking.

The project provided valuable learning experiences in creating a responsive web application, implementing JWT for API security, and handling email verification using Nodemailer. Despite the challenges faced, including initial deployment issues and the need to revise the authentication method, the project was completed and deployed successfully, showcasing both technical proficiency and problem-solving abilities.

Future Work:

- **Feature Enhancements:** Implement additional security features such as role-based access control (RBAC) to further protect user data. Expand the dynamic input field functionality to support more complex data structures within the invoice forms.
- **Advanced Analytics:** Incorporate more advanced analytics features, such as tracking user behavior

patterns and integrating these insights with a dashboard for easier analysis.

- **Mobile Optimization:** Optimize the application further for mobile devices, ensuring seamless performance and usability across all screen sizes.
- **Additional Integrations:** Explore integrating other third-party services, such as payment gateways, to extend the functionality of the application and offer a more comprehensive invoicing solution.

These enhancements will not only add value to the project but also provide more learning opportunities, helping to further develop skills and improve the overall quality of the application.

References

Some Documentation :

- https://excalidraw.com/ --- to draw a diagram
- https://chatgpt.com/ --- for any kind of help
- https://render.com/ --- for free hosting
- https://vercel.com/ --- for free hosting
- https://mui.com --- for Pre-built React Components
- https://youtu.be/6cm6G78ZDmM?feature=shared --- for solve the problem in deployment when the puppeteer not working on a hosted server.
- https://nodemailer.com/ --- for sent verification email

INMAT code :

- Frontend: https://github.com/Shreeraj-Parmar/INMAT-client
- Backend: https://github.com/Shreeraj-Parmar/INMAT-server

Deployed Link :

https://inmat-client.vercel.app/