Software Requirements Specification

for

News-Hub

by

Halai Akshay Nanjibhai (21SOECE11027),6CEA

Khunt Varshil Mahendrabhai (21SOECE11047), 6CEA

Kathiriya Yash Narshibhai (21SOECE11043), 6CEA

Department: B. Tech CE(6th semester)

RK University,Rajkot – Gujarat

MARCH **2024**

Table of Contents

1. Introduction

1.1 Purpose

The Software Requirements Specification (SRS) document for the News-Hub web application project serves as a comprehensive blueprint outlining the functional and non-functional requirements, user interactions, system architecture, and constraints. It establishes a shared understanding between stakeholders and the development team, guiding the design, implementation, and testing phases to ensure the successful delivery of a robust and user-friendly news aggregation platform.

1.2 Document Conventions

The document maintains a straightforward structure with numbered sections for clarity. It employs a uniform writing style and formatting, ensuring easy comprehension. Technical terms and abbreviations are defined, and symbols in charts or tables are accompanied by clear explanations for enhanced understanding.

1.3 Intended Audience and Reading Suggestions

The intended audience for this document includes project stakeholders, developers, and testers. Stakeholders can gain insight into project requirements, while developers and testers can refer to specific sections relevant to their roles for implementation and validation purposes. Readers are encouraged to focus on sections pertaining to their respective responsibilities to streamline comprehension and facilitate efficient project execution.

1.4 Project Scope

The project scope encompasses the development of a News-Hub web application, aiming to provide users with a centralized platform for accessing and consuming news content. Features

include user authentication, content aggregation from various sources, personalized news feed generation, and social sharing capabilities. The scope excludes advanced analytics and extensive social networking functionalities, focusing primarily on delivering a seamless news browsing experience.

1.5 References

2. Overall Description

2.1 Product Perspective

- The News-Hub web application is a new, self-contained product designed to provide users with a centralized platform for accessing news content.
- It is not a part of an existing product family but serves as an independent solution for news aggregation and consumption.

- The application is intended to replace traditional news browsing methods by offering a more streamlined and personalized experience.

- It interacts with various external sources to gather news articles, including news websites, RSS feeds, and social media platforms.

- The system comprises major components such as a user interface, backend server, database, content aggregator, and social sharing functionality.

- External interfaces include APIs for retrieving news data from external sources and integration with social media platforms for sharing capabilities.

- While it is a standalone product, future versions may incorporate additional features based on user feedback and market trends.

- The overall system architecture is depicted in a simple diagram illustrating the interaction between components and external interfaces.

2.2 Product Features

- User authentication: Allows users to create accounts, log in securely, and manage their profiles.

- Content aggregation: Aggregates news articles from various sources into a centralized platform.

- Personalized news feed: Provides users with personalized news recommendations based on their interests and preferences.

- Social sharing: Enables users to share news articles with their social networks via integrated sharing functionality.

- Search functionality: Allows users to search for specific news articles based on keywords, topics, or sources.

2.3 User Classes and Characteristics

- Casual Users: Infrequent visitors who browse news articles occasionally without creating accounts.

- Characteristics: Limited interaction with the platform, primarily interested in accessing news content quickly and easily.

- Registered Users: Regular visitors who create accounts to personalize their news feed and access additional features.

- Characteristics: Willing to invest time in setting up their profiles to receive tailored news recommendations.

- Power Users: Enthusiastic news consumers who actively engage with the platform, contribute comments, and share articles frequently.
- Characteristics: Highly engaged with the community aspect of the platform, seek to interact with other users and contribute to discussions.

- Administrators: Platform moderators responsible for managing user accounts, monitoring content, and enforcing community guidelines.

- Characteristics: Require elevated privileges to perform administrative tasks and ensure the platform's integrity and security.

- Developers: Technical personnel involved in maintaining and enhancing the platform's functionality and performance.

- Characteristics: Possess advanced technical expertise and access to backend systems for troubleshooting and development purposes.

2.4 Operating Environment

- Hardware Platform: The News-Hub web application, built using the MERN (MongoDB, Express.js, React.js, Node.js) stack, requires standard web servers with adequate processing power and storage capacity to support the technology stack.

- Operating System: Compatible with various operating systems including Windows, macOS, and Linux, allowing flexibility for deployment.

- Web Server: Works seamlessly with web server environments supporting Node.js applications, such as Node.js built-in HTTP server, or third-party servers like Express.js.

- Database System: Relies on MongoDB, a NoSQL database, for data storage and retrieval, necessitating compatibility with MongoDB versions specified by the MERN stack.

- Web Browser: Supports modern web browsers including Google Chrome, Mozilla Firefox, Microsoft Edge, and Safari, ensuring optimal user experience for accessing the React.js frontend.

- Internet Connection: Users require a stable internet connection to interact with the web application, leveraging Node.js for server-side operations and React.js for client-side rendering.

2.5 Design and Implementation Constraints

- Technology Stack: Developers must adhere to the MERN (MongoDB, Express.js, React.js, Node.js) stack for consistent design and implementation.

- Corporate Policies: Compliance with corporate policies regarding data privacy, security measures, and coding standards is mandatory.

- Regulatory Requirements: Adherence to relevant regulatory standards such as GDPR for handling user data and content moderation.

- Hardware Limitations: Consideration of hardware limitations in terms of server capacity, memory requirements, and processing power for scalability and performance.

- Interface Compatibility: Ensuring compatibility with external APIs or interfaces for integrating with third-party applications or services.
- Security Considerations: Implementation of robust security measures to prevent unauthorized access, data breaches, and ensure user privacy.

- Design Conventions: Following design conventions and programming standards established by the development team or the customer's organization for consistent codebase maintenance.

2.6 User Documentation

- User Manual: A comprehensive guide detailing how to navigate the News-Hub web application, create user accounts, customize news preferences, and utilize various features.

- Online Help: Interactive help resources integrated within the application to provide contextual assistance and guidance on using specific features or resolving common issues.

- Tutorials: Step-by-step tutorials or video guides demonstrating key functionalities, best practices, and tips for maximizing user experience.

- FAQs (Frequently Asked Questions): A compilation of frequently asked questions and their corresponding answers to address common user inquiries and troubleshooting scenarios.

- Knowledge Base: A repository of articles, guides, and resources covering advanced topics, troubleshooting techniques, and additional information for users seeking in-depth assistance.

2.7 Assumptions and Dependencies

- Assumption: The availability and reliability of third-party APIs for sourcing news content from various sources remain consistent throughout the project development phase.

- Assumption: The development team possesses the necessary skills and expertise to implement and integrate the MVC of ASP .NET components effectively.

- Assumption: The hosting environment for the web application provides sufficient scalability, security, and performance capabilities to accommodate user traffic and data storage requirements.

- Dependency: The timely delivery and compatibility of any external software components or libraries intended for integration with the News-Hub application, such as authentication modules or social media APIs.

- Dependency: Adherence to agreed-upon project timelines and milestones to ensure smooth coordination with stakeholders, resource allocation, and timely delivery of deliverables.

3. System Features

1. User Authentication
   - Description: Allows users to create accounts, log in securely, and manage their profiles.
   - Requirements:
     - User registration with email and password.
     - Secure login mechanism with session management.
     - Password recovery and account management functionalities.

2. Content Aggregation
   - Description: Aggregates news articles from various sources into a centralized platform.
   - Requirements:
     - Integration with external APIs and RSS feeds to fetch news content.
     - Automated content retrieval and updating mechanisms.
     - Categorization and tagging of news articles for easy navigation.

3. Personalized News Feed
   - Description: Provides users with personalized news recommendations based on their interests and preferences.
   - Requirements:
     - User profile management to capture user preferences.
     - Machine learning algorithms for content recommendation.
     - Customizable news feed based on user interactions and feedback.

4. Social Sharing

- Description: Enables users to share news articles with their social networks via integrated sharing functionality.
- Requirements:
  - Integration with social media platforms such as Facebook, Twitter, and LinkedIn.
  - Seamless sharing options within the application interface.
  - Tracking and analytics to monitor sharing activities.

5. Search Functionality
- Description: Allows users to search for specific news articles based on keywords, topics, or sources.
- Requirements:
  - Full-text search capabilities across news articles.
  - Advanced search filters for refining search results.
  - Search suggestions and autocomplete functionality for user convenience.

4. External Interface Requirements

4.1 User Interfaces

- Home Page: Intuitive interface displaying a curated selection of top news articles across various categories.

- Business Section: Dedicated section showcasing the latest business news and market updates in a visually appealing layout.

- Science Section: Explore groundbreaking discoveries and scientific advancements through an organized display of science-related news articles.

- Health Section: Access informative articles covering health tips, medical breakthroughs, and wellness advice for a healthier lifestyle.

- Entertainment Section: Engaging interface featuring entertainment news, celebrity updates, movie reviews, and cultural highlights.

- Sports Section: Dynamic section highlighting sports news, match summaries, player interviews, and live event coverage.

- Technology Section: Stay informed about the latest in technology trends, gadget reviews, and innovation stories in an easily navigable format.

- Registration Page: Simple and user-friendly registration form for new users to create accounts and access personalized features.

- Login Page: Secure login interface allowing existing users to access their accounts and personalized news feeds with ease.

- Signup Page: Convenient signup process enabling users to register quickly and start exploring news content tailored to their interests.

4.2 Hardware Interfaces

- Device Compatibility: Ensure the software is compatible with a wide range of devices including desktop computers, laptops, tablets, and smartphones.

- Data Transfer Protocol: Implement communication protocols such as HTTP or HTTPS for data exchange between the software and hardware components.

- Input/Output Devices: Support standard input/output devices like keyboards, mice, touchscreens, and displays for user interaction.

- Data Storage Devices: Interface with storage devices such as hard drives, solid-state drives, or cloud storage solutions for storing and retrieving user data and application files.

- Peripheral Devices: Interface with peripheral hardware devices like printers, scanners, or cameras if necessary for additional functionalities or integration requirements.

4.3 Software Interfaces

1. ASP.NET Core MVC
   - Interface: ASP.NET Core MVC will serve as the primary web application framework to handle HTTP requests and responses.
   - Data Items: Request parameters, session data, and response payloads.
   - Purpose: ASP.NET Core MVC will facilitate routing, middleware execution, and response handling for seamless interaction with clients.

2. Entity Framework Core (EF Core)
   - Interface: EF Core will be used for interacting with the database to store and retrieve structured data such as user profiles, news articles, and preferences.
   - Data Items: User credentials, user preferences, news articles, comments, and metadata.
   - Purpose: EF Core will serve as the primary ORM for persistent data storage and retrieval, providing a structured approach to defining models, schemas, and data access methods.

3. ASP.NET Core Identity
   - Interface: ASP.NET Core Identity will handle authentication and authorization processes, generating and validating secure authentication tokens.

- Data Items: JSON Web Tokens containing user authentication credentials and claims.
- Purpose: ASP.NET Core Identity will facilitate secure user authentication and authorization, enabling access control and session management.

4. ASP.NET Core HttpClient
- Interface: ASP.NET Core HttpClient will be used as the HTTP client library for making asynchronous HTTP requests to external APIs.
- Data Items: HTTP request parameters, response data, and error messages.
- Purpose: HttpClient will simplify integration with external APIs by providing a convenient and promise-based interface for making HTTP requests.

5. ASP.NET Core Bundling and Minification Middleware**
- Interface: ASP.NET Core bundling and minification middleware will package and optimize frontend assets such as JavaScript files, stylesheets, and images.
- Data Items: Bundled assets, configuration files, and build output.
- Purpose: This middleware will streamline the frontend development process by managing dependencies, optimizing code, and generating production-ready bundles.

6. ASP.NET Core Session State
- Interface: ASP.NET Core session state will be used for managing user session data and maintaining state across requests.
- Data Items: Session data specific to each user's session.
- Purpose: ASP.NET Core session state will enable the storage of user-specific data during a user's interaction with the web application.

You can copy this content and paste it into a Word file as needed.

4.4 Communications Interfaces

- HTTP Protocol: Utilized for communication between the client (web browser) and the server, ensuring reliable data transfer over the internet.
  - Specifies message formatting for requests and responses, facilitating seamless interaction between client and server.

- Secure Socket Layer (SSL) / Transport Layer Security (TLS): Implemented to ensure communication security and encryption, protecting sensitive data during transmission.
  - Ensures secure data transfer over the network, safeguarding against unauthorized access and data breaches.

- RESTful API: Used for client-server communication, defining standardized methods and message formats for data exchange.
  - Specifies HTTP methods (GET, POST, PUT, DELETE) for performing CRUD (Create, Read, Update, Delete) operations on server resources.

-

Other Nonfunctional Requirements

4.5 Performance Requirements

- Response Time: The web application should load within 3 seconds on average under normal traffic conditions to ensure a seamless user experience and prevent user frustration.

- Scalability: The system should be able to handle a concurrent user load of at least 1000 users without significant degradation in performance to accommodate potential spikes in user traffic.

- Database Query Performance: Database queries should execute within 200 milliseconds on average to ensure efficient data retrieval and responsiveness of the application.

- File Upload Speed: The system should support file uploads with a minimum transfer rate of 1 MB/s to ensure timely uploading of multimedia content such as images or videos.

- Search Query Response Time: Search queries should return results within 1 second on average to provide users with quick access to relevant content and enhance usability.

4.6 Safety Requirements

- Data Security: The system must implement encryption mechanisms to safeguard sensitive user data (such as passwords and personal information) stored in the database to prevent unauthorized access and data breaches.

- Error Handling: The application should have robust error handling mechanisms in place to gracefully handle unexpected errors and prevent system crashes or data corruption, ensuring uninterrupted service availability.

- User Authentication: Strong authentication measures, such as multi-factor authentication or CAPTCHA verification, should be implemented to prevent unauthorized access to user accounts and protect against account hijacking or identity theft.

- Backup and Recovery: Regular data backups should be performed to mitigate the risk of data loss due to hardware failure, software bugs, or malicious attacks. Additionally, the system should have a well-defined recovery process to restore data in case of an unforeseen event.

- Compliance with Regulations: The system must adhere to relevant data protection regulations such as GDPR (General Data Protection Regulation) or HIPAA (Health Insurance Portability and Accountability Act), ensuring compliance with legal requirements and safeguarding user privacy and rights.

4.7 Security Requirements

- User Authentication: Implement secure authentication mechanisms such as password hashing, session management, and account lockout policies to ensure only authorized users can access the system.

- Data Encryption: Utilize encryption techniques (e.g., SSL/TLS) for securing data transmission over the network and encrypt sensitive data at rest to protect against eavesdropping and unauthorized access.

- Access Control: Enforce role-based access control (RBAC) to restrict access to sensitive functionalities and data based on user roles and permissions, minimizing the risk of unauthorized actions.

- Security Auditing: Implement logging and monitoring functionalities to track user activities, system events, and security-related incidents for auditing purposes and timely detection of security breaches.

- Regular Security Assessments: Conduct periodic security assessments, penetration testing, and code reviews to identify and address potential vulnerabilities, ensuring continuous improvement of system security posture and resilience against cyber threats.

4.8 Software Quality Attributes

- Usability: The software should achieve a usability score of at least 80% in user testing assessments, measured through standardized usability metrics such as SUS (System Usability Scale) or NPS (Net Promoter Score), ensuring an intuitive and user-friendly interface.

- Reliability: The system should maintain an uptime of 99.9% over a one-month period, as measured by continuous monitoring and incident response protocols, ensuring consistent availability and reliability for users.

- Maintainability: The codebase should adhere to coding standards and best practices, achieving a code maintainability index score of 80 or above, facilitating ease of code maintenance, troubleshooting, and future enhancements.

- Security: The software should pass rigorous security assessments, achieving a minimum score of 90% in vulnerability scanning and penetration testing, ensuring robust protection against cyber threats and data breaches.

- Performance Efficiency: The application should achieve a loading time of less than 2 seconds for 90% of page loads, as measured by performance monitoring tools like Google Page Speed Insights or Lighthouse, ensuring optimal user experience and responsiveness.

5. Other Requirements

- Scalability: The system architecture should be designed to scale seamlessly with increasing user demand, ensuring consistent performance and availability as user traffic grows.

- Compliance: The application must adhere to relevant legal and regulatory requirements, including data protection laws, accessibility standards, and internationalization guidelines, to ensure legal compliance and user satisfaction.

Appendix A: Glossary

- ASP.NET Core MVC: Web application framework for ASP.NET Core, used for handling HTTP requests and responses similar to Express.js.

- ASP.NET Core Razor Pages: An alternative to ASP.NET Core MVC for creating dynamic web pages.

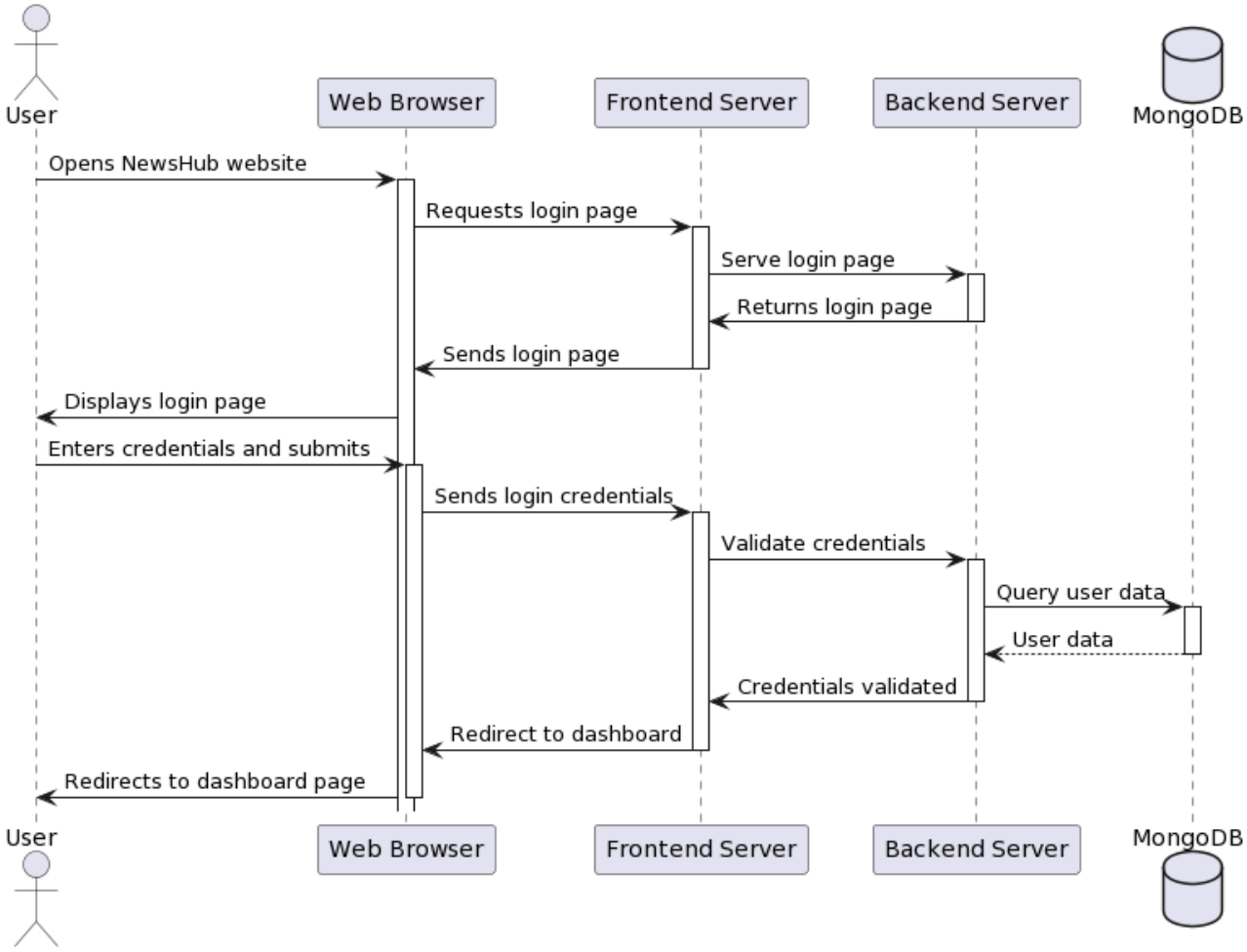
- API: Application Programming Interface, used for defining interactions between software components in ASP.NET Core applications.

- CRUD: Create, Read, Update, Delete. Basic operations for managing data in a database, similar to how it's used in ASP.NET Core applications.

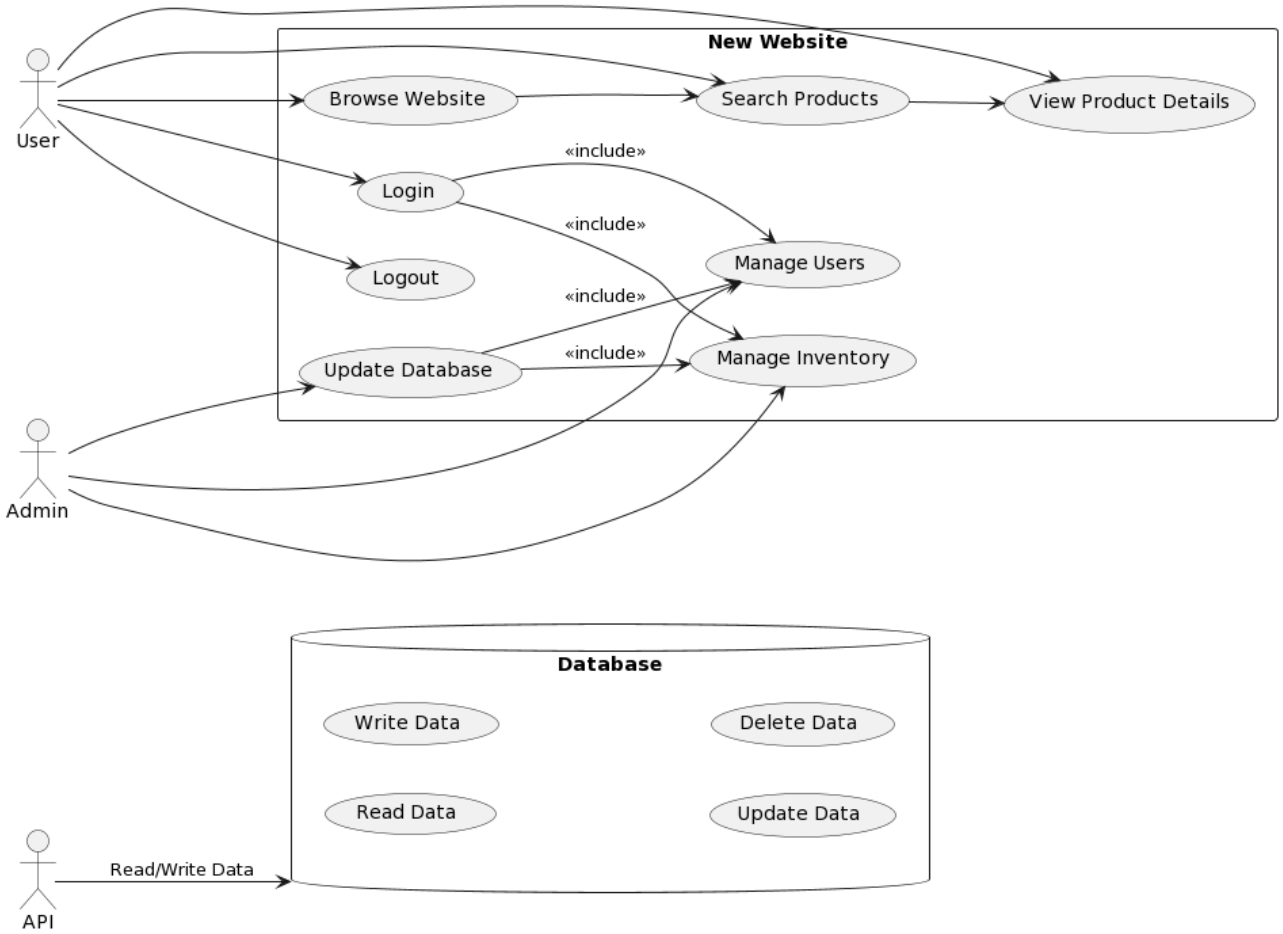
- HTTP: Hypertext Transfer Protocol. Protocol used for transmitting data over the internet, used in ASP.NET Core applications for client-server communication.

Appendix B: Analysis Models

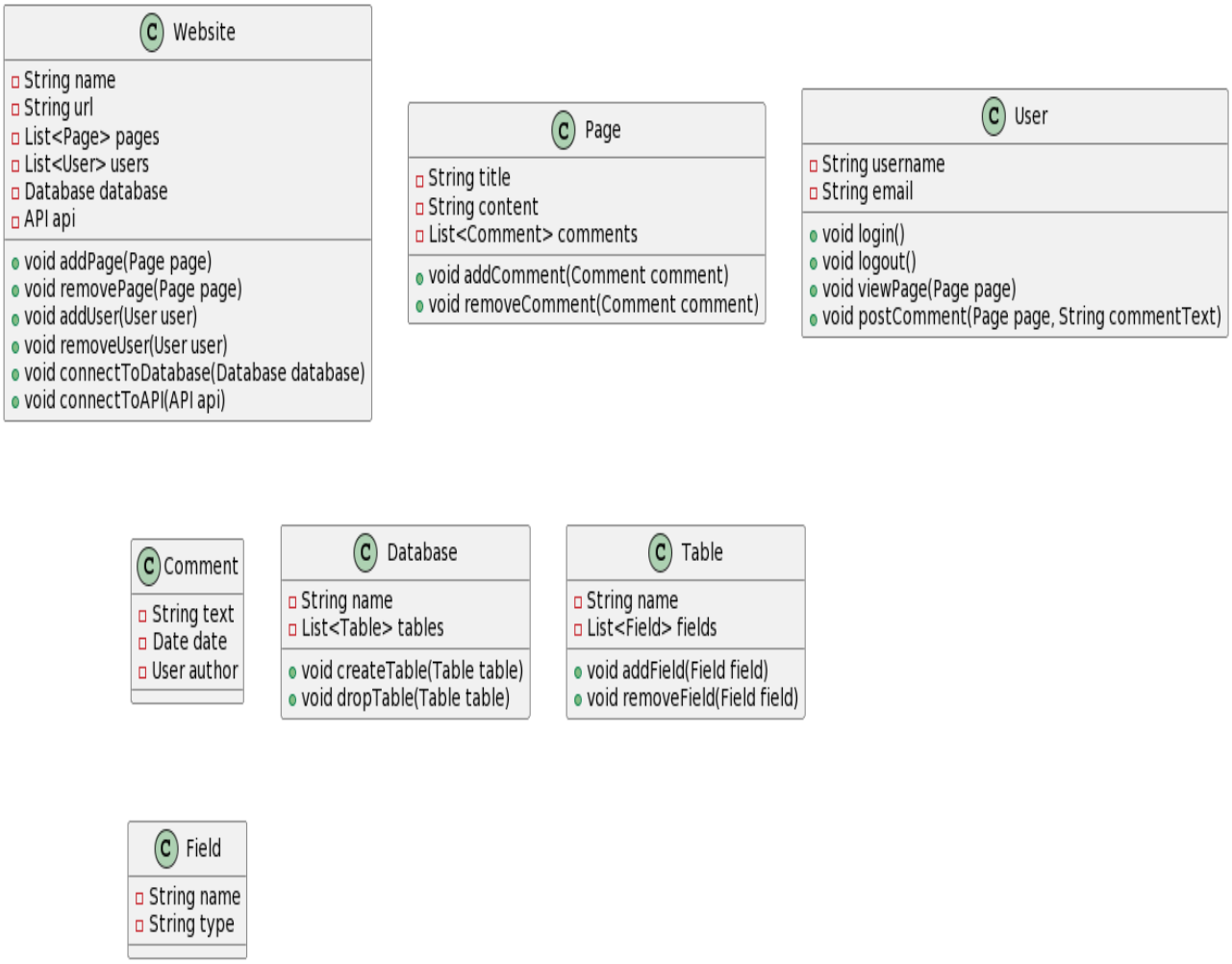
1)SEQUENCE DIAGRAM:



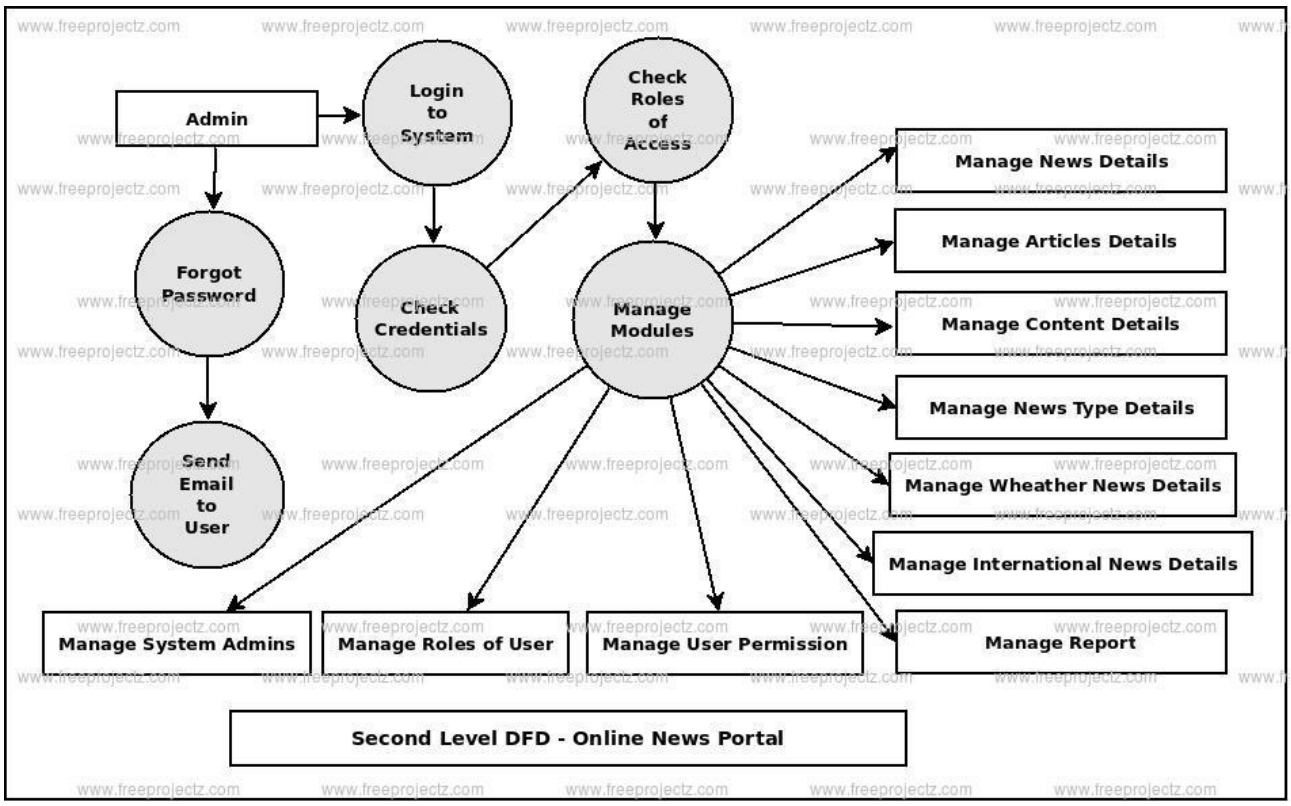
2) USECASE DIAGRAM:



3) CLASS DIAGRAM:



4) DFD DIAGRAM:



Appendix C: Issues List

- Clarify data validation requirements for user login credentials (TBD).

- Decide on the implementation approach for handling user session management (Pending decision).

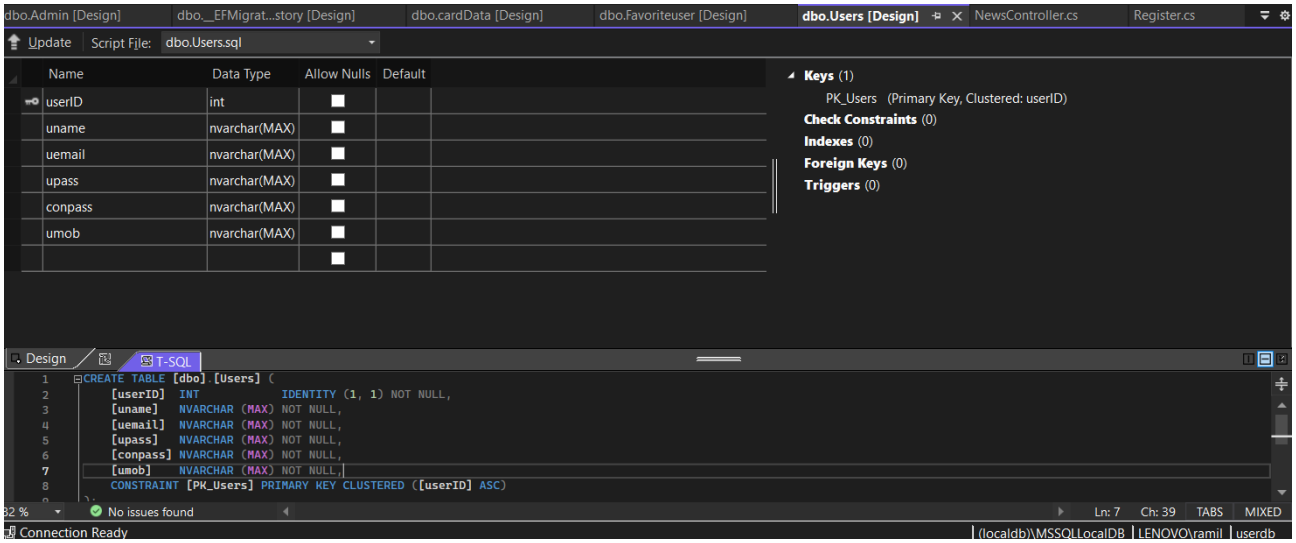
- Determine the error handling strategy for unexpected server-side errors (Pending decision).
- Obtain information on regulatory compliance requirements for user data protection (Information needed).

- Resolve conflicts in design preferences for the user interface layout (Conflicts awaiting resolution).

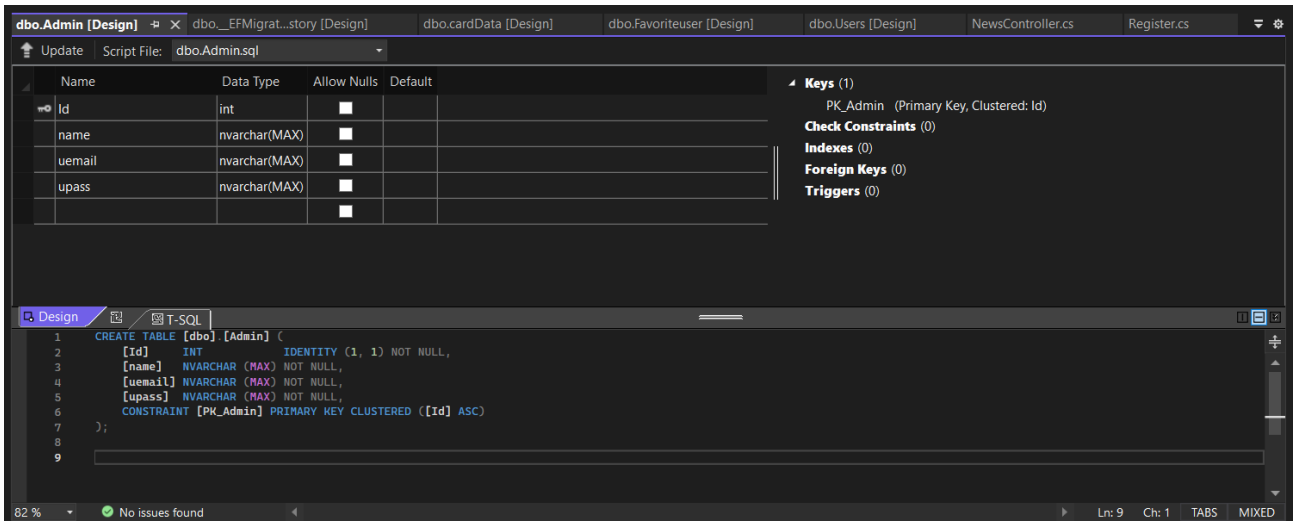
IMAGES OF THE WEB APPLICATION:

1) Database:

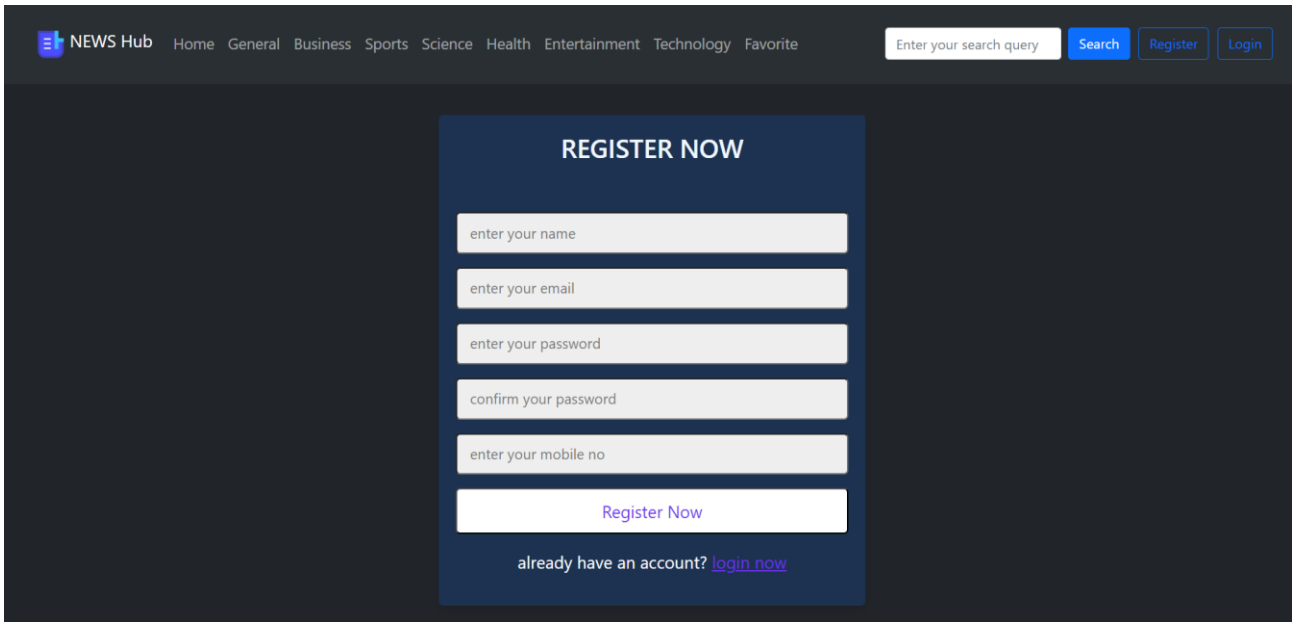
>USER:



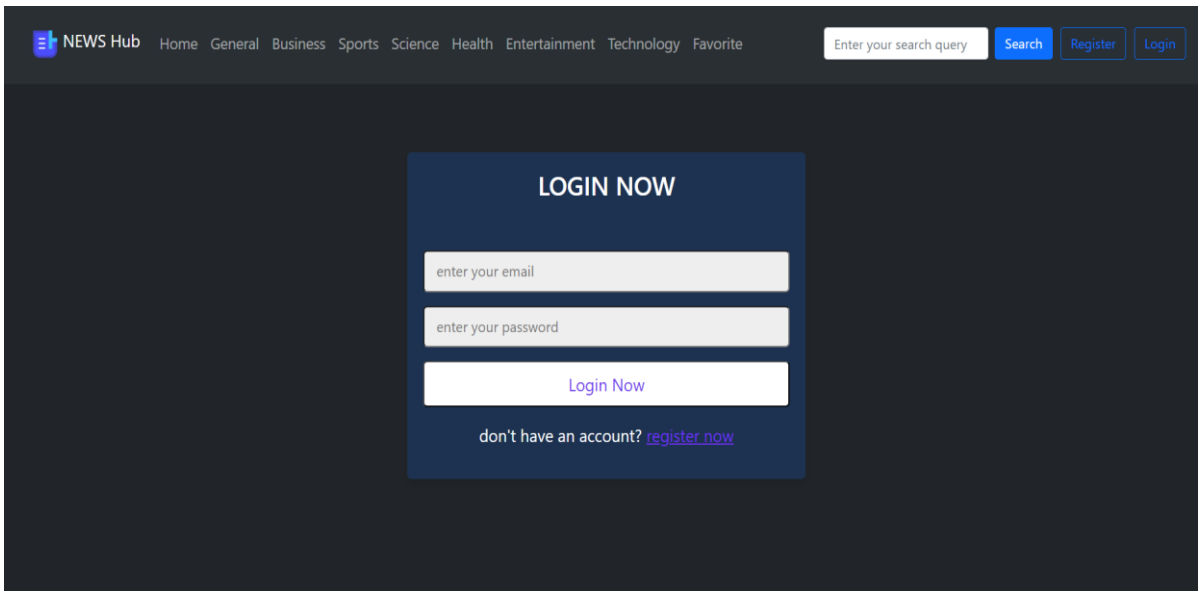
>ADMIN:



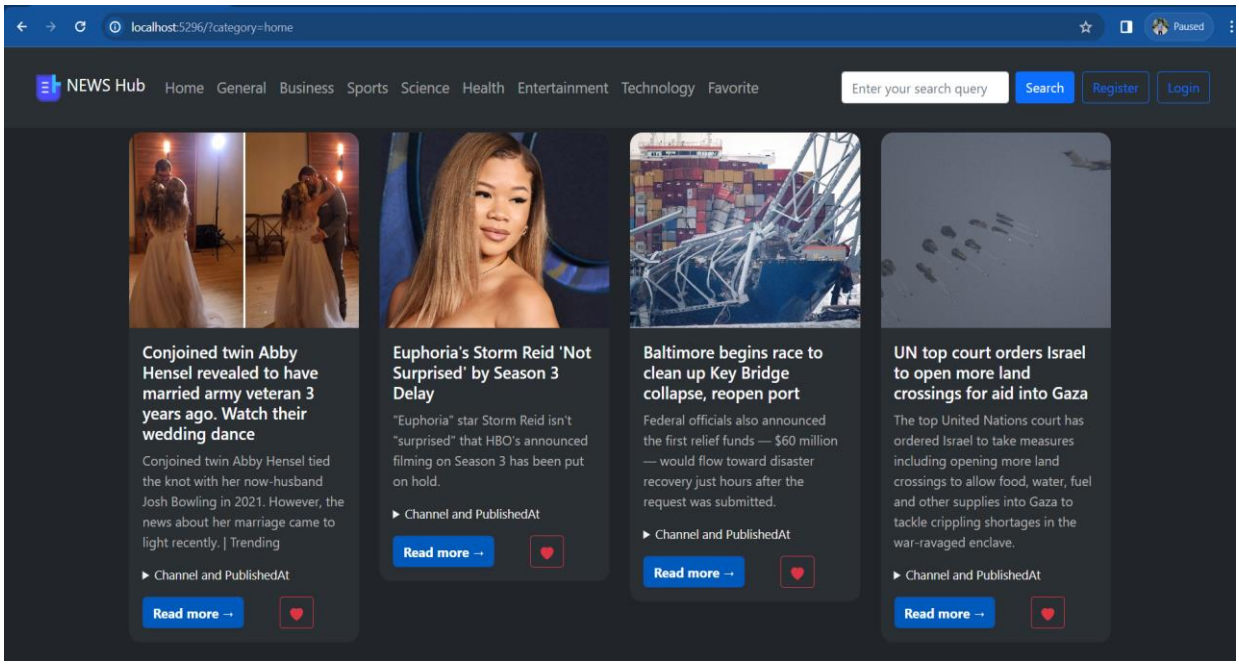
2) REGISTRATION PAGE:



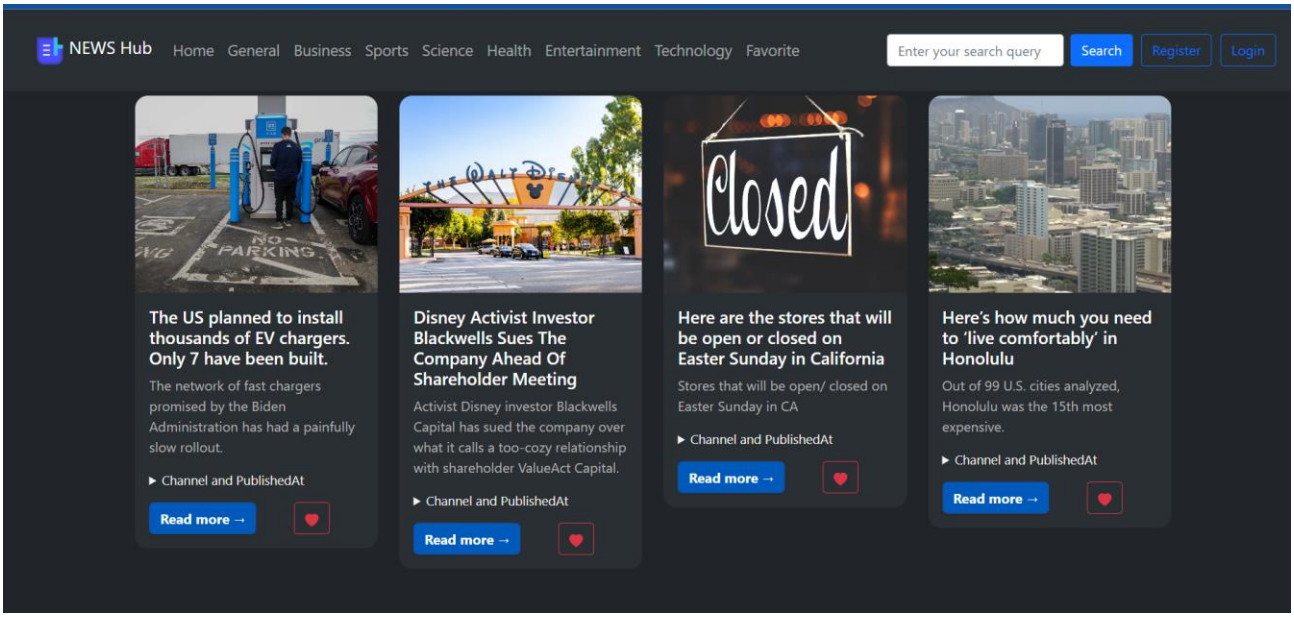
3) LOGIN PAGE:



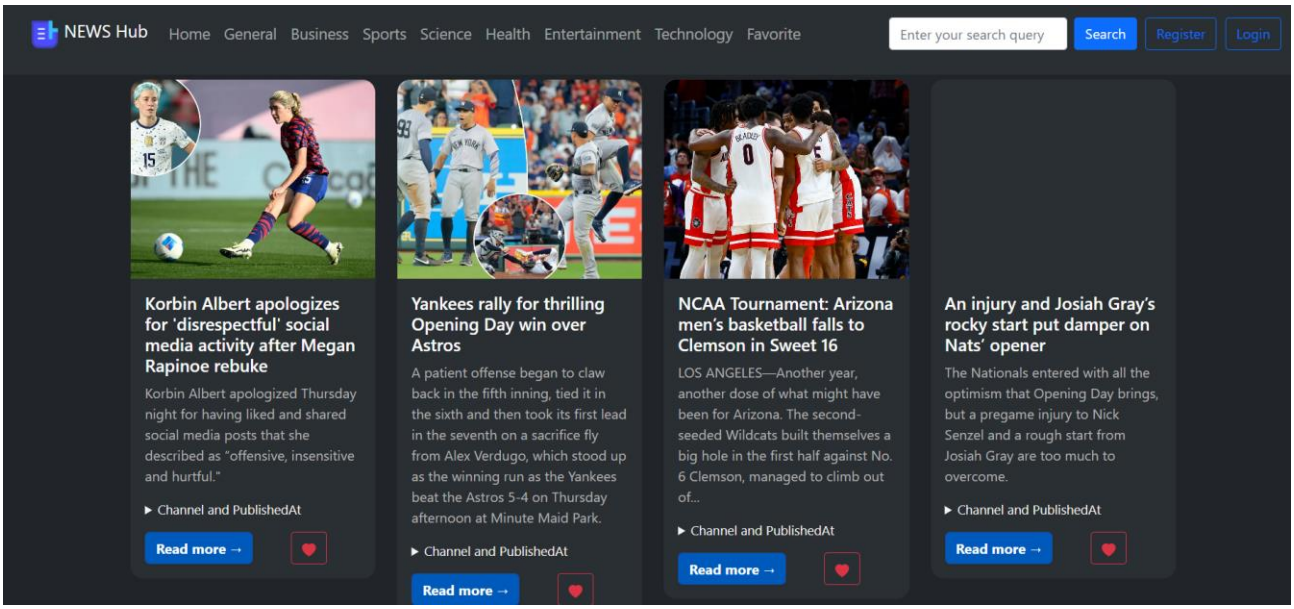
4) HOME PAGE:



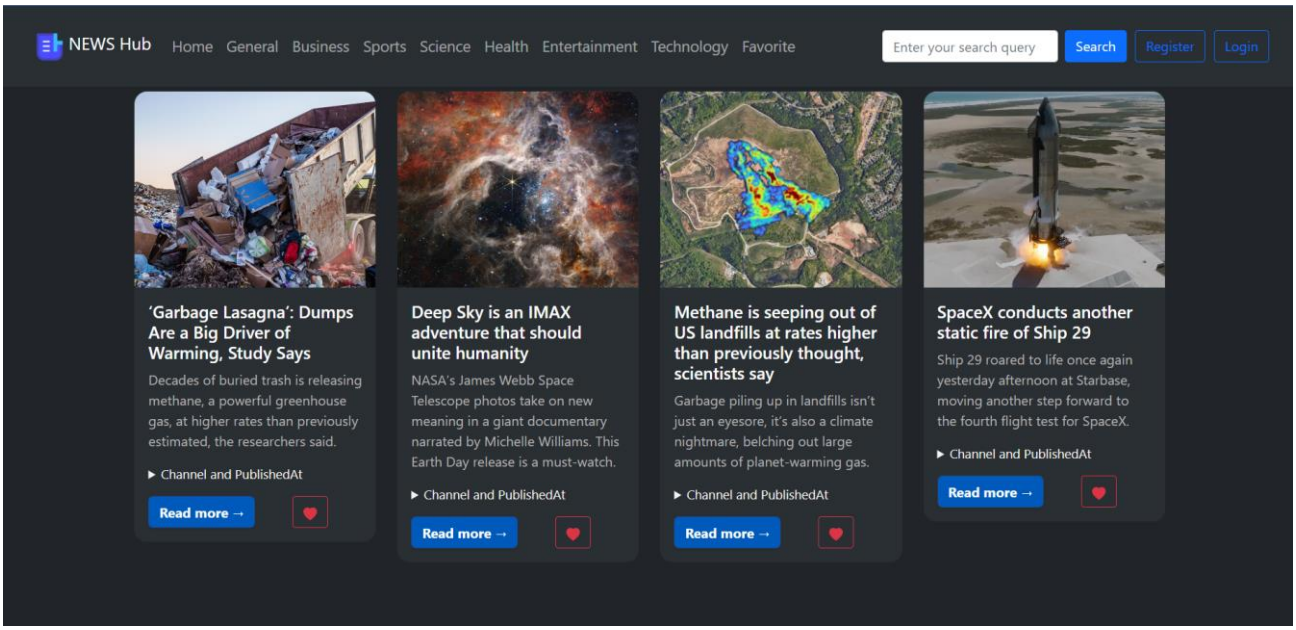
5) BUSINESS PAGE:



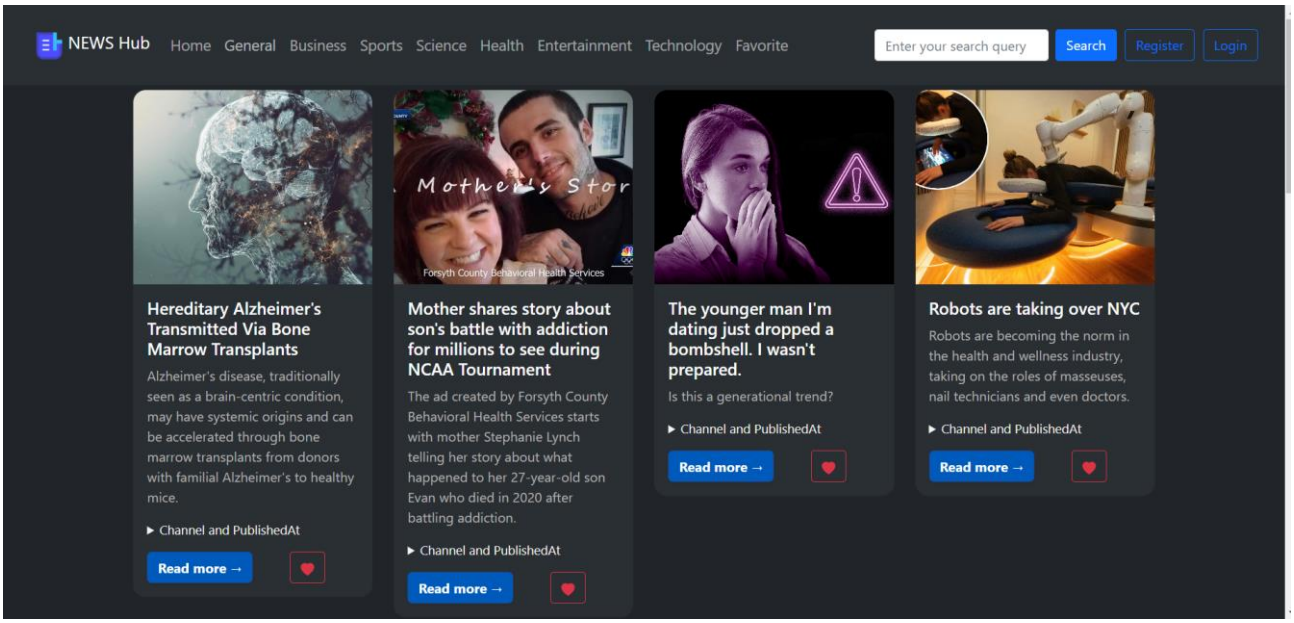
6) SPORT PAGE:



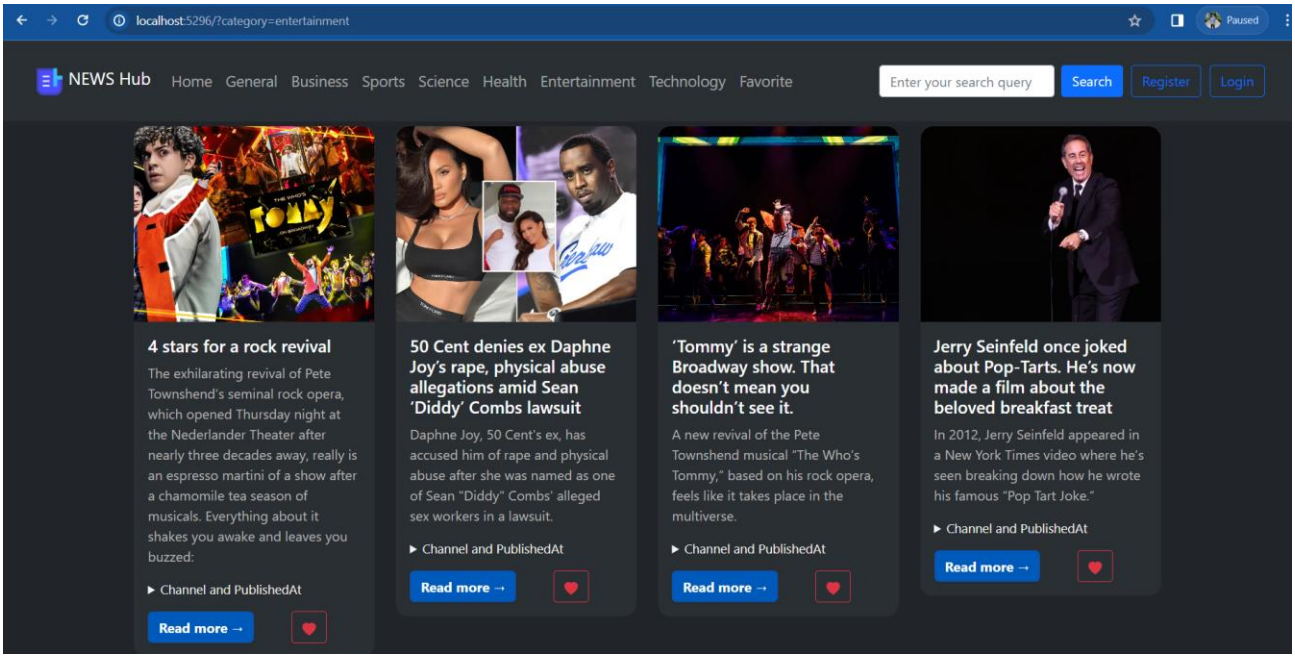
7) SCIENCE PAGE:



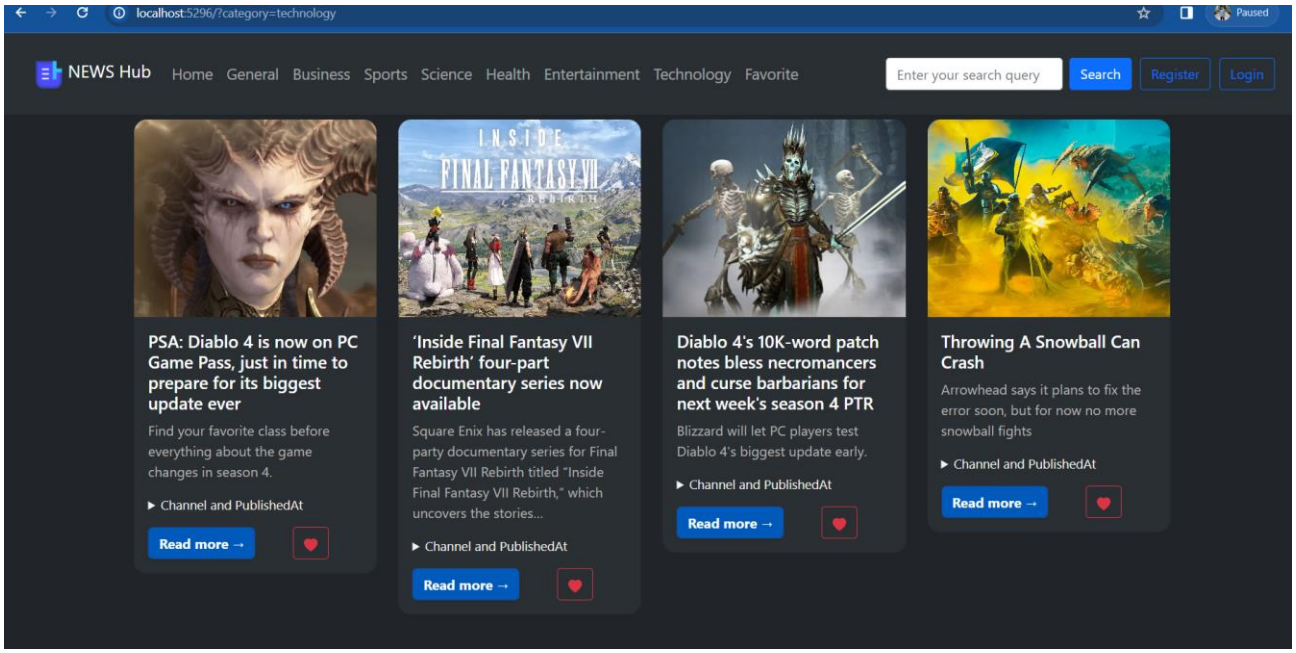
8) HEALTH PAGE:



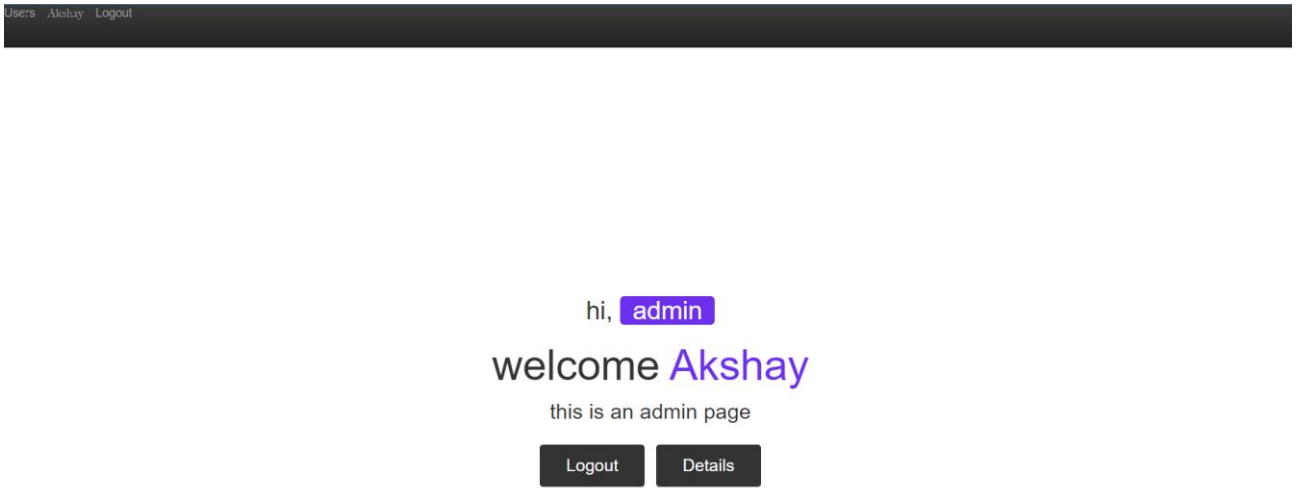
9) ENTERTAINMENT PAGE:



10)TECHNOLOGY PAGE:



⇨ ADMIN PNNEL :



=>DETAILS OF USERS: