

Project Report

On

“TECH TREKKER”



### SUBMITTED TO

**ROURKELA INSTITUTE OF MANAGEMENT STUDIES**

**(As a Partial fulfilment of the requirement for the award of degree) FOR**

## “MASTER IN COMPUTER APPLICATION “

**(2023-25) SUBMITTED BY**

SANTOSH BISWAL

**Registration Roll No:- 2305260021**

MCA 4th SEMESTER

### ROURKELA INSTITUTE OF MANAGEMENT STUDIES

***(Affiliated to Biju Pattnaik University of Technology, Odisha)***

Rourkela – 769015



**Rourkela Institute of Management Studies**

Rourkela

Department of Computer Science Rourkela Institute of Management Studies Chhend, Rourkela-15, Odisha

Phone:0661 2480482 Fax:91-0661-1480665

Mail: [rkl\_rimsgrol@sancharnet.in](mailto:rkl_rimsgrol@sancharnet.in) Visit: [www.rims-edu.com](http://www.rims-edu.com/)

#### CERTIFICATE OF EXAMINATION

This is to certify that this project report entitled "**TECH TREKKER**" submitted by **SANTOSH BISWAL** of 4th Semester, **Rourkela Institute of Management Studies, Rourkela**, is accepted as partial fulfillment of requirements for the degree in **Master in Computer Applications**, under **Biju Pattnaik University of Technology, Rourkela**, this has been verified by us and found be original up to our satisfaction.

External Examiner



**Rourkela Institute of Management Studies**

Rourkela

Department of Computer Science Rourkela Institute of Management Studies Chhend, Rourkela-15, Odisha

Phone:0661 2480482 Fax:91-0661-1480665

Mail: [rkl\_rimsgrol@sancharnet.in](mailto:rkl_rimsgrol@sancharnet.in) Visit: [www.rims-edu.com](http://www.rims-edu.com/)

#### CERTIFICATE OF EXAMINATION

This is to certify that this project report entitled "**TECH TREKKER**" submitted by **SANTOSH BISWAL** of 4th Semester, **Rourkela Institute of Management Studies, Rourkela**, is accepted as partial fulfillment of requirements for the degree in **Master in Computer Applications**, under **Biju Pattnaik University of Technology, Rourkela**, this has been verified by us and found be original up to our satisfaction.

Internal Examiner



**Rourkela Institute of Management Studies**

Rourkela

Department of Computer Science Rourkela Institute of Management Studies Chhend, Rourkela-15, Odisha

Phone:0661 2480482 Fax:91-0661-1480665

Mail: [rkl\_rimsgrol@sancharnet.in](mailto:rkl_rimsgrol@sancharnet.in) Visit: [www.rims-edu.com](http://www.rims-edu.com/)

#### CERTIFICATE

This is to certify that this project entitled **“TECH TREKKER”** has been and submitted by **SANTOSH BISWAL,**

M.C.A 2023-2025, **Rourkela Institute of Management Studies, Rourkela,** has been examined by us. She is found fit and approved for the award of **“Master in Computer Application “**Degree.

To the best my knowledge this work has not been submitted for the award of any other degree.

I wish all success in his life.

**DEAN ACADEMIC RIMS, ROURKELA**



Prof. Bibhudendu Panda Head of The Department, MCA

Rourkela Institute of Management Studies, Rourkela

#### CERTIFICATE

This is to certify that **SANTOSH BISWAL** student of **M.C.A, Rourkela Institute of Management Studies, Rourkela, Odisha** of Session 2023-2025 has completed the project successfully.

I wish all success in his life.

**(Prof. Bibhudendu Panda)**



#### DECLARATION

I am **SANTOSH BISWAL**, hereby declare that the project report Entitled “**TECH TREKKER**” is of my work. The above work I submitted to “**Biju Patnaik University of Technology, Rourkela”** for the award of **“Master in Computer Applications**” Degree.

To the best of my knowledge, this work has not been submitted or published anywhere for the award of any degree.

**SANTOSH BISWAL**



#### ACKNOWLGEMENT

I am deeply indebted to **Rourkela Institute of Management Studies, Chhend, Rourkela,** for providing me an opportunity to undertake a project work entitled **“TECH TEKKER”.**

I am grateful to my project guide **Prof. Bibhudendu Panda** without his guidance it would not have been possible on my part to complete the project. I acknowledge the help and co-operation received from all my team members in making this project.

I consider myself fortunate that I have successfully completed this project; I acknowledge my sincere gratitude to all those works and ideas that had helped me in writing this project.

#### SANTOSH BISWAL

**University Roll No: 2305260021**

**MCA (2023-2025)**

**Rourkela Institute of Management Studies, Rourkela.**

Abstract

In the digital era, managing and publishing technology-related content efficiently is crucial for bloggers, enterprises, and tech enthusiasts. A \*Tech Trekker\* is a web-based platform designed to streamline content creation, organization, and distribution. It offers an intuitive and user-friendly interface that enables authors, editors, and administrators to collaborate seamlessly in producing high-quality blog posts.

The system supports a robust content management framework, allowing users to draft, edit, schedule, and publish articles with ease. Advanced features such as \*content categorization, tagging, SEO optimization, and multimedia embedding\* enhance the reach and engagement of the blog. Additionally, the system incorporates \*role-based access control\*, ensuring that different user levels (admin, editor, contributor, and reader) have appropriate permissions to manage content efficiently.

A key aspect of this platform is \*AI-driven content recommendations, which help users discover trending topics, suggest keyword improvements, and optimize content for better search engine visibility. Moreover, \*\*social media integration\* allows seamless sharing across multiple platforms, increasing audience engagement. The inclusion of \*analytics and reporting tools\* provides insights into reader behavior, helping bloggers refine their content strategy based on real-time data.

To further enhance the blogging experience, the system includes features such as \*automated post scheduling, version control, comment moderation, and community engagement tools. The implementation of a \*\*secure and scalable backend\* ensures smooth performance, data integrity, and protection against cyber threats.

By integrating modern technologies like \*cloud storage, AI-powered writing assistance, and responsive design\*, the Tech Blog Management System serves as an efficient solution for individuals and organizations looking to establish a compelling online presence in the tech industry. It simplifies content workflows, boosts productivity, and ensures that high-quality technology-related content reaches the right audience effectively

TABLE OF CONTENTS

1. TITLE PAGE 1
2. CERTIFICATE OF EXAMINATION 2
3. CERTIFICATE OF EXAMINATION 3
4. CERTIFICATE 4
5. CERTIFICATE 5
6. DECLARATION 6
7. ACKNOWLEDGEMENT 7
8. ABSTRACT 8
9. TABLE OF CONTENTS 9
10. CHAPTER ONE:…………………………………………………..….11
    * INTRODUCTION
    * Background
    * Significance
    * Purpose
    * Motivation
11. CHAPTER TWO 13
    * LITERATURE REVIEW
    * Challenges
12. CHAPTER THREE 14
    * SYSTEM REQUIREMENTS
    * Hardware Requirements
    * Software Requirements
13. CHAPTER FOUR 15
    * TOOLS AND TECHNOLOGIES
    * HTML
    * CSS
    * SPRING BOOT
    * PHP
    * MY SQL
    * JAVA
14. CHAPTER FIVE 16
    * [SYSTEM DESIGN](#_TOC_250002)
    * [Use Case Diagram](#_TOC_250001)
    * [ER Diagram](#_TOC_250000)
15. CHAPTER SIX 18
    * SYSTEM ANALYSIS
    * Proposed System
    * Web Application
    * User Requirement
    * Admin
    * Inventory Control
16. CHAPTER SEVEN 20
    * IMPLEMENTATION
    * Primary Implementation Exploring Database
17. CHAPTER EIGHT 22
    * Coding Section
18. CHAPTER NINE 71
    * TESTING
    * Introduction
    * Unit Testing
    * Testing Scenarios
19. CHAPTER TEN 72
    * CONCLUSION
    * Conclusion
    * Project Limitations
20. CHAPTER ELEVEN 73
    * FUTURE ENHANCEMENTS
    * REFERENCES

**CHAPTER ONE**

INTRODUCTION

##### Background

According to today's digital landscape blog, the digital landscape is the ultimate source of information in today's digital landscape, providing readers with the most current trends, advancements, and discoveries in the tech industry. Nevertheless, handling high-tech blogs can present challenges as it requires effective content creation, organization, and distribution. A tech blog management system is designed to address the challenges faced by authors, editors, and administrators by offering an optimized platform that streamlines the management of blog posts.

This system incorporates user-friendly content management tools to simplify the process of designing, editing, planning, and publishing content. Contains essential elements like search engine optimization, multimedia embedding, and social media integration. Furthermore, system role-based access control incorporates different authentication levels for participants, editors, and administrators to guarantee seamless collaboration.

By incorporating artificial intelligence control recommendations, analysis, and automated planning, the technical blog management system streamlines content strategies, enhances audience engagement, and simplifies workflows. Safety and scalability are key considerations, guaranteeing data integrity, implementing security measures, and safeguarding against cyber threats.

This introductory provides the necessary background knowledge for a thorough understanding of how a technical blog management system can transform content creation and publishing, making it an essential tool for individual bloggers, tech startups, and businesses aiming to establish a strong online presence.

### Purpose

1. \*Efficient Content Management\* – Provides a structured platform for creating, editing, organizing, and publishing technology-related articles with ease, ensuring a seamless content workflow.

2. \*SEO and Audience Engagement\* – Enhances content visibility through built-in \*SEO tools, keyword optimization, and \*\*social media integration\*, helping blogs reach a wider audience.

3. \*Collaboration and Role-Based Access\* – Supports multiple user roles (admin, editor, contributor) with controlled permissions, enabling \*smooth teamwork\* and \*efficient content moderation\*.

4. \*Analytics and Performance Tracking\* – Provides \*data-driven insights\* into reader behavior, post engagement, and traffic patterns to help bloggers optimize their content strategy.

5. \*Automation and Security\* – Offers \*scheduled publishing, version control, and backup features\*, ensuring consistency and protection against data loss or cyber threats.

##### Motivation

##### In the constantly evolving digital landscape, technology blogs serve as a vital source of current information for readers, offering insights into the latest trends, innovations, and advancements. Nevertheless, effectively managing a tech blog can be challenging due to the requirement for content categorization, search engine optimization, audience engagement, and collaboration among writers, editors, and administrators. A tech blog management system is created to simplify these processes by offering a centralized platform for content creation, editing, scheduling, and publishing. With features such as automated SEO suggestions, analytics, role-based access control, and social media integration, bloggers can improve their content's visibility and attract a larger audience. Furthermore, built-in security features, version control, and backup mechanisms guarantee data integrity and safeguard against the possibility of data loss. By implementing a tech blog management system, bloggers, tech companies, and media organizations can streamline their content management processes, gain valuable insights from data, and effectively scale their content strategies, resulting in exceptional and captivating content for their audience.

**CHAPTER TWO**

#### LITERATURE REVIEW

Challenges

Managing a \*Tech Blog Management System\* comes with several challenges that can impact content quality, efficiency, and audience engagement. One of the primary difficulties is \*content consistency and quality control, as maintaining a steady flow of high-quality, well-researched articles requires careful planning and coordination among writers, editors, and administrators. Additionally, \*\*SEO optimization and audience reach\* pose challenges, as blogs must constantly adapt to evolving search engine algorithms and competition for visibility. \*Security concerns, such as data breaches, plagiarism, and unauthorized access, also require robust protective measures, including encryption and role-based access control. Another challenge is \*\*integrating multimedia elements\* like images, videos, and infographics while ensuring fast page load speeds and a seamless user experience. Furthermore, \*scalability and performance optimization\* become crucial as the blog expands, requiring efficient database management and server optimization. Lastly, \*engaging and retaining readers\* through interactive features, such as comments, newsletters, and social media integration, demands continuous innovation and audience analysis. Overcoming these challenges requires a \*well-structured, automated, and secure Tech Blog Management System\* that enhances productivity while maintaining content quality and security.

Database is a group of connected files that are structured to be

accessible to end users simultaneously. It gathers and organizes data

so that it can be displayed in a single location. They are created and

carried by a software known as Database Management System,

which deals with the way of information maintained, structured and

accessed. Structured Query Language is implemented by other

applications for processing data in a relational database

management systems.MY SQL and PHP are used in this system for

remote server services.

# CHAPTER THREE

## SYSTEM REQUIREMENTS

**Hardware Requirements**

* Display: 1920x1080 resolution Monitor.
* Processor: Intel Core i5 or higher.
* Network: Internet connectivity or Wi-Fi.
* Storage: At least 256GB SSD.
* RAM: 8GB or higher.

**Software Requirements**

* Visual Studio Code.
* Git
* My SQL
* XAMPP

# CHAPTER FOUR

### TOOLS AND TECHNOLOGIES

*HTML*

Hyper Text Mark-up Language is used for developing front-end Graphical user interface. It is standard language used for web pages.

*CSS*

Cascading style sheets are a style sheet language used to show a document or content written in mark-up language.

*SPRING BOOT*

Spring Boot is a Java framework that makes it easier to create and run Java applications. It simplifies the configuration and setup process, allowing developers to focus more on writing code for their applications.

*PHP*

The system was created with the LA ravel PHP framework. The software will be developed using PHP one of the most widely used and reliable technologies for developing custom software solutions.

*MYSQL*

For database MYSQL is used. This technology will ensure that the software is scalable, reliable, and secure.

*JAVA*

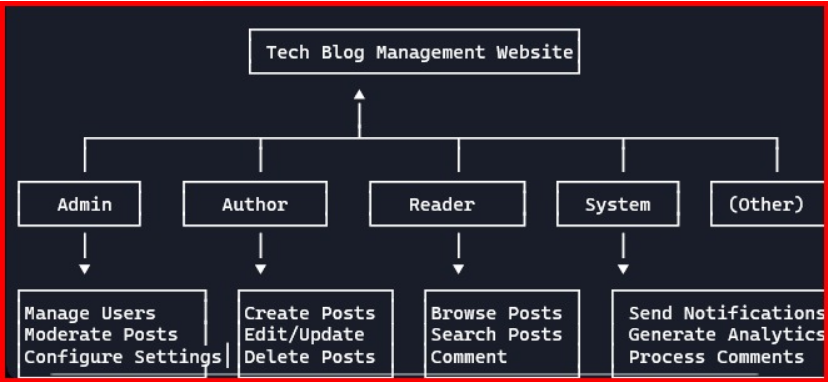
Java is a class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible. It is intended to let application

developers Write Once and Run Anywhere (WORA), meaning that compiled Java code can run on all platforms that support Java without the need for recompilation.

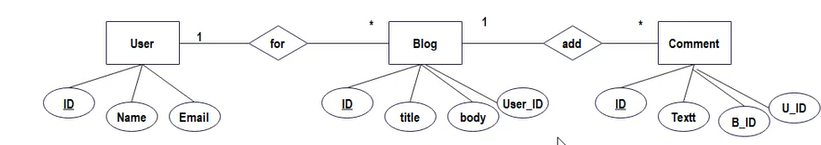
# CHAPTER FIVE

### SYSTEM DESIGN

**Use Case Diagram**

****

## ER Diagram

****

# CHAPTER SIX

### SYSTEM ANALYSIS

### A \*Tech Blog Management System\* aims to address the challenges associated with managing technology-related content, including \*content organization, user engagement, SEO optimization, and security. The existing methods of managing blogs often involve \*\*manual content handling, lack of automation, and poor analytics\*, leading to inefficiencies and lower audience engagement. Common issues faced in traditional blogging platforms include:

### - \*Inefficient Content Management:\* Difficulty in organizing, editing, and scheduling blog posts.

### - \*SEO and Visibility Challenges:\* Lack of built-in tools for search engine optimization, affecting traffic.

### - \*Collaboration Issues:\* Difficulty in managing multiple contributors and role-based access control.

### - \*Security and Data Loss Risks:\* Lack of proper security measures, making content vulnerable to cyber threats.

### - \*Limited Analytics and Monetization:\* Inability to track audience engagement and revenue generation efficiently.

### Proposed System\*

### To overcome these challenges, the proposed \*Tech Blog Management System\* will be a \*web-based platform\* that offers a \*structured, automated, and secure\* content management solution. The system will include the following key features:

### 1. \*User Role Management\*

### - Multiple user roles (Admin, Editor, Author, Reader) with controlled access.

### - Secure authentication and authorization mechanisms.

### 2. \*Content Management & Workflow Automation\*

### - Rich-text editor for blog creation with image/video embedding.

### - Drafting, scheduling, and automated publishing of posts.

### - Version control for tracking content updates.

### 3. \*SEO & Social Media Integration\*

### - SEO tools for keyword analysis and meta-tag optimization.

### - Social media sharing options for wider content reach.

### 4. \*Analytics & Performance Tracking\*

### - Real-time insights into blog views, engagement, and traffic sources.

### - Audience behavior analysis for content strategy improvements.

### 5. \*Security & Data Protection\*

### - Role-based access control (RBAC) for secure data handling.

### - Automated data backups and protection against cyber threats.

### 6. \*Monetization & Advertisement Support\*

### - Integration with Google AdSense and affiliate marketing tools.

### - Sponsored content management for additional revenue.

##### Web Application

A Tech Blog Web Application is a platform designed to efficiently manage and publish technology-related content. It features user authentication, role-based access, a rich-text editor, SEO tools, social media integration, and analytics to enhance engagement and visibility. Users can write, edit, schedule, and monetize their blogs with tools like Google AdSense and affiliate marketing. Built with React.js for the frontend and Node.js with Express.js for the backend, it ensures scalability, security, and performance. Cloud services like AWS or Firebase handle hosting and storage, while AI-driven content recommendations and PWA support enhance the user experience

**\*Scope of the Tech Blog Page\***

The \*Tech Blog Page\* serves as a comprehensive platform for creating, managing, and publishing technology-related content. Its primary focus is to \*deliver informative, engaging, and SEO-optimized content\* while ensuring a seamless user experience. The scope of this system includes:

1. \*Content Management\* – Authors can create, edit, schedule, and publish blogs with text, images, and videos.

2. \*User Roles & Access Control\* – Different user roles (Admin, Editor, Author, Reader) with controlled permissions.

3. \*SEO & Social Media Integration\* – Built-in SEO optimization tools, meta-tag management, and auto-sharing to social platforms.

4. \*Engagement Features\* – Interactive comment section, like/share buttons, and newsletter subscriptions for user interaction.

5. \*Analytics & Performance Tracking\* – Real-time insights into blog views, engagement metrics, and reader behavior.

6. \*Security & Data Protection\* – Secure authentication, encrypted storage, and automated data backups.

7. \*Monetization Support\* – Integration with \*Google AdSense, affiliate marketing, and sponsored content management\*.

8. \*Scalability & Future Enhancements\* – AI-powered content recommendations, voice-to-text blogging, and Progressive Web App (PWA) support.

**User Perquisites:**

1: Users should have a basic understanding of computer and internet skills, including web browsing, text editing, and navigating online platforms. 2: To access the authoring and management features, a registered account (via email, Google, or social login) is necessary. 3: Authors and editors involved in blogging and content creation should possess a fundamental understanding of writing, formatting, and structuring blog posts. 4: Having a basic understanding of SEO practices, including keyword usage and content sharing, will enhance the visibility of your blog. 5: Image and media management – users should be aware of how to upload and handle images, videos, and other multimedia components to enhance the quality of their content.

**Admin:**

The admin is the top authority in the tech blog management system, responsible for managing user accounts, moderating content, ensuring security, and maintaining the system's functionality. Administrators ensure smooth platform operation, enforce regulations, and manage monetization and analytics. Key responsibilities of an admin: User management involves creating, changing, and deleting user accounts, assigning specific roles (admin, editor, author, reader), and regulating access permissions. Content moderation entails the process of reviewing, approving, editing, or deleting blog posts, comments, and media to ensure high-quality content and compliance with established guidelines. 3: security & access control – monitor user activities, enforce authentication protocols, block spam, and safeguard against cyber threats. SEO and performance optimization – ensure blog posts follow seo best practices, manage metadata, and enhance site speed. 5: analytics and reports – track website traffic, user engagement, and content performance using dashboards and analytics tools. 6: monetization management – oversee the management of Google Ads, affiliate marketing, sponsored content, and ad placements to generate revenue. 7: system maintenance – perform regular backups, software updates, and bug fixes to guarantee the platform's security and efficiency. The admin role plays a vital role in maintaining content quality, fostering user engagement, ensuring security, and driving business growth within the tech blog management system

**Inventory Control:**

1. \*Digital Asset Management\* – Efficient handling of \*images, videos, infographics, and documents\*, ensuring optimized storage and accessibility.

2. \*Subscription & Premium Content Control\* – Tracking and managing \*exclusive content access, user subscriptions, and paid memberships\*.

3. \*Advertisement Slot Management\* – Controlling the \*availability, allocation, and monetization\* of ad spaces on the blog.

4. \*E-commerce Integration\* – If the blog sells tech-related merchandise (e.g., gadgets, e-books, or courses), an \*inventory system for stock tracking and order management\* is required.

5. \*Cloud Storage & Resource Allocation\* – Monitoring \*server space, bandwidth, and resource usage\* to optimize performance and scalability.

# CHAPTER SEVEN

#### IMPLEMENTATION

Primary Implementation Admin:

The administrator is accountable for managing the system. This entails establishing and managing user accounts, operating the database, and ensuring the system's efficiency.

Reports User:

The \*Report User\* feature allows users to report inappropriate behavior, spam, or violations of community guidelines, ensuring a safe and respectful environment. This feature helps admins take necessary actions such as warnings, content removal, or user suspension.

\*Key Aspects of the Report User Feature:\*

1. \*Report Categories\* – Users can report others for reasons like \*spam, harassment, offensive content, plagiarism, or fake accounts\*.

2. \*Report Submission\* – A simple \*"Report" button\* is available on user profiles, comments, or posts, allowing users to provide details.

3. \*Admin Dashboard for Reports\* – Admins receive \*detailed reports\*, including the reported content, reason, and reporter details, for review.

4. \*Automated Warnings & Actions\* – Based on \*severity\*, admins can issue warnings, restrict user access, or suspend accounts.

5. \*User Appeal System\* – Reported users can appeal bans or restrictions if they believe the report was unfair.

This feature ensures that the \*Tech Blog Management System\* remains a \*safe, high-quality, and engaging platform\* for all users.

## Exploring Database

Implementation of the PHP and MySQL-based Sales and Stock Management System for a small retail has the following steps:

* **Hosting:** A cloud-based hosting platform should be selected. This hosting site should have enough storage space, bandwidth to deal with large volumes of data is required.
* **Database Design**: For storing data regarding transactions and supplies. It is necessary to develop well organized database. The database must contain fields for transactions, products and inventory and the fields need to be interconnected in a proper way.
* **User Interface Design**: The entire system should have a user -friendly dashboard which allows customers to enter and display both sales and inventory data. The interface should be customizable and function properly on a number of platforms like laptops and mobile phones.
* **PHP Programming:** PHP scripts need to be developed to provide the system all the features like entering information, searching the data and generating reports. The code should be developed to be secure as well as expanding and must deal with error management and store them appropriately.
* **Integration with Point-of-Sale (POS) Systems**: PHP and My-SQL systems should be interconnected in order to get real-time data.
* **Testing and Deployment**: The system needs to be tested properly in order to ensure that it functions as planned also ensuring that the date precise and secured. When the testing is completed the system can be uploaded on the hosting site and made accessible to all the clients.
* **Maintenance and Upgrades**: Regular maintenance and upgrades must be performed on the system to ensure that it continues to function effectively and provide the necessary level of service. This may involve fixing bugs, adding new features, and upgrading the underlying software and hardware components.

CHAPTER EIGHT

## CODING SECTION:

<!DOCTYPE html>

<html lang="en">

  <head data-th-replace="~{fragment/basic-fragment :: head}">

    <!-- <link rel="stylesheet" href="home.css" /> -->

  </head>

  <body class="basic-layout">

    <link rel="stylesheet" href="/css/blog-details.css" />

    <header data-th-replace="~{fragment/basic-fragment :: header}"></header>

    <main>

      <section class="blog-section">

        <article data-th-object="${details}">

          <h4 data-th-text="\*{blogTitle}"></h4>

          <img data-th-src="|/blog/image/\*{blogId}|" alt="......." />

          <p data-th-text="\*{blogDescription}"></p>

        </article>

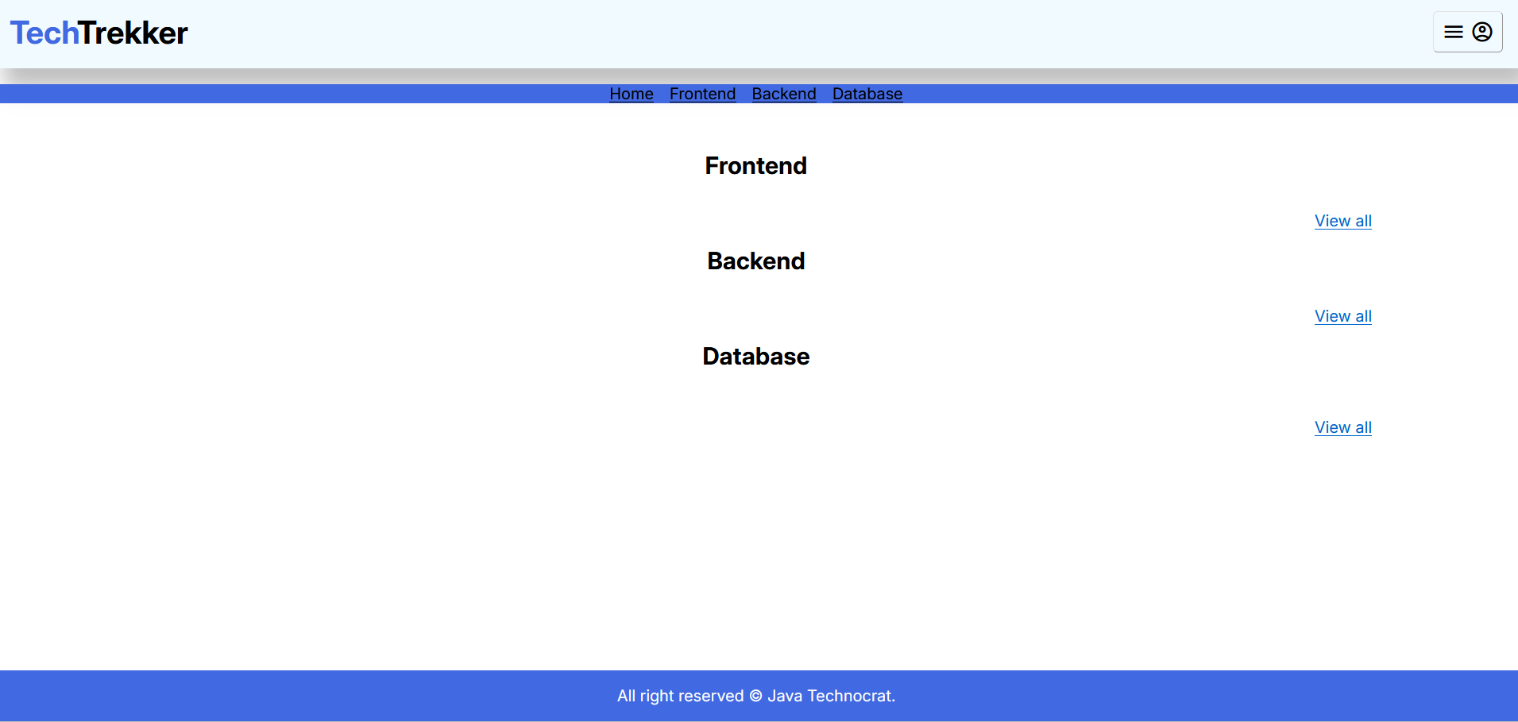
      </section>

    </main>

    <!-- <footer data-th-replace="/fragment/basic-fragment :: footer"></footer> -->

  </body>

</html>



<!DOCTYPE html>

<html lang="en">

  <head data-th-replace="~{fragment/basic-fragment :: head}">

    <!-- <link rel="stylesheet" href="home.css" /> -->

  </head>

  <body>

    <link rel="stylesheet" href="/css/create-blog.css" />

    <link rel="stylesheet" href="/css/form.css" />

    <header data-th-replace="~{fragment/basic-fragment :: header}"></header>

    <section>

      <h2 class="text-center">Create New Blog</h2>

      <hr>

      <form

        action="/writer/create-blog"

        method="post"

        enctype="multipart/form-data"

      >

            <div class="form-group">

              <label for="title">Title</label>

              <input type="text" id="title" name="blogTitle" required class="form-control"/>

            </div>

            <div class="form-group">

              <select name="category" id="category" class="form-control">

                <option value="FRONTEND">Frontend</option>

                <option value="BACKEND">Backend</option>

                <option value="DATABASE">DataBase</option>

              </select>

            </div>

            <div class="form-group">

              <label for="description">Description</label>

              <textarea

                id="description"

                name="blogDescription"

                required

                rows="5"

                class="form-control"

              ></textarea>

            </div>

            <div class="form-group">

              <div class="banner-preview">

                <label for="banner">Select a Banner</label>

              </div>

              <input type="file" id="banner" name="banner" />

            </div>

        <input type="hidden" name="\_csrf" data-th-value="${\_csrf.token}" />

        <div class="text-center">

          <input type="submit" />

        </div>

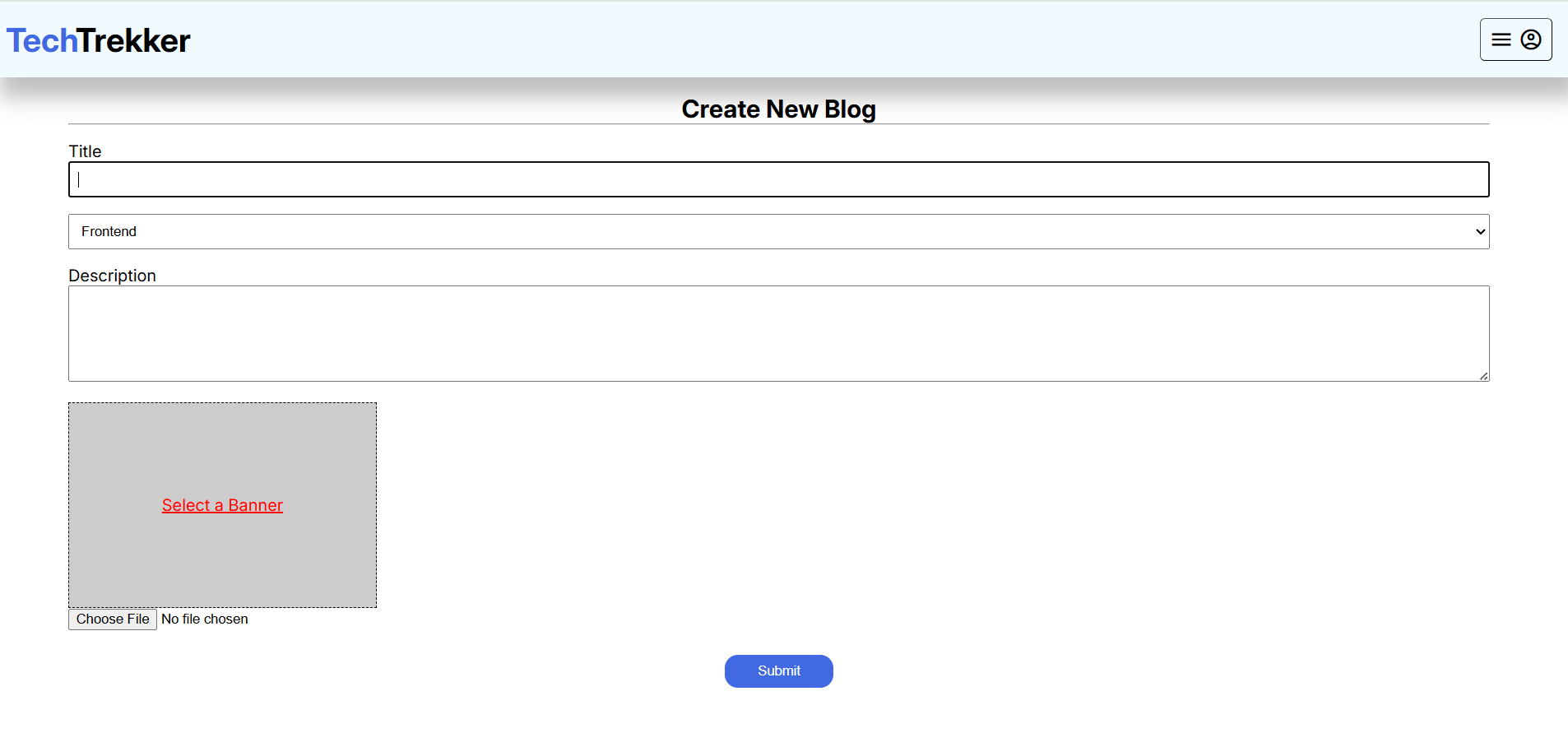
      </form>

    </section>

    <script src="/js/create-blog.js"></script>

  </body>

</html>



<!DOCTYPE html>

<html lang="en">

  <head data-th-replace="~{fragment/basic-fragment :: head}">

    <!-- <link rel="stylesheet" href="home.css" /> -->

  </head>

  <body class="basic-layout">

    <link rel="stylesheet" href="/css/home/recent-section.css" />

    <link rel="stylesheet" href="/css/home/home.css" />

    <link rel="stylesheet" href="/css/home/frontend-section.css" />

    <link rel="stylesheet" href="/css/home/backend-section.css" />

    <link rel="stylesheet" href="/css/home/database-section.css" />

    <header data-th-replace="~{fragment/basic-fragment :: header}"></header>

    <nav data-th-replace="~{fragment/basic-fragment :: nav}"></nav>

    <main data-th-object="${response}">

      <!-- recent blogs -->

      <section class="recent-blog-section">

        <article

          data-th-each="blog : \*{recentBlogInfos}"

          data-th-object="${blog}"

        >

          <div class="image-container">

            <img data-th-src="|/blog/image/\*{blogId}|" alt="......" />

          </div>

          <div class="details-container">

            <p class="line-clamp-2" data-th-text="\*{blogTitle}"></p>

            <a data-th-href="|/tech-trekker/blog-details?blogId=\*{blogId}|"

              >Read more</a

            >

          </div>

        </article>

      </section>

      <!-- Frontend Blogs -->

      <section id="frontend-section">

        <h2 class="text-center">Frontend</h2>

        <div class="frontend-blogs">

          <article

            data-th-each="blog : \*{frontendBlogInfos}"

            data-th-object="${blog}"

          >

            <div class="image-container">

              <img data-th-src="|/blog/image/\*{blogId}|" alt="......" />

            </div>

            <div class="title-container">

              <p data-th-text="\*{blogTitle}" class="line-clamp-2"></p>

              <a data-th-href="|/tech-trekker/blog-details?blogId=\*{blogId}|"

                >Read more</a

              >

            </div>

          </article>

        </div>

        <div class="link-container text-right">

          <a data-th-href="|/tech-trekker/view-all?category=FRONTEND|">View all</a>

        </div>

      </section>

      <!-- Backend Blogs -->

      <section id="backend-section">

        <h2 class="text-center">Backend</h2>

        <div class="backend-blogs">

          <article

            data-th-each="blog : \*{backendBlogInfos}"

            data-th-object="${blog}"

          >

            <div class="image-container">

              <img data-th-src="|/blog/image/\*{blogId}|" alt="......" />

            </div>

            <div class="title-container">

              <p data-th-text="\*{blogTitle}" class="line-clamp-2"></p>

              <a data-th-href="|?blogId=\*{blogId}|">Read more</a>

            </div>

          </article>

        </div>

        <div class="link-container text-right">

          <a data-th-href="|/tech-trekker/view-all?category=BACKEND|">View all</a>

        </div>

      </section>

      <!-- Database Blogs -->

      <section id="database-section">

        <h2 class="text-center">Database</h2>

        <div class="database-blogs">

          <article

            data-th-each="blog : \*{databaseBlogInfos}"

            data-th-object="${blog}"

          >

            <div class="image-container">

              <img data-th-src="|/blog/image/\*{blogId}|" alt="......" />

            </div>

            <div class="title-container">

              <p data-th-text="\*{blogTitle}" class="line-clamp-2"></p>

              <a data-th-href="|/tech-trekker/blog-details?blogId=\*{blogId}|">Read more</a>

            </div>

          </article>

        </div>

        <div class="link-container text-right mb-1">

          <a data-th-href="|/tech-trekker/view-all?category=DATABASE|" |>View all</a>

        </div>

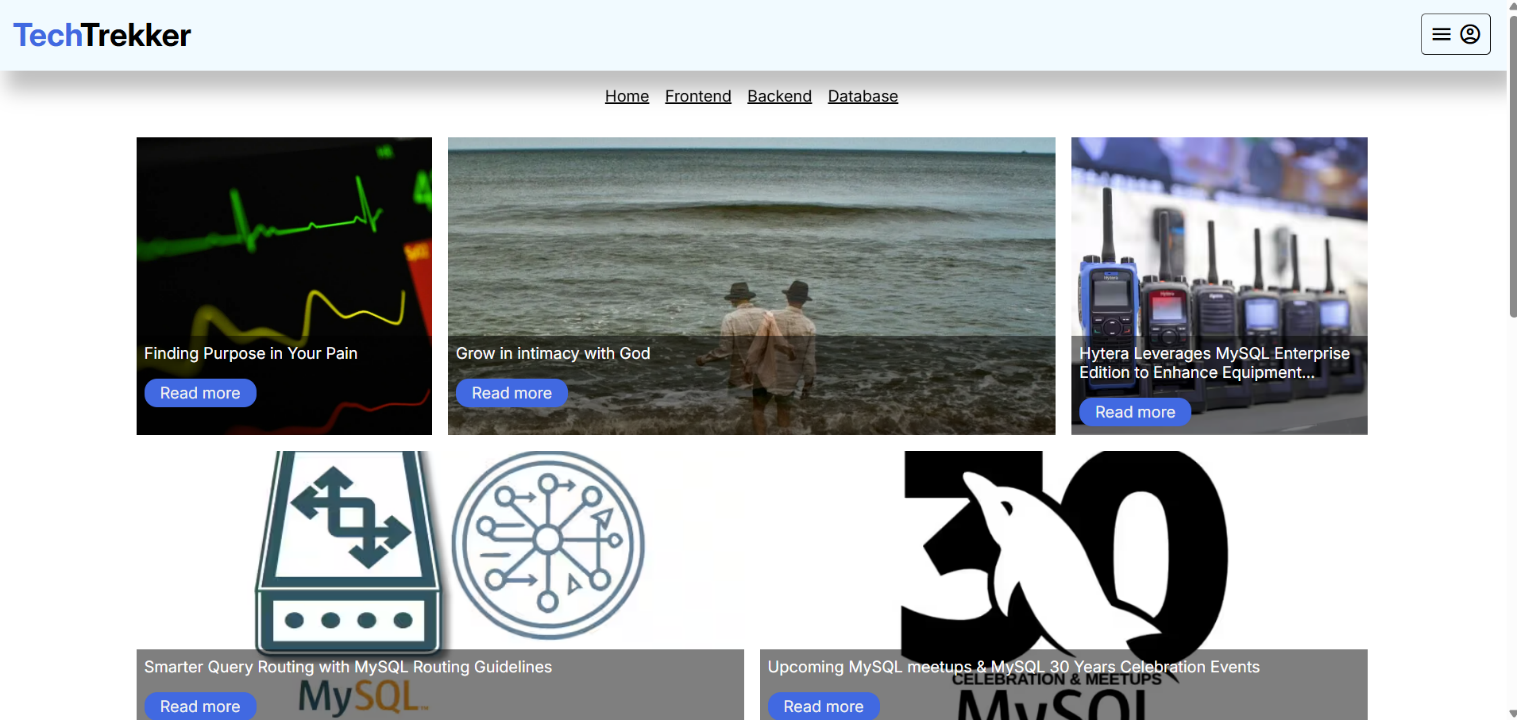
      </section>

    </main>

    <footer data-th-replace="/fragment/basic-fragment :: footer"></footer>

  </body>

</html>



<!DOCTYPE html>

<html lang="en">

  <head data-th-replace="~{fragment/basic-fragment :: head}"></head>

  <body class="basic-layout">

    <link rel="stylesheet" href="/css/form.css" />

    <header data-th-replace="~{fragment/basic-fragment :: header}"></header>

    <section class="form-section">

      <h2>Login</h2>

      <hr />

      <label for="" data-th-if="${param.error}!=null"

        class="error">Invalid Credential!!</label

      >

      <form action="/login" method="post" enctype="multipart/form-data">

        <div class="form-group">

          <label for="email">Email</label>

          <input

            type="email"

            name="username"

            id="email"

            required

            class="form-control"

          />

        </div>

        <div class="form-group">

          <label for="password">Password</label>

          <input

            type="password"

            name="password"

            id="password"

            required

            class="form-control"

          />

        </div>

        <div class="text-center">

          <input type="submit" name="submit" id="submit" value="Login" />

        </div>

        <div class="link text-right">

          <a href="/tech-trekker/signup">Sign up here</a>

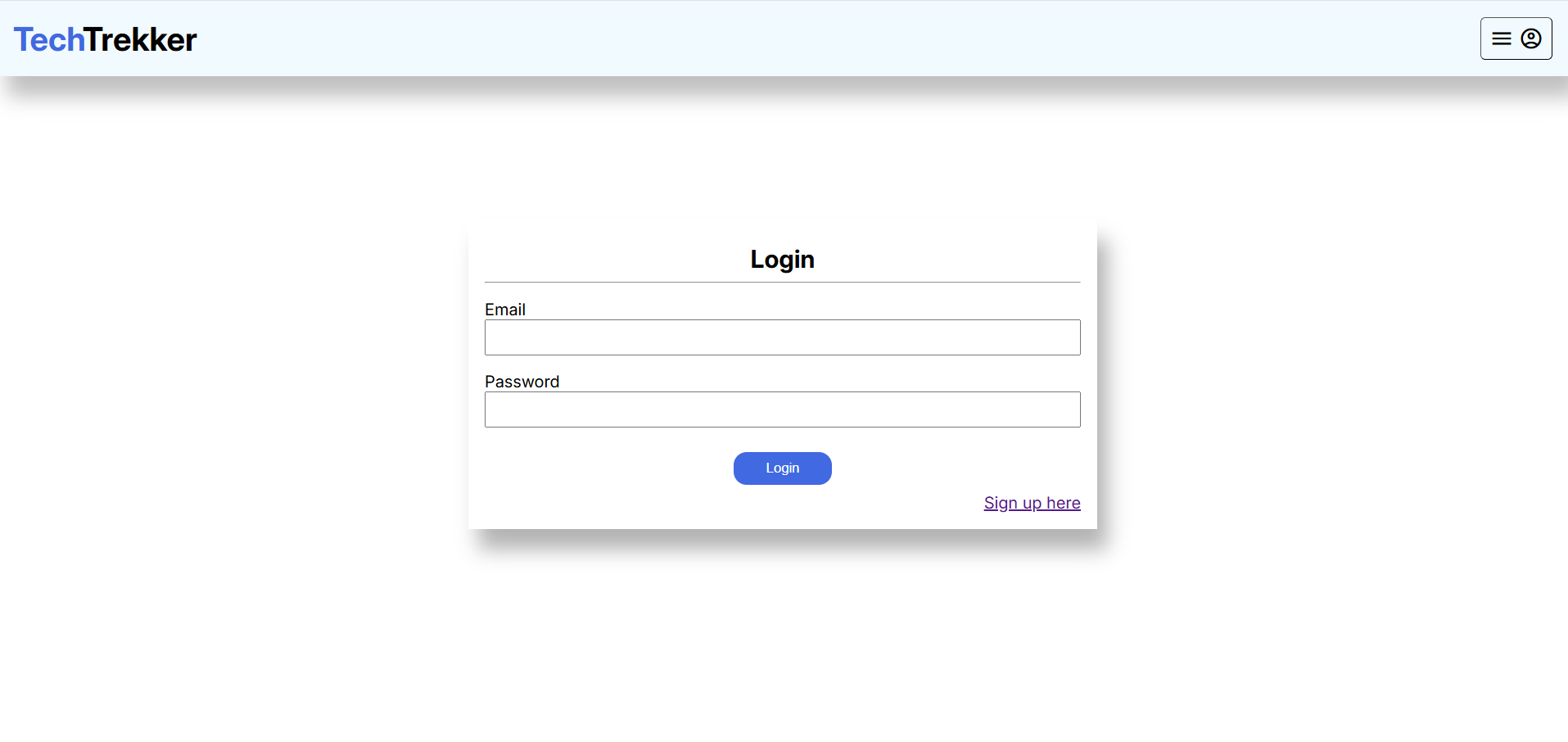
        </div>

      </form>

    </section>

  </body>

</html>



<!DOCTYPE html>

<html lang="en">

  <head data-th-replace="~{fragment/basic-fragment :: head}"></head>

  <body class="basic-layout">

    <link rel="stylesheet" href="/css/form.css" />

    <header data-th-replace="~{fragment/basic-fragment :: header}"></header>

    <section class="form-section">

      <h2>Signup</h2>

      <hr />

      <form

        action="/tech-trekker/signup"

        method="post"

        enctype="multipart/form-data"

      >

        <div class="form-group">

          <label for="name">Name</label>

          <input

            type="text"

            name="writerName"

            id="name"

            required

            class="form-control"

          />

        </div>

        <div class="form-group">

          <label for="email">Email</label>

          <input

            type="email"

            name="email"

            id="email"

            required

            class="form-control"

          />

        </div>

        <div class="form-group">

          <label for="password">Password</label>

          <input

            type="password"

            name="password"

            id="password"

            required

            class="form-control"

          />

        </div>

        <div class="form-group">

          <label for="profile-pic">Picture</label>

          <input

            type="file"

            name="profilePicture"

            id="profile-pic"

          />

        </div>

        <div class="form-group">

          <label for="profession">Profession</label>

          <input

            type="text"

            name="writerProfession"

            id="profession"

            required

            class="form-control"

          />

        </div>

        <div class="submit-btn-container text-center">

          <input type="submit" name="submit" id="submit" value="Signup"/>

        </div>

        <div class="link text-right">

          <a href="/login-page">Sign in here</a>

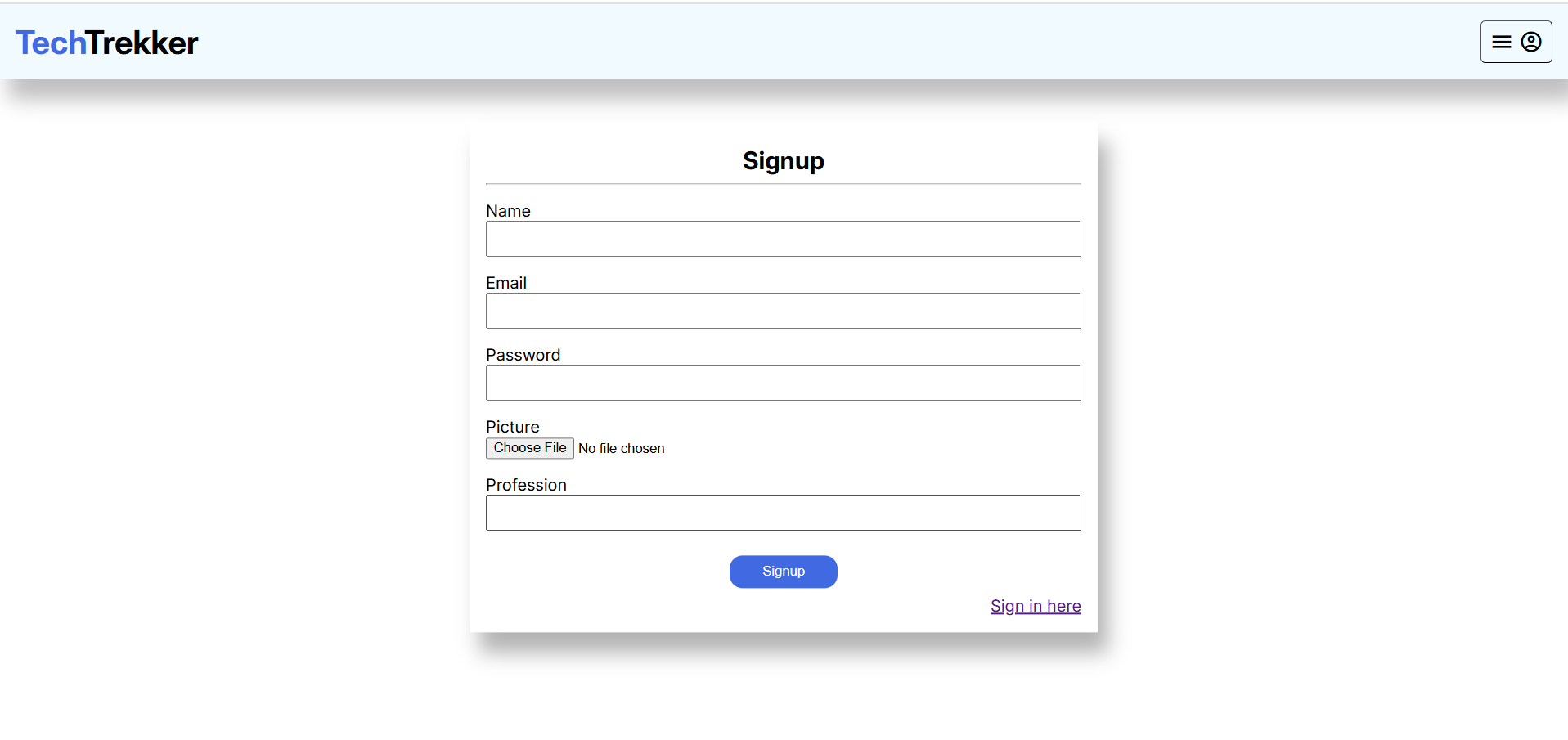
        </div>

      </form>

    </section>

  </body>

</html>



<!DOCTYPE html>

<html lang="en">

  <head data-th-replace="~{fragment/basic-fragment :: head}">

    <!-- <link rel="stylesheet" href="home.css" /> -->

  </head>

  <body class="basic-layout">

    <link rel="stylesheet" href="/css/view-all.css" />

    <script src="/js/view-all.js" defer></script>

    <header data-th-replace="~{fragment/basic-fragment :: header}"></header>

    <nav data-th-replace="~{fragment/basic-fragment :: nav}"></nav>

    <main>

      <section class="search-section">

        <form action="/tech-trekker/view-all-search" method="post">

          <input

            type="search"

            placeholder="Search any title..."

            name="searchTerm"

            data-th-value="${response.searchTerm}"

          />

          <input

            type="hidden"

            name="category"

            data-th-value="${param.category}"

          />

          <button type="submit">

            <span class="material-symbols-outlined"> search </span>

            Search

          </button>

        </form>

      </section>

      <section class="blog-section">

        <div class="all-blogs">

          <article

            data-th-each="blog : \*{response.blogs}"

            data-th-object="${blog}"

          >

            <div class="image-container">

              <img data-th-src="|/blog/image/\*{blogId}|" alt="......" />

            </div>

            <div class="details-container">

              <h4 data-th-text="\*{blogTitle}"></h4>

              <p data-th-text="\*{blogDescription}"></p>

              <a data-th-href="|/tech-trekker/blog-details?blogId=\*{blogId}|"

                >Read more</a

              >

            </div>

          </article>

        </div>

        <div class="pagignation">

          <button

            data-th-disable="${response.currentPage} == 1"

            data-th-onclick="|onPageChange(${response.currentPage}, '-')|"

          >

            <span class="material-symbols-outlined"> arrow\_back\_ios </span>

          </button>

          <p

            data-th-text="|${response.currentPage} - ${response.totalPage} pages|"

          ></p>

          <button

            data-th-disable="${response.currentPage} == ${response.totalPage}"

            data-th-onclick="|onPageChange(${response.currentPage}, '+')|"

          >

            <span class="material-symbols-outlined"> arrow\_forward\_ios </span>

          </button>

        </div>

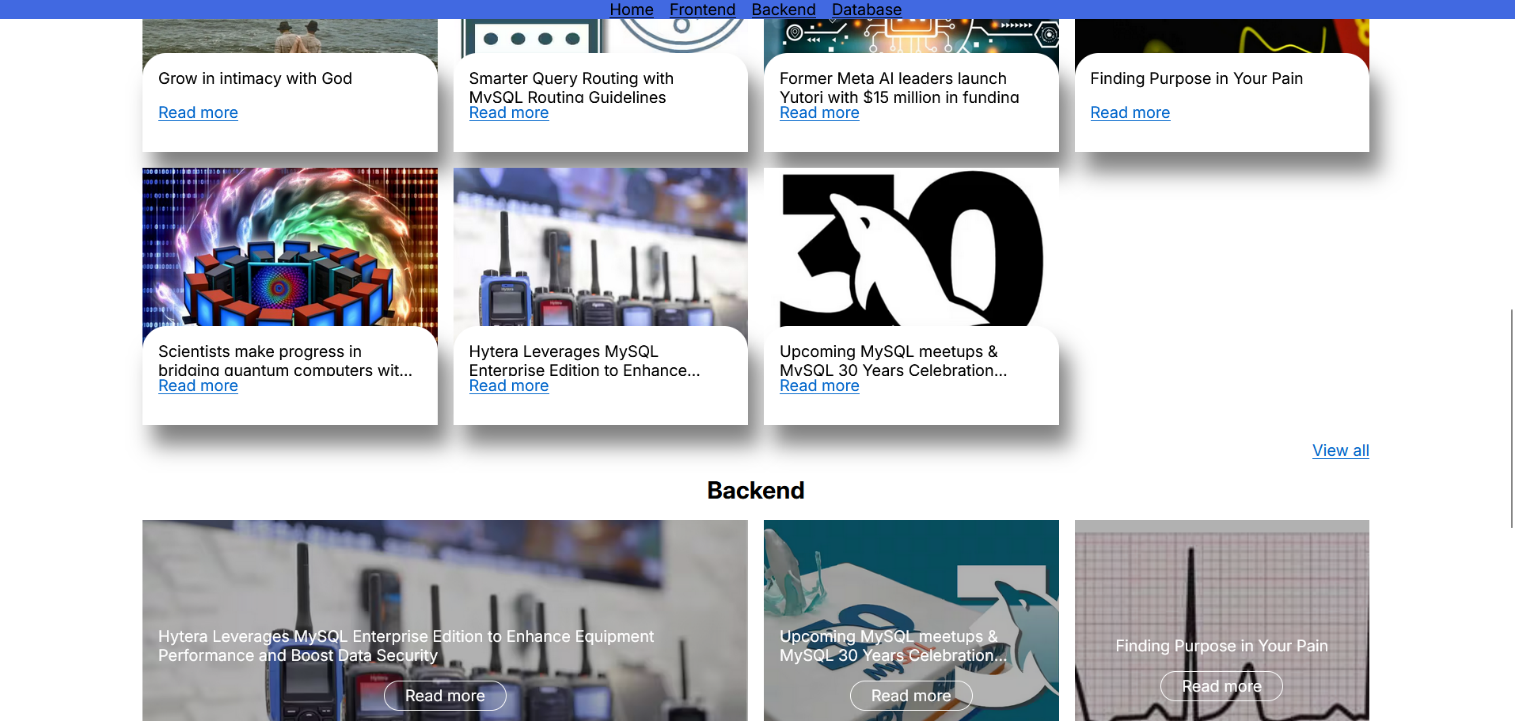
      </section>

    </main>

    <footer data-th-replace="~{/fragment/basic-fragment :: footer}"></footer>

  </body>

</html>



# CHAPTER NINE

## TESTING

##### Introduction

Testing is a crucial phase in the \*Tech Blog Management System\* development process to ensure the platform functions efficiently, securely, and as expected. It involves evaluating different components, including \*user authentication, content management, SEO integration, security protocols, and performance analytics\*, to identify and fix potential issues before deployment.

The primary goal of testing is to ensure a \*bug-free, user-friendly, and high-performance\* system that meets the requirements of \*admins, authors, and readers. Various testing methods, such as \*\*unit testing, functional testing, performance testing, security testing, and user acceptance testing (UAT)\*, are applied to validate the system’s reliability.

By implementing a structured testing approach, the \*Tech Blog Management System\* can provide a \*seamless experience, prevent security vulnerabilities, and enhance overall platform performance.

##### Unit Testing

\*Unit Testing\* is a software testing technique that focuses on testing individual components or functions of the \*Tech Blog Management System\* to ensure they work correctly. Each module, such as \*user authentication, blog post creation, commenting system, and SEO tools\*, is tested in isolation to verify its correctness.

\*Key Aspects of Unit Testing:\*

1. \*Objective\* – Ensure that each function, method, or module performs as expected before integrating it with the rest of the system.

2. \*Testing Scope\* – Covers individual components like \*user login, post publishing, comment submission, and media uploads\*.

3. \*Tools Used\* – Common unit testing frameworks include:

- \*Jest & Mocha\* (for JavaScript/Node.js)

- \*PyTest & Unittest\* (for Python-based backends)

- \*JUnit\* (for Java-based systems)

4. \*Test Cases Examples:\*

- Verify \*successful user login\* with valid credentials.

- Ensure \*blog posts save correctly\* to the database.

- Check if the \*commenting system prevents spam submissions\*.

- Validate \*SEO metadata generation\* for blog posts.

5. \*Automation & Continuous Integration (CI)\* – Unit tests can be automated and integrated into \*CI/CD pipelines\* to detect errors early in development.

# CHAPTER TEN

#### CONCLUSION

The \*Tech Blog Management System\* is a comprehensive platform designed to \*streamline content creation, enhance user engagement, and ensure efficient blog management. By integrating \*\*role-based access control, SEO tools, analytics, security features, and monetization options, the system provides a \*\*scalable and user-friendly\* solution for tech bloggers, readers, and administrators. With features such as \*content moderation, user reporting, and automated workflows, the system fosters a \*\*safe, organized, and interactive blogging environment. Additionally, \*\*future enhancements\* like \*AI-driven content recommendations, voice-to-text blogging, and Progressive Web App (PWA) support\* will further optimize the platform. In conclusion, this system empowers \*bloggers, content creators, and tech enthusiasts\* by providing a \*secure, efficient, and feature-rich\* environment for managing and growing their digital presence.

##### Project Limitation

##### While the \*Tech Blog Management System\* offers a robust platform for content creation and management, it has certain limitations that may affect its functionality and scalability. These include: 1. \*Scalability Constraints\* – As the number of users and blog posts increases, the system may experience \*performance issues\* if not optimized properly.

##### 2. \*Security Risks\* – Despite authentication and security measures, vulnerabilities like \*spam attacks, data breaches, and unauthorized access\* remain potential risks.

##### 3. \*SEO and Algorithm Changes\* – Search engine algorithms frequently update, requiring \*continuous SEO adjustments\* to maintain blog visibility.

##### 4. \*Limited Offline Functionality\* – The system primarily operates \*online, meaning users may have \*\*limited access\* to blog content and editing features without an internet connection.

##### 5. \*Dependency on Third-Party Services\* – Features like \*Google AdSense, cloud storage, and social media integration\* rely on third-party services, which may change policies or APIs, affecting system functionality. Addressing these limitations through \*regular updates, scalability improvements, and enhanced security protocols\* can help ensure a \*better user experience and long-term reliability\*.

# CHAPTER ELEVEN

## FUTURE ENHANCEMENTS

To improve the \*Tech Blog Management System\* and enhance user experience, several future enhancements can be implemented:

**1. \*AI-Powered Content Recommendations\*** – Implement \*machine learning algorithms\* to suggest relevant articles to users based on their reading history and interests.

**2. \*Voice-to-Text Blogging\*** – Enable bloggers to create posts using \*speech recognition technology\*, making content creation faster and more accessible.

**3. \*Progressive Web App (PWA) Support\*** – Develop a \*PWA version\* of the blog, allowing users to access and interact with content \*offline\* and on mobile devices seamlessly.

**4. \*Dark Mode & Customizable Themes\*** – Offer users the ability to switch to \*dark mode\* or customize the blog's appearance for better readability.

**5. \*Advanced Monetization Options\*** – Introduce \*membership plans, premium content access, and integrated e-commerce features\* for enhanced revenue generation.

**6. \*Blockchain-Based Content Verification\*** – Use \*blockchain technology\* to verify the authenticity of blog posts, preventing plagiarism and ensuring content integrity.

**7. \*Improved Comment Moderation\*** – Integrate \*AI-based spam detection and sentiment analysis\* to automatically moderate and filter inappropriate comments.

**8. \*Multi-Language Support\*** – Implement \*automatic translation\* and multi-language content creation to reach a global audience.

**9. \*Collaborative Blogging\*** – Allow multiple authors to collaborate on a single post with \*real-time editing and version control\*.

**10. \*API for Third-Party Integrations\*** – Develop a \*RESTful API\* to allow seamless integration with \*external applications, analytics tools, and automation services\*.

## REFERENCES

## Website:

## <https://blog.playstation.com>/

## <https://technorati.com/>

## <https://gigaom.com/>

## <https://www.droid-life.com/>