NextGen Tech

* Muhammad Abdullah 22-NTU-CS-1358
* Maryam Sameen 22-NTU-CS-1354
* Muhammad Kaif 22-NTU-CS-1364



Department of Computer Science, Faisalabad

BS-AI 6th

Submitted To: Mr. Zahid Javed

## Table of Contents

1.Project overview -----------------------------------------------------------------------------------------------------------------1

1.1 Introduction ------------------------------------------------------------------------------------------------1

* 1. Objective and Goals ------------------------------------------------------------------------------------1
  2. Target Audience -----------------------------------------------------------------------------------------2
  3. Scope of Project -----------------------------------------------------------------------------------------2

2. Project Design--------------------------------------------------------------------------------------------------------------------3

2.1 Data Model ---------------------------------------------------------------------------------------------------4

2.2 Process Model -----------------------------------------------------------------------------------------------5

2.2.1 Activity Diagram ------------------------------------------------------------------------------------6

2.2.2 Use-Case Diagram ----------------------------------------------------------------------------------7

3. Implementation Technologies--------------------------------------------------------------------------------------------------8

3.1 Authentication -------------------------------------------------------------------------------------------------9

3.2 MongoDB Database ------------------------------------------------------------------------------------------10

4. Website features -----------------------------------------------------------------------------------------------------------------11

4.1 Landing Page with CTA --------------------------------------------------------------------------------------11

4.2 Responsive Design and layout -------------------------------------------------------------------------------11

4.3 Reusable Component system --------------------------------------------------------------------------------11

4.4 Middleware Based Routing ----------------------------------------------------------------------------------11

4.5 SEO and Performance Optimization ------------------------------------------------------------------------11

5. User Interface --------------------------------------------------------------------------------------------------------------------12

5.1 Landing Page ----------------------------------------------------------------------------------------------------12

5.2 Registration -------------------------------------------------------------------------------------------------------13

5.2.1 Login --------------------------------------------------------------------------------------------------13

5.2.2 Sign up ------------------------------------------------------------------------------------------------14

5.3 Startups -----------------------------------------------------------------------------------------------------------15

5.4 Contact Forn -----------------------------------------------------------------------------------------------------16

5.5 Admin Panel -----------------------------------------------------------------------------------------------------17

6. Limitations and Future Scope -------------------------------------------------------------------------------------------------18

7. Conclusion -----------------------------------------------------------------------------------------------------------------------20

## List Of figures

Figure 2.1: Flowchart ---------------------------------------------------------------------------------------------------------------3

Figure 2.2: ERD Diagram ----------------------------------------------------------------------------------------------------------4

Figure 2.3: Activity Diagram -------------------------------------------------------------------------------------------------------6

Figure 2.4: Sequence Diagram -----------------------------------------------------------------------------------------------------7

Figure 3.1: Next.js Workflow ------------------------------------------------------------------------------------------------------9

Figure 5.1: Landing Page ----------------------------------------------------------------------------------------------------------12

Figure 5.2: Login Form ------------------------------------------------------------------------------------------------------------13

Figure 5.3: Sign up Form ----------------------------------------------------------------------------------------------------------14

Figure 5.4: Startups -----------------------------------------------------------------------------------------------------------------15

Figure 5.5: Contact Form ----------------------------------------------------------------------------------------------------------16

Figure 5.6: Admin Panel -----------------------------------------------------------------------------------------------------------17

## List of Tables

Table1: User Table -------------------------------------------------------------------------------------------------------------------5

Table2: Services Table --------------------------------------------------------------------------------------------------------------5

Table3: Content Table ---------------------------------------------------------------------------------------------------------------5

## Chapter 1

1 Project Overview

## Introduction

In today’s fast-paced digital landscape, startup needs robust, scalable and engaging web platforms to efficiently connect with their audience and grow their ventures. The tech startup presented here is a modern web application designed to serve as a foundational launchpad for innovative businesses looking to establish a strong online presence.

This web app offers a streamlined and professional interface built using Next.js, enabling startups to showcase their services, highlight key features, and engage with customers through a responsive, high-performance website. Whether it’s introducing a product, building brand trust, or collecting leads, this platform simplifies the process of building and managing a compelling digital experience. With a clean design, modular architecture, and deployment-ready environment, this project serves as an ideal starting point for any early-stage startup aiming to make an impact in the tech-driven world.

## 1.2 Objective and Goals

**Objective:**

The primary objective of this project is to develop a fully functional and professionally designed web platform tailored for tech startups. This application aims to help emerging ventures establish a digital presence, communicate their mission, and engage with potential users or clients. The platform is built to deliver a fast, scalable, and user-centric experience by utilizing modern technologies like Next.js and TypeScript, making it ideal for showcasing services, onboarding users, and growing a startup’s reach online.

**Goals:**

* **Modern and Clean UI:** Develop a visually appealing and intuitive interface that aligns with current UI/UX best practices, ensuring that users can easily navigate, access key content, and interact with the platform seamlessly.
* **Responsive and Mobile-friendly design:** Ensure full responsiveness across devices, allowing users to have a consistent and optimized experience whether they access the platform on a desktop, tablet, or mobile phone.
* **Component-Based Architecture:** Build the platform using reusable and scalable components, enabling fast development cycles, easier maintenance, and the flexibility to expand functionality as the startup evolves.
* **Deployment and Configuration Flexibility:** Support environment-based configuration using .env files to facilitate easy deployment across staging, development, and production environments using platforms like Vercel.

### 1.3 Target Audience

NextGen Tech is designed for a diverse group of users and stakeholders who are integral to the growth and functionality of a modern online business. The platform targets:

* **Early-Stage Startup Founders:** Entrepreneurs seeking a professional and scalable digital presence to validate their business ideas, attract investors, or begin customer acquisition.
* **Tech Teams and Developers:** Developers who want a clean, modular starting point to quickly build and iterate on startup features without reinventing core frontend structure.
* **Potential Clients and Users:** Visitors who are interested in the startup's

offerings—whether it's a product, service, or platform—and need a clear, fast, and engaging interface to explore what’s offered.

* **Investors and Incubators:** Stakeholders looking to evaluate a startup’s tech readiness and digital branding before considering funding or partnership.

### 1.4 Scope Of Project:

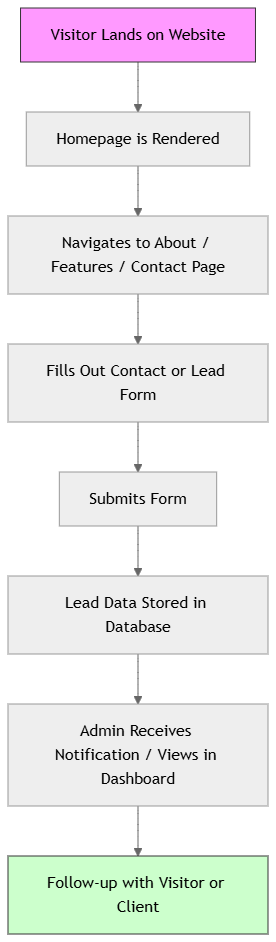
The scope of this project encompasses the creation of a feature-rich, adaptable frontend platform for tech startups, with room for future backend integrations. Key deliverables include:

* **Responsive Frontend Interface:** A fully functional, responsive frontend developed using Next.js and TypeScript with clean design principles suitable for public product pages, landing content, and company information.
* **Scalable Codebase:** A component-based architecture enabling seamless code reuse, rapid iteration, and straightforward extension for future feature additions such as authentication, dashboards, or APIs.
* **Custom Middleware and Routing Logic:** Implementation of basic middleware routing logic, preparing the app for future use cases like protected routes and dynamic user flows.
* **Environment-Specific Deployment:** Built-in support for development, staging, and production environments using .env configuration, simplifying the deployment process across hosting platforms.

## Chapter 2

## 2 Project Design

To design a web app, thoughtful planning of the database structure is essential to ensure scalability, maintainability, and data integrity. The design process is divided into two parts: the **data model**, which outlines the entities and relationships within the system, and the **process model**, which describes how data will be manipulated via application logic and queries.



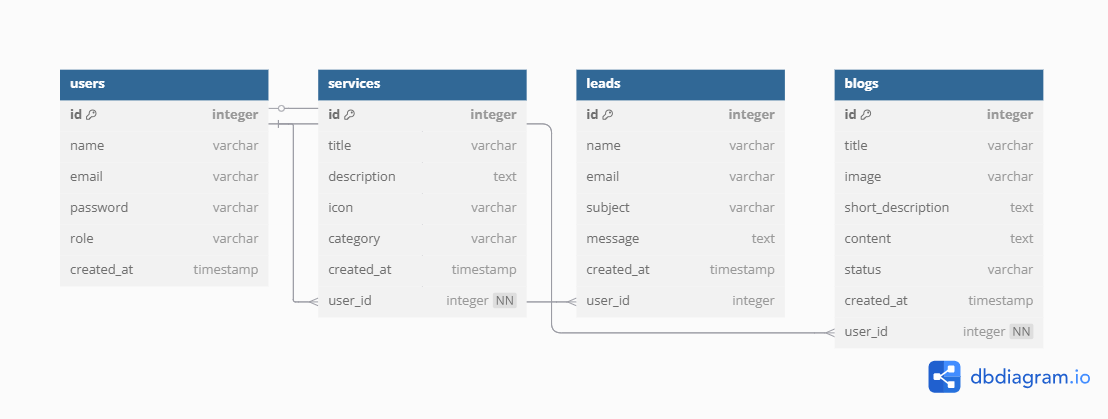
*Figure 2.1: Flowchart*

**Explanation:**

* A user navigates through the frontend pages.
* Engages with a call-to-action form (like "Contact Us").
* Upon submission, the backend stores the info.
* Admin sees this lead and can take follow-up actions.

### 2.1 Data Model

A data model provides a blueprint of how data is structured in the system. For this tech startup platform, the application primarily handles informational content, users, and possibly client leads. Below is the Entity-Relationship Diagram (ERD) followed by the description of key data collections.



*Figure 2.2: ERD Diagram*

**Explanation:**

* USER: Represents registered users/admins of the site.
* Services Stores modular content blocks/pages (like About, Features).
* LEADS: Captures user-submitted inquiries or contact form messages.
* Blogs: Represents blogs added by the user.
* Relationships:
  + A USER may manage multiple PAGECONTENT entries.
  + A USER may also be related to incoming LEADS (e.g., an admin reviewing them).

### **Database Design:**

**User:**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Type** | **Description** |
| **\_id** | ObjectID | Primary Key |
| **Name** | String | Full name of the user |
| **Email** | string | Email address (unique) |
| **Password** | string | Hashed password for login |
| **role** | string | Role of the user |
| **createdAt** | Date | Registration timestamp |

*Table1: User Table*

**Services:**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Type** | **Description** |
| **\_id** | ObjectID | Primary Key |
| **title** | String | Name of the startup |
| **description** | string | Short description |
| **category** | string | Category label |
| **createdAt** | Date | Timestamp of the service |

*Table2: User Table*

**Content:**

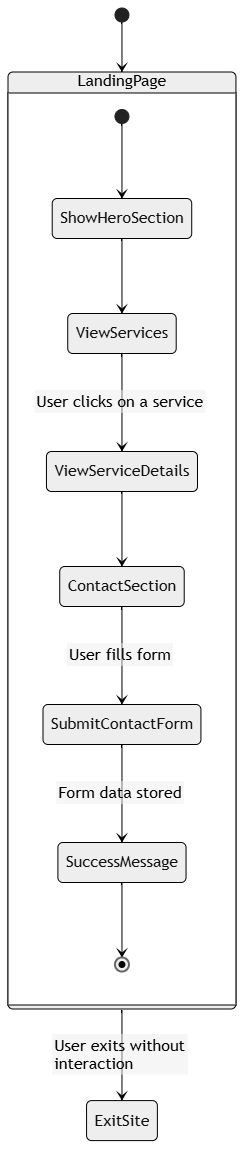
|  |  |  |
| --- | --- | --- |
| **Attribute** | **Type** | **Description** |
| **\_id** | ObjectID | Primary Key |
|  |  |  |
| **title** | string | Title of page section |
| **body** | string | Rich text or markdown content |
| **updatedAt** | Date | Last updated timestamp |

*Table3: Content Table*

### 2.2 Process Model

The process model defines how operations are performed on the data. It involves crafting efficient queries and mutation logic that align with both performance goals and the user experience. This model ensures that all CRUD (Create, Read, Update, Delete) operations can be securely and efficiently performed.

### 2.2.1 Activity Diagram

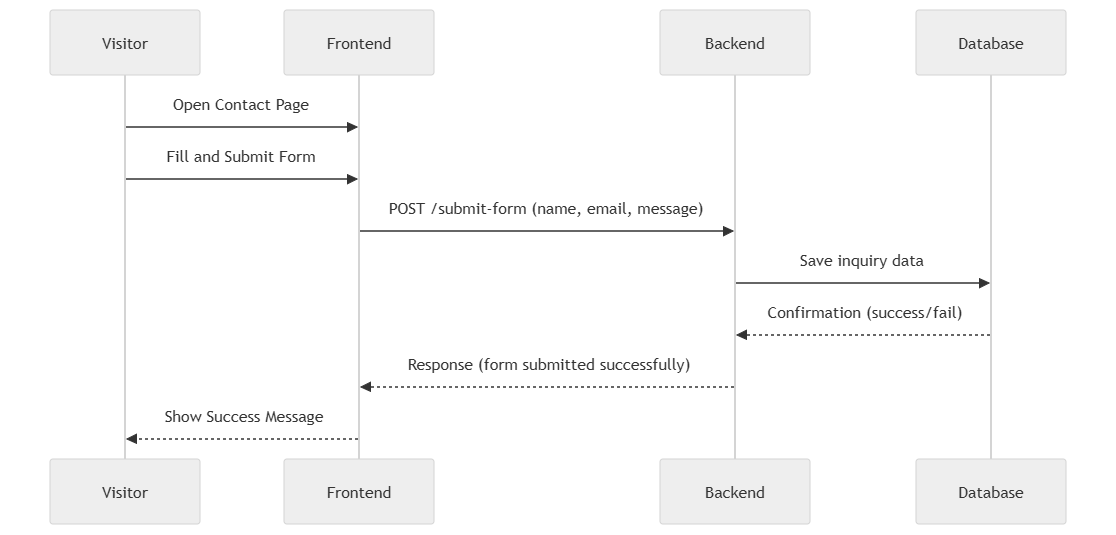


*Figure 2.3: Activity Diagram*

**Explaination:**

* LandingPage is a composite state that groups user navigation behavior on the homepage.
* Transitions represent user interactions (e.g., viewing services, clicking, submitting a form).
* The diagram also includes an alternate path where the user exits early.

### 2.2.2 Sequence Diagram



*Figure 2.4: Sequence diagram*

**Explanation:**

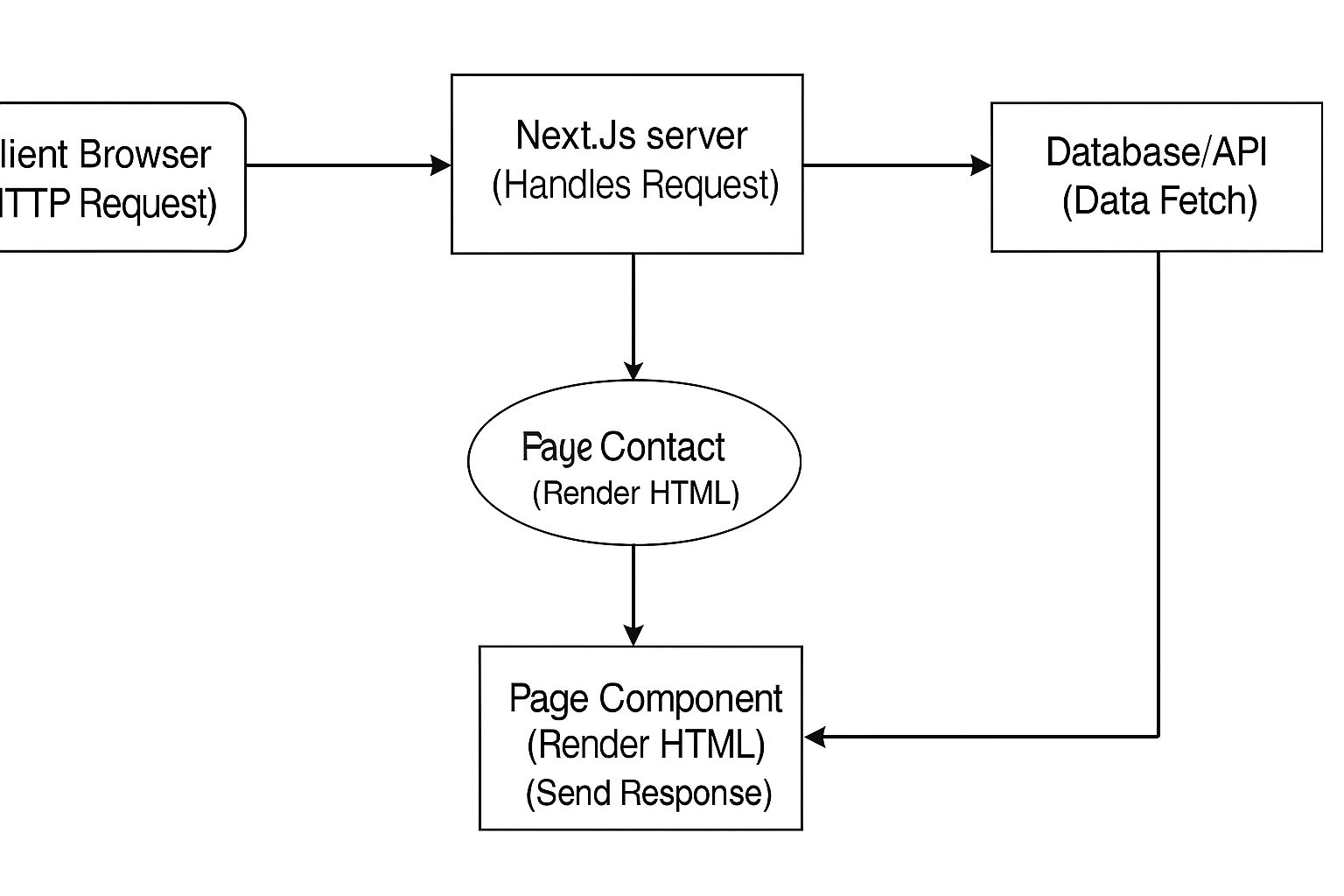
* Visitor initiates contact via a form
* The Frontend collects and submits data
* The Backend processes and stores it in the Database
* The user receives feedback once the process is complete

## Chapter 3

## 3 Implemented Technologies

The scope of this tech startup web app is defined by its objective: to provide startups with a scalable, professional, and responsive platform to represent their services, engage users, and establish a digital presence. This is achieved through a modern technology stack focused on performance, maintainability, and user experience. Key implementations include:

* **Node.js Web Server:** The application uses a Node.js-powered backend, enabling non-blocking I/O and fast HTTP request handling. The Node.js server listens for HTTP requests such as page loads, form submissions, and API calls, responding with static pages or dynamic content depending on the request.
* **Next Js Framework:** The app is built using the Next.js framework, a powerful React-based tool for creating server-rendered and statically generated web apps. Benefits include:
  + **Server-Side Rendering (SSR):** Improves SEO and performance by pre-rendering pages on each request.
  + **Static Site Generation (SSG):** Speeds up loading by generating pages at build time.
  + **Hybrid Routing:** Enables both page-based routing and dynamic API routes.
* **User Interaction and Forms:** The platform allows users to:
  + Browse the landing page
  + View a curated list of services (e.g., AI, Cybersecurity, DevOps)
  + Submit contact or inquiry forms
* **Database Integration:** The app integrates with a NoSQL database (e.g., MongoDB or Firebase) to store:
  + User details
  + Service Listings
  + Inquiry Submission
* **Environment Configuration:** Environment variables are managed via .env files for easy switching between development, staging, and production environments. This supports secure API keys, database URIs, and site-specific configs.
* **Data Fetching Methods in Next.js:** 
  + **getMethod:** Used for server-side rendering of key pages like Services and About.
  + **getMethod:** Ideal for static marketing pages like Home or About that don’t change frequently.
  + **API Routes:** Built-in API endpoints (e.g., /api/contact) handle server logic such as storing form submissions.



*Figure 3.1: Next.js Workflow*

### 3.1 Authentication with Clerk

We chose **Clerk** as our authentication solution over traditional libraries like NextAuth.js because Clerk offers a comprehensive, fully managed identity platform that simplifies user authentication, authorization, and user management out of the box. Clerk provides pre-built components for sign-up, sign-in, user profile, and session handling, which significantly accelerated our development process and ensured a seamless user experience with minimal configuration. Unlike NextAuth.js, which often requires manual setup of providers, session logic, and database integrations, Clerk handles these features with a modern API-first approach and built-in security. Additionally, Clerk supports OAuth, passwordless login, multi-factor authentication (MFA), and role-based access control-making it a scalable and secure choice for our tech startup platform. Its developer-friendly dashboard and real-time monitoring tools also make it easy to manage users and sessions, which aligns well with our goal of building a secure and efficient platform with minimal overhead.

.

### 3.2 MongoDB Database

In this tech startup platform, **MongoDB** is used as the backend database to store dynamic application data. MongoDB is a **NoSQL, document-oriented** database known for its flexibility, scalability, and performance-making it well-suited for modern applications with evolving data requirements.

Unlike relational databases that use rigid schemas and tables, MongoDB stores data in **JSON-like documents (BSON)**. This schema-less design allows the application to evolve quickly as new fields or entities are introduced-perfect for a growing startup that may later add new services, features, or client interactions.

**Key Advantages for the Startup Platform:**

* **Flexible Schema:** Allows rapid iteration and adaptation of data models such as user info, leads, or service listings.
* **High Performance:** Handles real-time reads/writes efficiently, ideal for dynamic frontends and CMS-like behavior.
* **Horizontal Scalability:** Can scale across distributed systems using sharding, allowing seamless handling of increased traffic as the startup scales.
* **Open Source Ecosystem:** Supported by a vibrant community and vast tooling ecosystem (MongoDB Atlas, Compass, etc.).

**Integrating MongoDB with Mongoose:**

To interface with MongoDB efficiently, the application uses Mongoose, a powerful ODM (Object Data Modeling) library for Node.js. Mongoose allows defining schemas for each collection (like users, services, leads) and enforces validation and structure at the application level.

**How it Works:**

* A connection URI is stored in the .env file for secure and environment-specific access.
* On each request (API or server-side), Mongoose connects (or reuses an existing connection) to interact with the MongoDB database.
* Collections like User, Service, and Lead are modeled using Mongoose schemas and mapped to their respective documents in the database.

## Chapter 4

## 4 Website Features

This tech startup platform is built to provide a streamlined, modern, and scalable experience for showcasing services and engaging with potential clients or partners. The web app includes both public-facing features and admin-side capabilities for content management and user interaction.

### 4.1 Landing Page with CTA

The application features a modern landing page designed to introduce the startup's mission and offerings. It includes strategic **call-to-action (CTA)** buttons that guide users to explore services, sign up, or contact the team.

### 4.2 Responsive Design and Layout

The website is fully responsive, adapting seamlessly to desktops, tablets, and mobile devices. Layouts are optimized using a grid system and media queries, ensuring a consistent and intuitive user experience across screen sizes.

### 4.3 Reusable Component System

Built using modular architecture, the app incorporates a collection of **reusable UI components** for buttons, cards, layouts, and forms. This allows for fast development and scalability as the project evolves.

### 4.4 Middleware-Based Routing

With custom middleware (middleware.ts), the system supports conditional routing and future-ready **authentication logic**. This ensures protected routes and smooth handling of dynamic content without full page reloads.

### 4.5 SEO and Performance Optimization

Leveraging Next.js's features like **server-side rendering** and **image optimization**, the app is built for high performance and SEO readiness, helping the startup rank better in search engines and load faster for users.

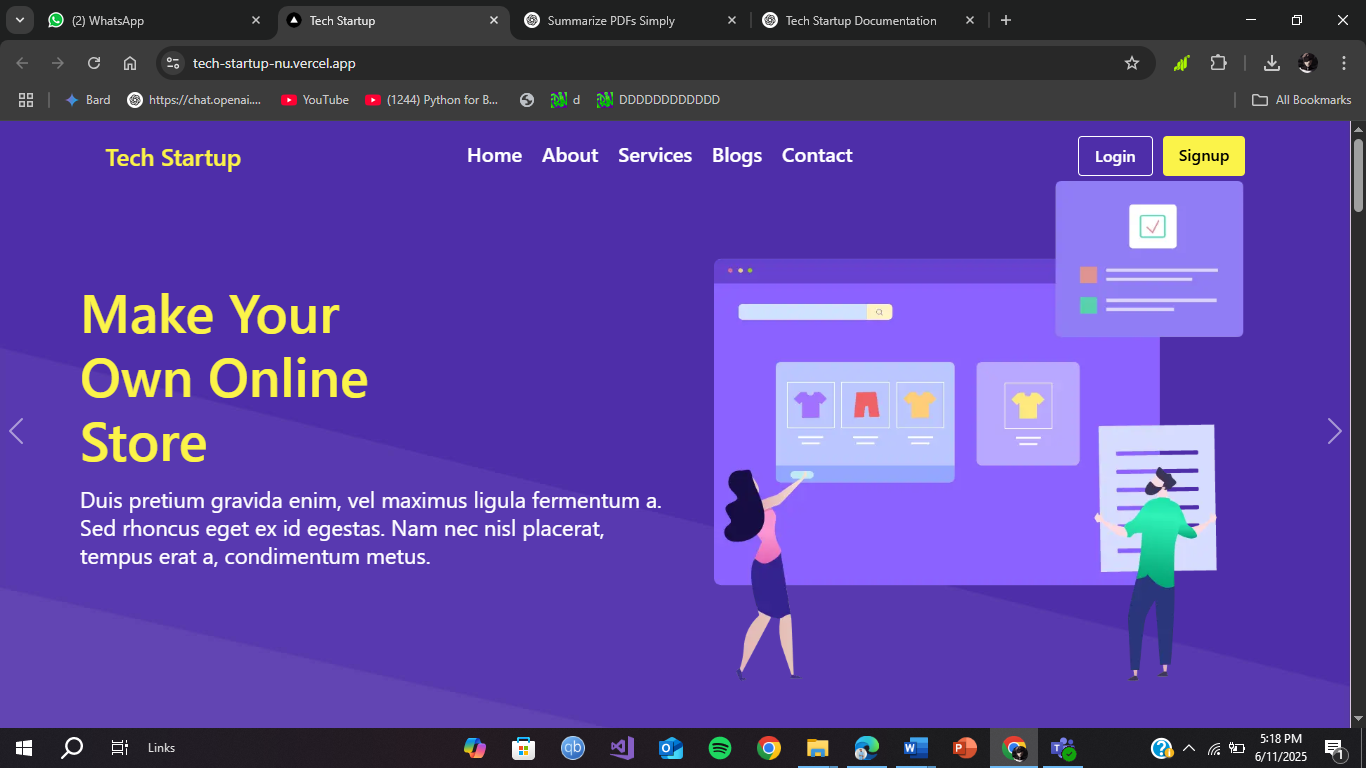
## Chapter 5

## 5 User Interface

The objective of this web application is to provide **tech startups** with a professional and interactive online platform where they can showcase their services, establish credibility, and connect with potential clients or investors. The platform is designed to be **user-friendly**, visually appealing, and responsive across devices, delivering an excellent user experience from landing to contact.

### 5.1 Landing Page

When a visitor enters the web address in the browser, they are directed to the **landing page**, which serves as the homepage of the startup. This page is thoughtfully designed with modern aesthetics and strong visual hierarchy to immediately communicate the startup’s value proposition.



*Figure 5.1: Landing Page*

**Key Features of the Landing Page:**

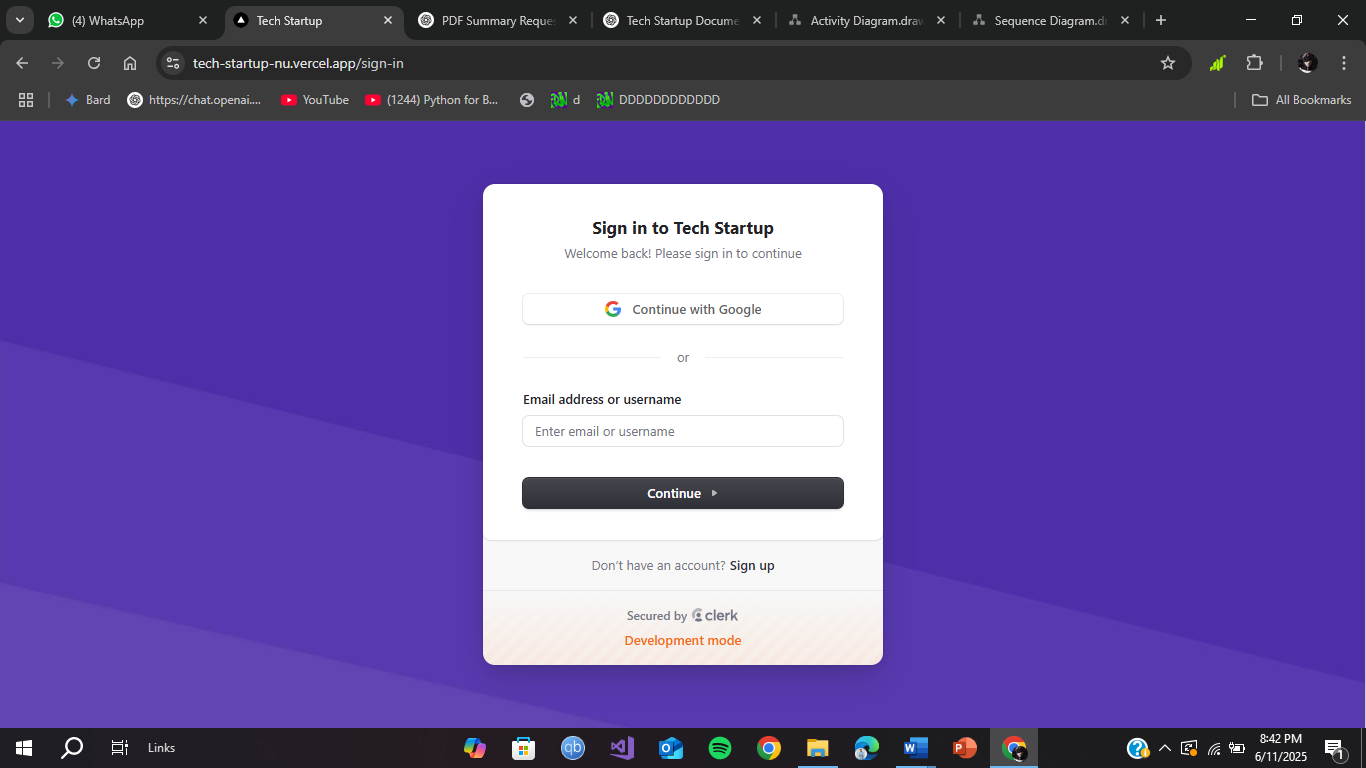
* **Hero Section:** A large banner area featuring a headline, subheading, and a call-to-action button (e.g., “Get Started” or “Explore Services”).
* **Startup Introduction:** A short paragraph that introduces the startup’s mission and core values.
* **Call-to-Action (CTA):** Multiple CTAs are placed strategically to encourage engagement such as scrolling, exploring services, or submitting a query.
* **Smooth Navigation:** The navbar enables easy access to sections like Services, About Us, Contact, and more.
* **Responsive Slider or Showcase:** A service slider or grid layout dynamically showcases key services or features offered by the startup.

### 5.2 Registration

The registration Page consists of Login and Sign up located on the homepage. On the top right corner, it can be found.

### 5.2.1 Login

For login, the user can either his his/her google account or sign in using the email id or username and password.



*Figure 5.2 Login Form*

**Explanation:**

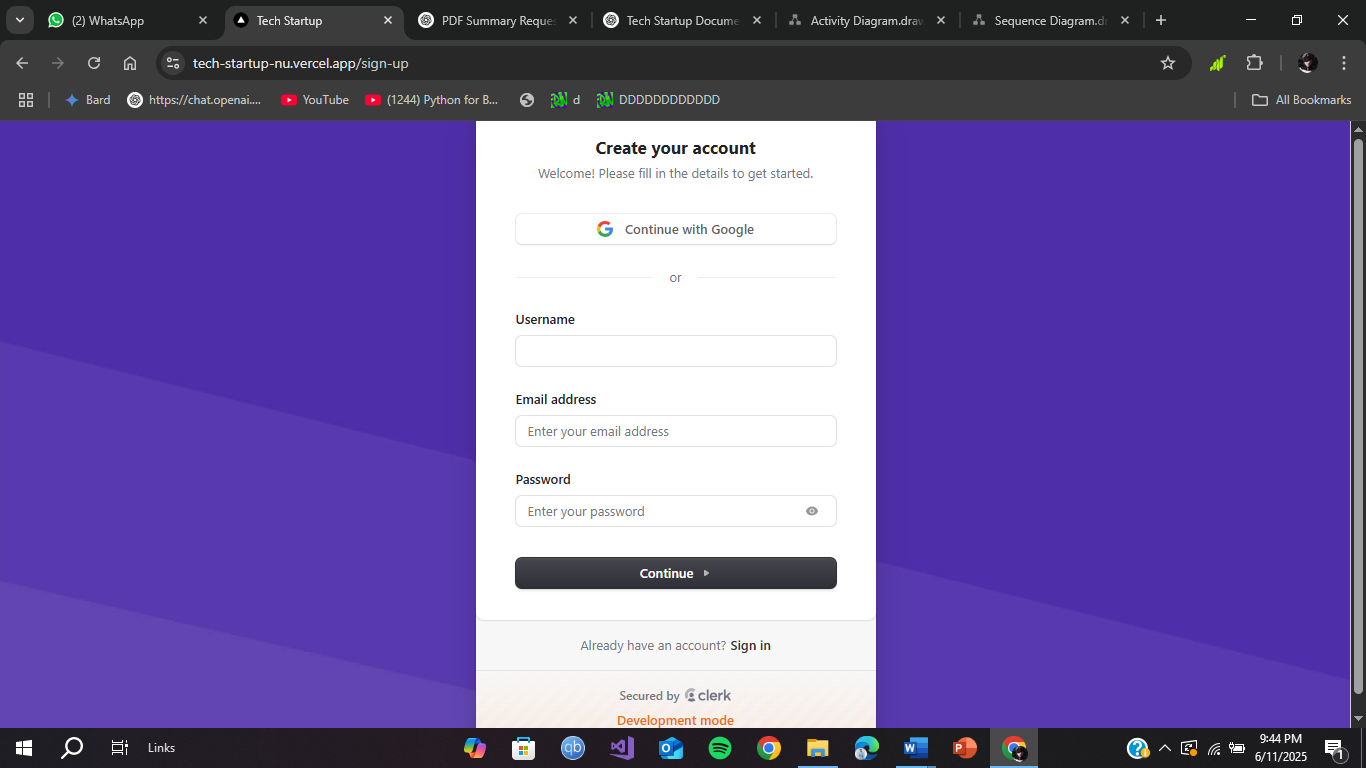
* Clean, centered card layout with a modern purple gradient background
* Option to sign in using Google or with email/username credentials
* Secured by Clerk for reliable authentication and session management
* Simple, intuitive design with clear form fields and call-to-action buttons
* Responsive layout optimized for all screen sizes
* Includes a link for new users to sign up and a welcoming message for returning users

### 5.2.2 Sign up

If a user did not sign in, he/she can sign up by providing the required credentials. For this, the user can use google account or can sign up using Username, Email Address and Password.

* **Username:** Name of the user
* **Email Address:** Email Address of the user (must be valid)
* **Password:** At least 8 characters

If any or all the credentials are taken by another user, the current credentials can’t be used again.



*Figure 5.3: Sign up Form*

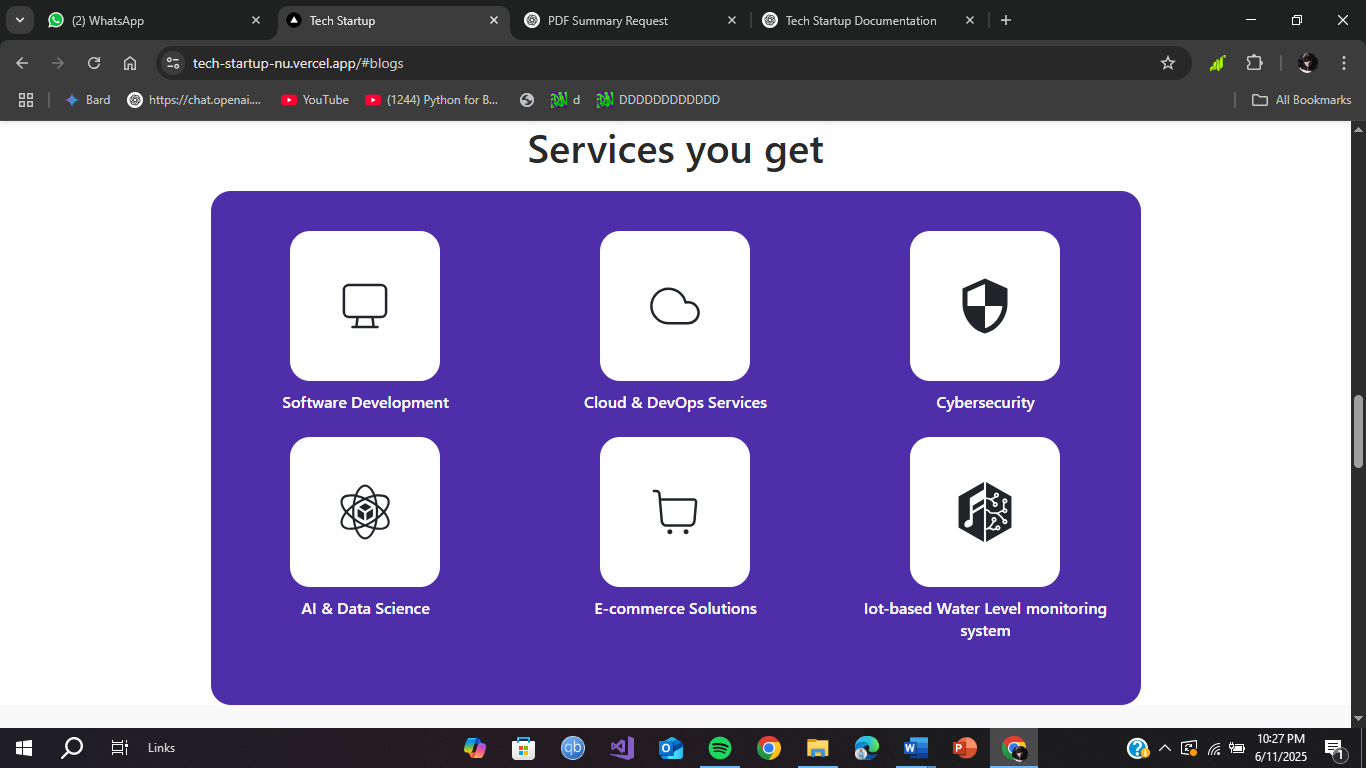
**Explanation:**

* Clean and centered layout with a modern purple background design
* Allows sign-up using Google or manual form input (username, email, password)
* Form fields are clearly labeled and include placeholder guidance
* Secure password input with visibility toggle for user convenience
* “Continue” button enables smooth submission after validation
* Includes a redirect link for users who already have an account to sign in

### 5.3 Startups

NextGen Tech offers a variety of startups including:

* Software Development
* Cloud & DevOps Services
* Cybersecurity
* AI & Data Science
* E-Commerce Solutions
* IoT based water level Monitoring System



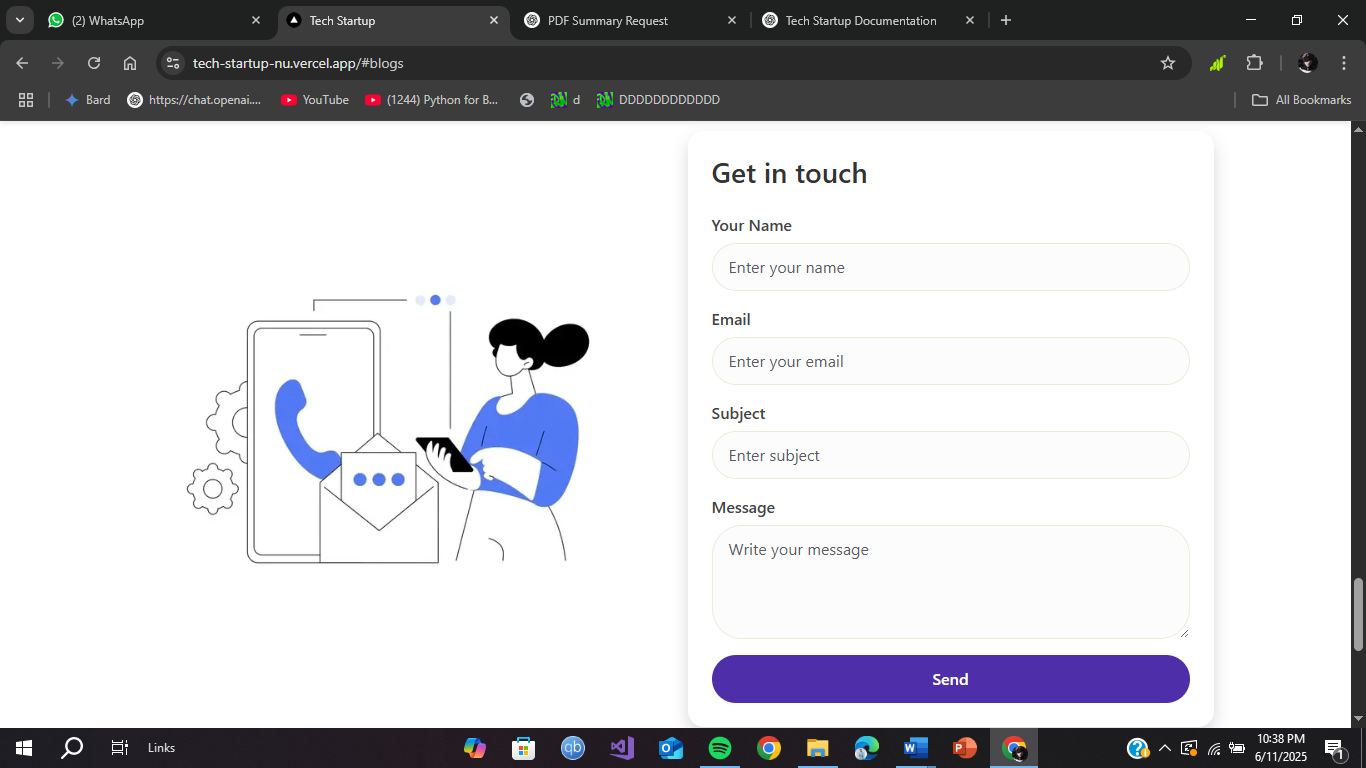
*Figure 5.4: Startups*

**Explanation:**

* Section titled “Services you get” prominently introduces the startup’s offerings
* Services are displayed in a clean, responsive 3x2 grid layout
* Each service is represented with a minimalistic icon for visual clarity
* Titles include diverse offerings like Software Development, AI, Cybersecurity, etc.
* Background uses a vibrant purple container for visual separation and emphasis
* Layout ensures consistency across desktop and mobile screen sizes

### 5.4 Contact Form

The "Get in Touch" section of the tech startup web application features a clean, user-friendly contact form designed to facilitate smooth communication between visitors and the startup team. Positioned within a visually balanced two-column layout, the left side features a minimalistic illustration depicting digital communication, reinforcing the modern and tech-forward theme of the platform. On the right side, a rounded, card-like form presents four essential input fields: name, email, subject, and message-each clearly labeled with placeholder text to guide the user through the process. The form design is intuitive and responsive, ensuring usability across devices. A prominent purple "Send" button at the bottom matches the site’s branding and invites user interaction. This section ensures that visitors can easily reach out for inquiries, partnership opportunities, or support, enhancing the platform’s professionalism and accessibility.



*Figure 5.5: Contact Form*

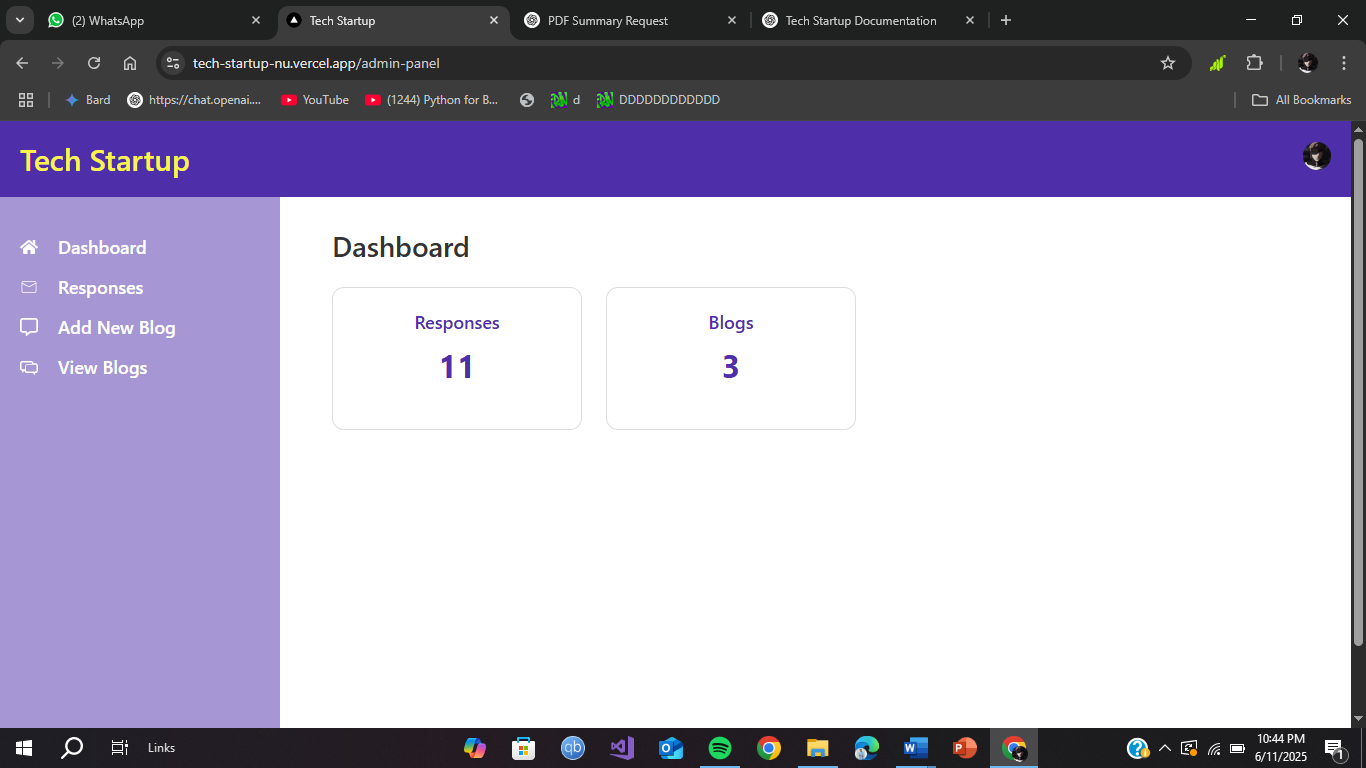
**Explanation:**

* Clean two-column layout with an illustration on the left and form on the right
* Clearly labeled input fields for name, email, subject, and message
* Rounded input boxes and smooth spacing enhance visual appeal and readability
* Large, prominent purple "Send" button encourages user interaction
* Fully responsive design ensures functionality on mobile, tablet, and desktop
* Ideal for inquiries, partnership requests, or customer support communication

### 5.5 Admin Panel

The admin panel of the Tech Startup web application provides a clean and functional interface designed exclusively for authorized admin users. Accessible only after authentication, this panel enables the management and monitoring of platform content and user interactions. On the left, a sidebar menu provides intuitive navigation with following options:

* **Dashboard:** The Dashboard page gives a quick overview of key metrics, including the number of user responses and published blogs.
* **Responses:** In the Responses section, admins can view a detailed table of contact messages, with information like name, email, subject, message, date, and actions such as marking as read or deleting entries.
* **Add New Blog:** The Add New Blog section includes a rich text editor for composing content, file input for uploading a blog image, and fields for title and short description—making content creation seamless.
* **View Blogs:** The **View Blogs** tab allows admins to see published blogs, update their status (e.g., Active/Inactive), and perform actions like edit or delete.



*Figure 5.6: Admin Panel*

## Chapter 6

## 6 Limitations and Future Scope

### Limitations

While the tech startup platform is designed to offer a seamless user experience, flexible service presentation, and administrative control, there are some limitations that may affect scalability and long-term performance. These include:

* **Content Management Flexibility:** Currently, the system relies on hardcoded service cards or manual entry via admin panels. The lack of dynamic CMS integration limits non-technical team members from editing or updating content in real time.
* **Scalability Constraints:** As the platform grows (e.g., hundreds of leads, services, or blog posts), MongoDB queries and serverless API routes may require optimization to prevent slow response times.
* **No Mobile App:** While the application is responsive, the absence of a native mobile app may hinder user engagement among mobile-first users who prefer dedicated apps.
* **Single Language Support:** The platform currently operates in a single language (likely English), which can limit accessibility for international users or localized startup markets.

### Future Scope

To enhance the platform's capabilities and market reach, several improvements and innovations can be implemented in future versions:

* **CMS Integration:** Integrating a headless CMS (e.g., Sanity, Strapi, or Contentful) can empower non-developers to manage and update site content, blogs, and service offerings easily.
* **AI-Powered Inquiry Insights:** Applying machine learning to analyze submitted inquiries can help categorize interest types, predict user intent, and provide smarter lead prioritization.
* **Mobile App Development:** Building native Android/iOS apps using React Native or Flutter would offer users a more tailored and engaging experience, especially for startups promoting tech-based solutions.
* **Multilingual Support:** Adding internationalization (i18n) support to enable users to access content in multiple languages would increase global reach and user inclusivity.
* **Admin Analytics Dashboard:** Introducing charts and statistics on the admin panel (e.g., inquiries per day, blog engagement, or traffic sources) would help stakeholders make informed decisions.
* **Search and Filter Features:** Implementing real-time search and filtering for blogs, services, or messages will improve content discoverability and user satisfaction.
* **AR & Interactive Features:** In future iterations, features like interactive demos, animated explainer sections, or augmented reality previews can elevate the tech-forward image of the platform.
* **Third-Party Integrations:** Adding support for third-party tools like Mailchimp (for newsletters), Slack (for team alerts), or HubSpot (for CRM) can streamline marketing and operations.

## Chapter 7

## 7 Conclusion

In conclusion, NextGen Tech, our innovative tech startup platform, emerges as a dynamic and forward-thinking solution in the digital ecosystem. Designed to empower emerging ventures with a strong and scalable online presence, NextGen Tech is committed to redefining how startups connect with audiences, present their services, and build lasting digital credibility. With its elegant interface, intuitive navigation, and powerful backend infrastructure, the platform lays a solid foundation for tech entrepreneurs to thrive in the competitive digital space.

Despite encountering common challenges such as limited content flexiblility, and user adoption, NextGen Tech addresses these with robust architectural decisions, secure authentication flows, and a roadmap centered on adaptability and user empowerment. Its modular architecture and admin-driven content controls allow seamless management of blogs, user interactions, and service visibility-ensuring the platform evolves in tandem with the needs of the business.

Looking ahead, NextGen Tech envisions expansive growth fueled by strategic enhancements. Future iterations will include advanced analytics dashboards, native mobile applications, AI-powered recommendations, and multilingual support to accommodate a broader user base. The integration of CMS platforms and real-time communication tools will further enhance operational agility and collaboration.

The platform also upholds a commitment to technological sustainability-adopting scalable cloud infrastructure, secure coding practices, and accessibility principles that prioritize performance, security, and inclusivity. By exploring partnerships with other SaaS providers, design firms, and service aggregators, NextGen Tech aims to continuously elevate its offering and deliver unmatched value to startups navigating the early stages of digital transformation.

In essence, NextGen Tech is not just a web platform-it is a launchpad for tomorrow’s innovations. Through continuous improvement, user-centric development, and a vision rooted in scalability and excellence, it is poised to become a trusted digital ally for startups, catalyzing growth, credibility, and impact across global markets.