International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 06 | June – 2025 DOI: 10.55041/ISJEM04016
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

DYNAMIC ONLINE NEWS AND MEDIA PORTAL USING DJANGO

T. Arif Mohiuddin *1, Ms. S. Chitra Nayagam *2

[1]PG Student, Department of Computer Application, Dr. M.G.R. Educational and Research Institute, Chennai, Tamil Nadu, India.

*2 Assistant Professor, Department of Computer Application, Dr. M.G.R. Educational and Research Institute, Chennai, Tamil Nadu, India.

ABSTRACT

The implementation and development of a News Web Application that offers a modern interface with advanced features to make the process of reading news easy and interesting. The application caters to users who read the newspaper daily, providing them with the ability to read and listen to news articles. The homepage of each user is dynamically generated based on their past searches to suggest relevant articles.
A comment section has been implemented to allow users to interact with articles in real- time, providing a platform for discussions and feedback. The application also includes an admin user portal, which allows the admin to perform CRUD operations and manage users. The application has been developed using HTML, CSS, JavaScript, and React, making it fully responsive and accessible from any device, including mobile, laptop, and tablet.
The application's backend is powered by a scalable and robust architecture utilizing Node.js, Express.js, and MongoDB to handle user authentication, data storage, and retrieval. The application's performance has been evaluated through extensive testing and benchmarking, demonstrating its ability to handle a large number of users and deliver content with minimal latency.
The application's security features, including encryption, authentication, and access control, have been implemented to ensure user data privacy and protection against cyber-attacks.
The paper concludes by highlighting the potential of the application and discussing possible future directions for development and research.

Key Words: Node.js, Express.js, React.js, MangoDB, CRUD operation.

I. INTRODUCTION

Nowadays, technology plays a vital role in our lives, and we cannot imagine a single moment without it. The era of computer technology has revolutionized the world, and most of our daily tasks depend on web applications. Websites have become a primary source of information for people, and they can access them anytime, anywhere, at low cost.
In this information age, information is a valuable resource, and we are developing our project to raise awareness among people. This project is an online news portal that aims to overcome the challenges faced by the traditional manual system. The main intention of this project is to develop a portal for managing web-based news, providing a convenient and easy-to-use display for people worldwide to learn and gain knowledge about current events.
The portal has two user types: users who can view and add comments, and administrators who manage and control the website. Users can view relevant information based on various categories provided by the administrator. They can also search and add comments, but their comments will require approval

from the administrator after providing their name and email. The website includes basic pages that provide relevant information, and users can view them accordingly.

Administrators are responsible for managing all the relevant actions that ensure the website's proper functioning, allowing users to view information and generate reports. This project's ultimate goal is to create a user-friendly news portal that serves as a reliable source of information for people worldwide.

In the digital age, online news portals have become the primary source of information for people around the world. This project, Dynamic Online News and Media Portal using Django, aims to create a fast, reliable, and interactive platform for publishing and accessing real-time news content. Built using Django, a powerful Python-based web framework, the portal allows administrators to easily manage news articles, categories, and multimedia content, while providing users with an engaging and seamless browsing experience.

II. LITERATURE SURVEY

The rise of digital platforms has revolutionized how news is produced, distributed, and consumed. This literature survey explores existing systems and previous works related to online news portals, content management systems, and web application development using Django, with a focus on identifying gaps and opportunities for innovation.

1. Traditional News Portals

Conventional news websites like **The Hindu**, **BBC News**, and **CNN** deliver real-time news content to users via manually managed CMSs. These platforms often use large-scale backend frameworks like Drupal or custom-built CMS platforms. While they offer powerful tools, they are often expensive to maintain and lack flexibility for smaller or educational projects.

Limitation: High maintenance cost, less flexibility, and complexity in customization for small teams.

2. Open Source CMS Platforms (WordPress, Joomla)

WordPress and Joomla are commonly used for developing news portals. They offer a wide range of plugins and themes, making it easier to manage content. However, they come with limitations in terms of **backend customization, scalability**, and **security**, especially when developing tailored solutions.

Limitation: Over-reliance on plugins, security vulnerabilities, and lack of control over backend logic.

3. Django-based News Portals

Django, a high-level Python web framework, is gaining popularity for building customized content portals. Its built-in **admin panel**, **ORM**, and **robust security** features make it ideal for building scalable and secure applications.

Reference: *"Django for Professionals" by William S. Vincent* – discusses how Django can be used to build professional-grade applications including content portals.

Strength: Highly customizable, secure, and supports rapid development.

4. Existing Academic Projects

Several academic research papers and student projects have developed Django-based content platforms. For instance, student-built platforms often focus on blog sites, educational resource sharing portals, or event management systems. These provide a foundation but often lack real- time news categorization and multimedia handling.

Gap Identified: Lack of category-based filtering, media uploads, and user interactivity (e.g., likes/comments).

5. News Aggregation Systems (e.g., Google News, Flipboard)

These platforms use AI and ML to aggregate news from various sources and personalize the content. While they offer high engagement, building such systems requires complex backend algorithms, which may not be feasible for a simple Django project.

Limitation: Complex infrastructure, heavy use of machine learning, and high resource requirements.

III. PROPOSED SYSTEM

The proposed Django-based online news portal addresses the limitations of existing systems with a modern, dynamic, and responsive solution:

• Automated content publishing and editing.

• Responsive layout for desktops, tablets, and smartphones.

• Structured article organization using categories and tags.

• Enhanced user experience with search and comment features.

• Integrated admin panel for full content control.

In this project we have develop an Online News Paper website. It is a dynamic system. It can be maintain and changed easily because it is based on database. Its contain web pages that are generated in real-time. These pages include Web scripting code, such as PHP. It is fully secured from unauthorized access. In a word it can say that our Online News Paper website is a completely dynamic website.

The **proposed system** is a modern, fully dynamic **online news and media portal** developed using the Django web framework. This system aims to overcome the limitations of traditional static or semi-dynamic news websites by providing an efficient, scalable, and interactive platform for delivering news and multimedia content in real time.

Built on Django, a high-level Python web framework, the proposed system leverages its powerful built-in features, such as an ORM (Object Relational Mapping), a robust admin interface, user authentication, and URL routing, to create a feature-rich and responsive news platform.

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 06 | June – 2025 DOI: 10.55041/ISJEM04016
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

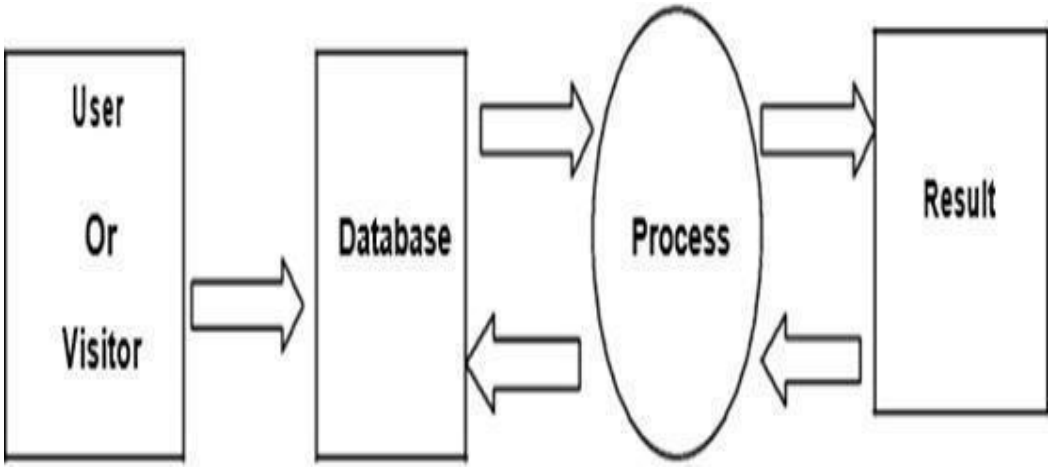

Figure 1: Architecture

IV.ALGORITHM

1. User Registration and Login

Step 1: User visits the website and selects the **"Register"** or **"Login"** option.
Step 2:

- If registering, the user fills in their username, email, and password.
- The system validates the input and creates a user account.
- If logging in, credentials are verified using Django's built-in authentication.
  Step 3: If authentication is successful, the user is redirected to the home/dashboard.
  Step 4: If credentials are invalid, an error message is displayed.

2. Admin Login and Dashboard Access

Step 1: Admin logs in through a dedicated admin panel (/admin).
Step 2: Django checks for admin permissions. **Step 3:** Admin is granted access to the dashboard. **Step 4:** Admin can perform CRUD operations on:

- News Articles
- Categories
- Comments

3. News Article Creation (Admin Only)

Step 1: Admin selects "Add Article" from the dashboard.
Step 2: Admin inputs:

- Title
- Category (dropdown)
- Content
- Image/Media File
  Step 3: System validates and stores data in the database.
  Step 4: The article is published and appears on the homepage or respective category page.

4. News Browsing by User

Step 1: User lands on the homepage.
Step 2: The system fetches the **latest articles** from the database.
Step 3: User clicks on a **category** or **searches** for a keyword.
Step 4: The system filters articles based on the request and displays them.

5. Reading a Full Article

Step 1: User clicks on a news headline.
Step 2: The system fetches the full content and media associated with that article using the article's ID.
Step 3: The article detail page is rendered with content and optional comment section.

6. Commenting on Articles (Optional)

Step 1: Logged-in user types a comment under an article.
Step 2: System stores the comment with:

- User ID
- Article ID
- Timestamp
  Step 3: Comment is displayed below the article.

7. Logout

Step 1: User or admin clicks on "Logout."
Step 2: Django ends the session and redirects the user to the homepage or login page.

Tools Used in Algorithm Execution

- **Django Views & Templates** for rendering logic
- **Django ORM** for database interaction
- **URL routing** for view dispatching
- **Django Auth** for session handling and login/logout
- **Media & Static Files** for content delivery

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 06 | June – 2025 DOI: 10.55041/ISJEM04016
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

Sequence Diagram

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

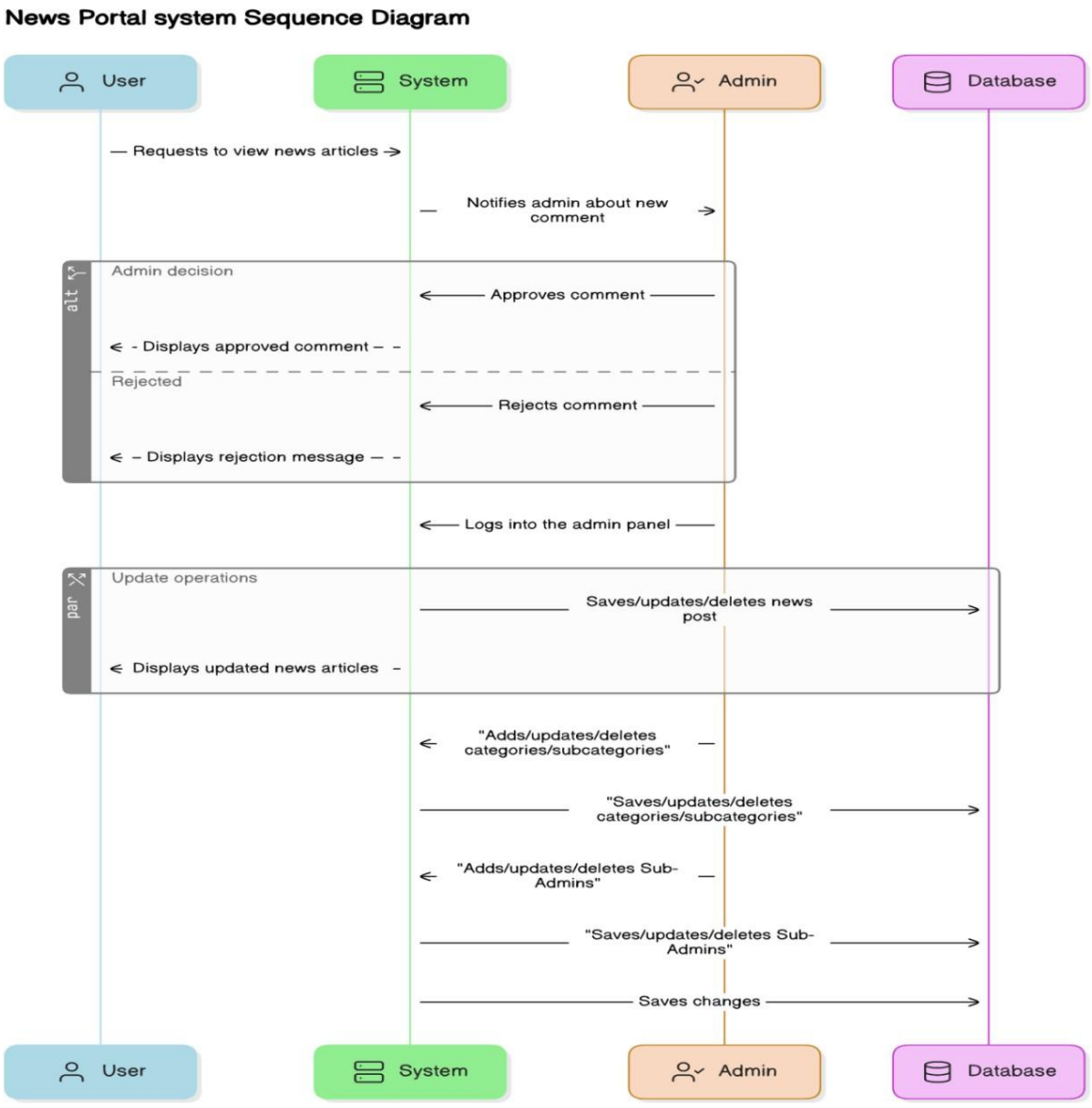

Figure 2: Sequence Diagram

V.CONCLUSION AND FUTURE ENHANCEMENT

CONCLUSION

The **Dynamic Online News and Media Portal** developed using **Django** successfully provides a platform for publishing, managing, and reading news articles in real-time. The system ensures smooth interaction between **admins** (who create and manage content) and **users** (who read, search, and comment on articles). Features like user authentication, category-based browsing, and a clean responsive interface make it user-friendly and efficient.
This project demonstrates how Django's powerful MVC architecture and built-in admin features can be used to build robust, scalable content management systems for real-world applications.

FUTURE ENHANCEMENTS

Integrate an AI-based recommendation engine that analyzes user behavior (such as reading history, time spent on articles, likes, shares, and comments) to provide personalized news content for each user. This could be powered by collaborative filtering or content-based filtering using machine learning models (e.g., using Django with Python's scikit-learn or TensorFlow).

Benefits:

- Enhances user engagement by showing relevant articles. Increases session time and return visits.
- Gives a competitive edge over traditional static portals.

Optional Add-ons:

- Integration with Natural Language Processing (NLP) to understand article content better. Real-time
- recommendation updates as the user interacts with content.

REFERENCE

1. Django Software Foundation. (2024). **Django Documentation**. Retrieved from: https://docs.djangoproject.com/en/stable/
2. William S. Vincent. (2022). **Django for Beginners: Build websites with Python and Django**. ISBN: 978-1735467207.
3. William S. Vincent. (2022). **Django for Professionals: Production websites with Python & Django**. ISBN: 978-1735467221.
4. Bootstrap Documentation. (2024). **Official Bootstrap Framework Docs**. Retrieved from: https://getbootstrap.com/
5. W3Schools. **HTML, CSS, JavaScript Tutorials**. Retrieved from: https://www.w3schools.com/
6. Stack Overflow Community. **Common Django Issues & Solutions**. Retrieved from: https://stackoverflow.com/questions/tagged/django
7. R. W. Sebesta. (2018). **Programming the World Wide Web** (8th Edition). Pearson Education.
8. MDN Web Docs. **Web Technologies Documentation**. Retrieved from: https://developer.mozilla.org/
9. GeeksforGeeks. **Django Framework Tutorials and Examples**. Retrieved from: https://www.geeksforgeeks.org/django-tutorial/
10. Real Python. **Tutorials on Django and Python Web Development**. Retrieved from: https://realpython.com/tutorials/django/