A

Mini Project Report

on

BuzzFeed

S.E.–Computer Science and Engineering – Data Science

Submitted By

Aayush Balip	**24207010**
Sujal Jain	**24207015**
Keval Shah	**24207020**
Vinayak Kshirsagar	**22107046**

Under The Guidance Of

Ms. Ujwala Pagare

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING DATA SCIENCE

A. P. SHAH INSTITUTE OF TECHNOLOGY

G. B. Road, Kasarvadavali, Thane(W), Mumbai-400615

UNIVERSITY OF MUMBAI

Academic year : 2024-25

CERTIFICATE

This to certify that the Mini Project report on BuzzFeed has been submitted by Aayush Balip (24207010), Sujal Jain (24207015), Keval Shah (24207020), Vinayak Kshirsagar (22107046) who are Bonafide students of A. P. Shah Institute of Technology, Thane as a partial fulfillment of the requirement for the degree in **Computer Science and Engineering Data Science**, during the academic year **2023-2024** in the satisfactory manner as per the curriculum laid down by University of Mumbai.

Ms. Ujwala Pagare
Guide

Ms. Anagha Aher
HOD, CSE Data Science

Dr.Uttam D. Kolekar
Principal

External Examiner:
1.

Internal Examiner:
1.

Place: A. P. Shah Institute of Technology, Thane
Date:

ACKNOWLEDGEMENT

This project would not have come to fruition without the invaluable help of our guide, Ms. Ujwala Pagare. We express our gratitude towards our HoD, Ms. Anagha Aher, and the Department of CSE Data Science for providing us with the opportunity as well as the support required to pursue this project. We would also like to thank our project coordinator Ms. Avani Nair, who gave us valuable suggestions and ideas when we needed them. Additionally, we would like to thank our peers for their helpful suggestions.

TABLE OF CONTENTS

Chapter 1

Introduction

In our rapidly evolving world, where information travels at the speed of light, staying informed is no longer a luxury—it's a necessity. Timeliness is paramount; whether it's breaking news, market updates, or scientific discoveries, being aware of events as they unfold allows us to make informed decisions promptly. Knowledge empowers us—it enables active participation in discussions, advocacy for causes, and engagement in civic responsibilities.

In today's fast-paced digital landscape, staying informed is essential. However, existing Python news applications face several limitations that hinder user experience and flexibility. These include heavy reliance on third-party APIs, dependency on update intervals, and a lack of comprehensive features for a user-centric experience. To bridge these gaps, we propose the development of a free-to-use Tkinter-based news application—the BuzzFeed App. Our goal is to create a user-friendly platform that offers comprehensive features without subscription fees. By leveraging Tkinter, we aim to provide an intuitive interface for users to access news seamlessly.

With its array of dynamic features, this innovative news application endeavors to redefine the user experience, offering a seamless blend of real-time updates, personalized content curation, and insightful data. At its core lies a meticulously crafted dashboard, powered by robust web scraping technology, providing users with instant access to the latest headlines sourced from news websites. Through a user-friendly interface, individuals can effortlessly navigate through various categories spanning economics, entertainment, politics, and more, tailoring their news consumption to suit their interests and preferences. Furthermore, BuzzFeed goes beyond mere consumption, empowering users to engage with content on a deeper level through features such as bookmarking, search functionality, and personalized news updates. Whether it's staying abreast of global events or delving into intricate economic trends, BuzzFeed stands as a trusted companion, offering unparalleled accessibility and insight in an ever-evolving digital landscape.

1.1 Purpose

The BuzzFeed project embodies a visionary spirit, aiming to redefine the way individuals engage with and consume news content. For news enthusiasts, this translates to a platform rich with curated headlines from various sources, each accompanied by comprehensive information and resources, ensuring that users stay informed with confidence and ease. On the flip side, seasoned journalists and analysts have the opportunity to effortlessly organize and present news stories, providing readers with vital details and insights into current

events. This app transcends traditional news consumption, offering a dynamic and accessible space where the thirst for information meets the thrill of discovery. It’s about uniting users with the stories that matter to them, fostering informed discussions and a deeper understanding of the world around us. In an era of digital empowerment, the BuzzFeed app embodies efficiency, accessibility, and global reach, opening up a world of news exploration for users around the globe.

Key Features:

- **Real-Time News Dashboard**: Implement a dynamic "Dashboard" feature where users can access real-time news headlines sourced from various reputable sources.
- **Categorized News**: Offer a comprehensive categorization system where news articles are organized into distinct categories such as economics, entertainment, politics, and more.
- **Search Functionality**: Provide users with a powerful search feature, enabling them to easily discover news articles related to specific topics or keywords.
- **Bookmarking and Personalization**: Enable users to bookmark news articles for later reading and reference, with the option to save articles directly to their personalized profile within the app.
- **My News**: Feature where users can customize their news feed based on preferred categories, receiving daily updates tailored to their interests.
- **Infographic Data Representation**: Introduce an innovative "Infographic" feature where users can explore visual representations of data and statistics related to trending news topics.
- **Email Feature**: Users can opt to receive email updates on preselected categories or topics.
- **Interests**: Users get news tailored to their specific interests.

1.2 Problem Statement

The problem at hand revolves around the challenges associated with accessing timely, comprehensive, and personalized news content in today's digital age. Existing news consumption platforms often lack the flexibility, feature richness, and user-friendly interface required to meet the diverse needs and preferences of users. These platforms may heavily rely on third-party APIs for content aggregation, limiting their customization options and real-time updates.

To address this issue, the proposed solution is the development of the BuzzFeed app, a Python-based news application designed to provide a flexible and feature-rich platform for news consumption. By leveraging web scraping technology with the BeautifulSoup library, the app aims to offer real-time news updates

sourced directly from Bing News, reducing reliance on third-party APIs. The app's objectives include aggregating real-time headlines, categorizing news into vital categories, implementing robust search functionality, enabling bookmarking of articles, providing personalized daily updates, visualizing trends through infographics, and offering an email feature for users to opt in to receive updates.

By developing the BuzzFeed app, we aim to revolutionize the way users consume news by providing them with a comprehensive, user-friendly, and personalized platform that caters to their diverse interests and preferences. This solution addresses the core challenges associated with accessing news content in the digital age, offering users a seamless and enriching experience in staying informed about global events and topics of interest.

1.3 Objectives

1. To provide flexible and feature-rich based news consumption platform using tkinter GUI

2. To enable real-time news updates for users by web scraping technology.

3. Provide a dynamic dashboard that aggregates real-time headlines offering users an instant overview of global events.

4. Develop a robust search option to empower users to find specific topics or keywords, enabling efficient and precise news discovery .

Explanation:

1. Build a flexible and feature-rich news consumption platform using Tkinter GUI

- Use Python's Tkinter library to design an intuitive and interactive desktop application.

- The GUI should be user-friendly, allowing users to easily navigate between news sections.

- Include various features and widgets, such as:

- Drop-down menus for category selection (Tech, Sports, Finance, etc.)

- Scrollable frames to browse through articles

- Optional themes (light/dark mode)

- Tkinter enables layout flexibility using `Frame`, `Canvas`, and `ttk` modules, ensuring a neat structure for the interface.

2. Enable real-time news updates through web scraping technology

- Integrate a web scraping engine (using `requests` + `BeautifulSoup` or `Selenium`) to fetch headlines and summaries from live news websites.

- Make sure scraping is respectful (add user-agents, control request frequency) and robust against layout changes on the source sites.

3. Create a dynamic dashboard to aggregate and display real-time global headlines

- Design a central news dashboard that aggregates headlines from multiple sources and categories.

- Display headlines in real-time as they're fetched, with clickable links or summaries.

- Tabs or frames for category-specific headlines

4. Implement a robust search function for keyword-based news discovery

- Includes a search bar allowing users to enter keywords (e.g., "Elections", "AI", "SpaceX").

- Filter and display only the news items that match the keyword from the current or full list.

Chapter 2

Proposed System

The proposed system aims to develop a next-generation Python-based news application titled "BuzzFeed." This application will provide users with real-time news updates, personalized content curation options, and interactive infographics for data visualization. It will offer features such as daily news updates, search functionality, user profiles with bookmarking capabilities, and customized email newsletters based on chosen categories, as illustrated in Figure 2.1. By leveraging advanced technologies like Tkinter, BeautifulSoup (bs4), and Matplotlib, "BuzzFeed" seeks to revolutionize the news consumption experience. It addresses the shortcomings of existing Python news applications and offers a comprehensive solution for users to stay informed in a user-friendly and personalized manner.

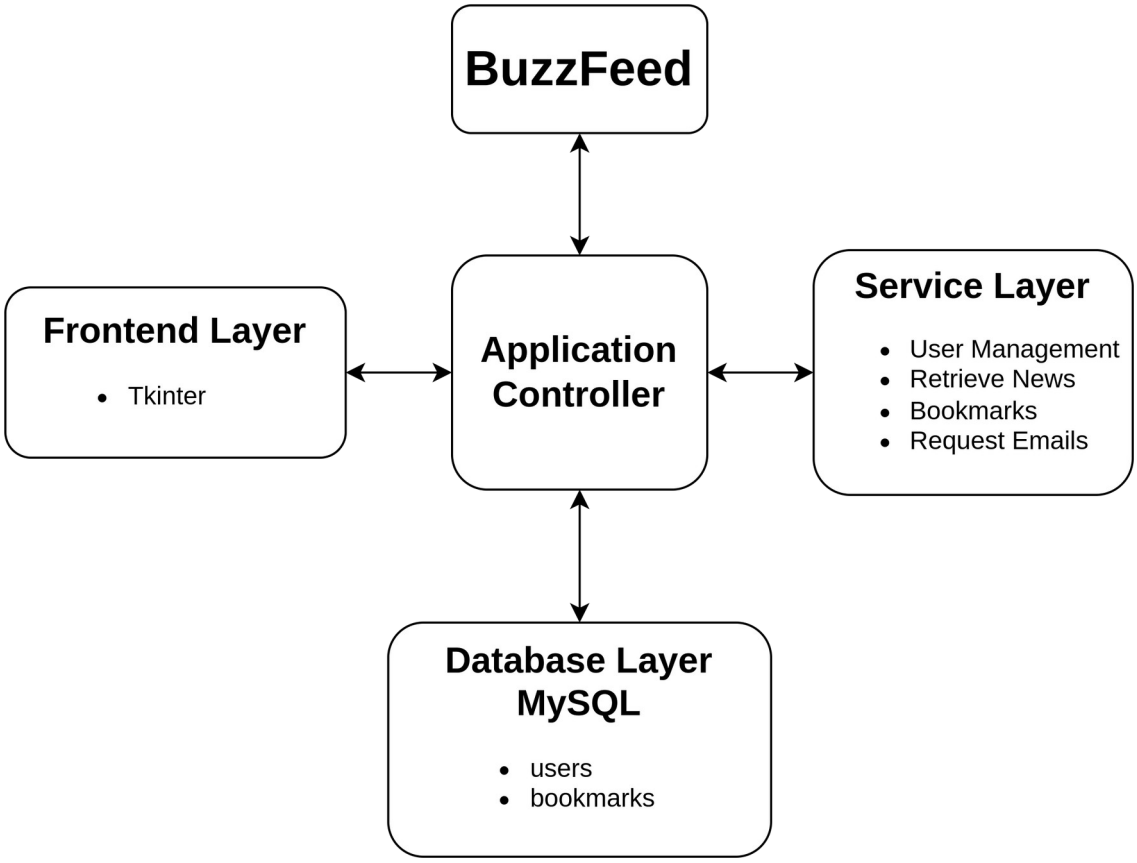

Figure 2.1 – Block Diagram of Proposed System of BuzzFeed application

1. Frontend Layer

- Technology Used: Python Tkinter

- Provide a clean, intuitive user interface.

- Enable interaction with features such as News Feed, Bookmark Article, My Profile, and News Section.
- Handle category selection and article viewing with smooth transitions.

2. Service Layer

- Technology Used: BeautifulSoup
- Webscrape the news content from “bing.com” and show the title to the user dashboard

3. Database Layer

- Efficient schema to store:
- Bookmarked articles.
- User profiles (preferences, saved articles, email and password).

2.1 Features and Functionality

1. **Secure Login System**
    - Users log in using registered credentials (email and password).
    - Ensures secure access to personalized features and user data.
    - Acts as the entry point to the full functionality of the application.

2. **Dynamic News Category Page**
    - Displays predefined news categories such as Sports, Finance, Politics, World, Entertainment, and Sci/Tech.
    - Allows users to explore news articles based on personal interests.
    - Simplifies content navigation through an intuitive and user-friendly interface.

3. **Search News Feature**
    - Users can enter keywords or phrases to search for specific news articles.
    - Provides a curated list of articles matching the user's search criteria.

4. **Daily News Updates**
    - Automatically fetches the latest news articles using web scraping techniques.
    - Ensures users stay updated with current events without manual searching.

5. **Email-Based News Delivery**
    - Users can subscribe to receive daily or keyword-based news via email.

- Delivers timely and relevant content directly to the user’s inbox.
- Supports personalized and convenient news consumption.

6. Interactive Infographics

- Visual representation of news data using charts and graphs (via Matplotlib).
- Improves understanding of data-heavy or statistical content.

7. User Profile Management

- Users can view and edit their personal information in the "My Profile" section.
- Allows customization of content preferences for a tailored experience.
- Stores user preferences to enhance future interactions.

8. My News

- Displays a history of recently searched topics and keywords.
- Enables users to revisit and continue exploring previous searches.

9. Bookmarks

- Lets users save favorite or important news articles.
- Facilitates quick access to saved content without repeated searching.

10. Web Scraping with BeautifulSoup (bs4)

- Extracts real-time news data from online sources in HTML/XML format.
- Ensures the application provides up-to-date content with minimal manual effort.

Chapter 3

Project Outcomes

The News Application project successfully met its objectives by delivering a functional, user-friendly, and efficient platform for accessing and managing news. The key outcomes of the project include:

1. User-Friendly Interface

The application was designed with a strong focus on usability and accessibility. It features an intuitive layout, clean design, and responsive behavior across different screen sizes and devices. The interface ensures that users, regardless of their technical background, can navigate and interact with the application easily. Buttons, menus, and content sections are organized logically, making the user experience smooth and pleasant.

2. Efficient News Aggregation

The system is capable of aggregating news articles from various reliable online sources using APIs and web scraping techniques. These articles are automatically sorted into different categories such as politics, sports, entertainment, technology, and more. This multi-source aggregation ensures content diversity and eliminates the need for users to browse multiple websites manually.

3. Real-Time Updates

One of the standout features of the application is its ability to fetch and display the latest news in real time. By scheduling automatic background updates or using webhooks (if available), the app ensures that users always have access to the most current news headlines. This real-time capability increases user engagement and makes the application more dynamic and relevant.

4. Improved Content Management

The application allows users to personalize their news feed by selecting their preferred categories or topics. This customized experience ensures that users receive only the news they are interested in, reducing information overload. Additionally, the backend supports content filtering, keyword-based recommendations, and bookmarking for later reading.

5. Robust Performance

The application's backend has been optimized for speed and reliability. Efficient database queries, minimal server load, and effective caching mechanisms contribute to its high performance. Load testing showed that the application handles concurrent user access smoothly, with minimal latency or errors, ensuring a consistent and dependable service.

6. Scalability

The architecture of the application was designed with scalability in mind. It supports modular development, meaning new features (like push notifications, comment sections, or user discussions) can be added without disrupting the core functionalities. Additionally, the database schema and server resources can be expanded to accommodate a growing user base or increased data volume.

7. Security and Compliance

The system uses secure authentication mechanisms, including password hashing and optional multi-factor authentication, to protect user accounts. User data is encrypted during transmission using HTTPS and stored securely in the database. The project also follows basic data protection regulations and privacy standards, ensuring that user information is handled responsibly.

8. Positive User Feedback

User testing and early feedback sessions revealed a high level of user satisfaction. Test users praised the application’s design, ease of use, speed, and relevance of news content. Constructive feedback was used during development to improve features, and the final product reflects a user-driven design process. The overall response indicates that the project successfully addressed the needs of its target audience.

Chapter 4

Software Requirements

The development of the BuzzFeed news application requires a comprehensive and robust software environment that supports web scraping, data visualization, interactive GUI design, and backend database integration. The following are the core software components used in the project:

Front-End Technology

The front-end is responsible for how the application looks and interacts with the user. It involves Graphical User Interface (GUI) design, user inputs, navigation controls, and data visualization. The technologies used in the front-end are:

- **Python Version 3.12.1 (64-bit)**

  Python is chosen as the primary programming language due to its versatility, readability, and rich ecosystem of libraries. Python 3.12.1 provides enhanced performance, security, and long-term support. It enables efficient integration of backend processes, data scraping, and GUI design.

- **Tkinter & CustomTkinter**

  Tkinter, Python's built-in GUI toolkit, is utilized to build the graphical user interface. CustomTkinter enhances the appearance and adds modern widgets and themes, offering a sleek and user-friendly experience.

- **Matplotlib**

  Matplotlib is used for creating visual representations such as bar charts, line graphs, and pie charts. It adds interactivity and visual appeal to the application by presenting news statistics and trends effectively.

Back-End Technology

The back-end handles the logic, data processing, storage, and retrieval functions of the application. It supports user authentication, web scraping, data analysis, and storage.

- **BeautifulSoup (bs4)**

  BeautifulSoup is employed for web scraping purposes. It parses HTML and XML documents to extract relevant news content (e.g., titles, links, summaries) from various web sources, ensuring users receive the most recent updates.

- **Git Version 2.43.0**

  Git is employed for managing the source code, tracking changes, and enabling collaboration among team

members. It ensures that multiple developers can work simultaneously while maintaining version consistency and allowing easy rollback when necessary.

- **MySQL Version 8.0.36 for Win64 (x86_64)**

  MySQL is a widely used relational database management system known for its speed, reliability, and scalability. In this project, it is used to store:

  - User credentials and profile data
  - Bookmarked news articles
  - User search history and preferences
  - Scraped news article metadata

Integrated Development Environment (IDE)

- **Visual Studio Code (VS Code)**

  VS Code is the primary IDE for developing BuzzFeed. Its features such as intelligent code completion, integrated terminal, Git support, and debugging tools contribute to a seamless and productive development experience.

Chapter 5

Project Design

In your BuzzFeed news application, users begin their journey by exploring a wide array of news categories, allowing them to personalize their content experience based on their interests. Upon launching the app, users can either register as new members or log in if they already have an account, granting them access to a fully personalized news feed. Once inside, users can dive into Daily News updates, which provide real-time articles fetched through intelligent web scraping mechanisms. For more tailored exploration, the Category Selection Page allows users to browse news segmented into predefined topics such as Sports, Finance, Politics, World, Entertainment, and Sci/Tech. Those seeking specific information can use the Search News module, where entering keywords returns curated results instantly. This module also allows users to subscribe to daily email newsletters, ensuring they're updated on their favorite topics even while offline. The Infographics Page enriches the reading experience by offering visually engaging charts and graphs, helping users better understand data-driven news and trends. These infographics are powered by the Matplotlib library for rich, interactive visual content. Every user has access to their My Profile section, where they can manage personal information, review past searches in the My News area, and revisit saved content in the Bookmarks section. This ensures that their favorite articles and recently explored topics are always within reach.

5.1. System Architecture

1. **Dashboard**

   This is the main landing page of the BuzzFeed application. Upon launching the app, users are presented with a visually structured dashboard that offers direct access to the key modules of the system. It simplifies navigation and enhances user interaction by prominently displaying options such as Daily News, Category Selection, Infographics, Profile, and Search modules.

2. **Registration**

   New users must first register to use the app. The registration form collects essential user information such as name, email address, and password. This process is crucial for account creation, secure access, and delivering a personalized news experience.

3. **Login**

   After registration, users can securely log in using their credentials. This block acts as a gateway to all personalized features like bookmarks, email newsletters, and profile customization.

4. **News Web Server (to extract HTML/XML)**

   This block represents the source of real-time news data. It delivers HTML or XML content from various news portals, which is later processed for relevant article extraction.

5. **Web Scraper Extraction (using bs4)**

   BeautifulSoup (bs4), a powerful Python library, is used in this block to parse and extract meaningful content (e.g., headlines, summaries, article links) from the raw HTML/XML received from the News Web Server.

6. **Daily News**

   This module displays the most recent news articles fetched via the web scraper. It provides users with up-to-date content across all general topics without any manual selection or filtering.

7. **Category Selection**

   Users can explore news based on predefined categories like Sports, Finance, Politics, World, Entertainment, and Sci/Tech. This categorization helps users focus on areas of interest and makes content discovery seamless.

   - **Sports**: News related to sports events, updates, and athlete achievements.
   - **Finance**: Covers topics like stock market updates, banking, and economic news.
   - **Politics**: Includes political developments, government announcements, and election updates.
   - **World**: Presents international news, foreign affairs, and global happenings.
   - **Entertainment**: Focuses on movies, celebrity news, and pop culture.
   - **Sci/Tech**: Provides updates on scientific research, innovations, and technology news.

8. **Search News Module**

   This feature enables users to input custom keywords to find news articles that match their specific interests. It returns a filtered list of news items dynamically generated based on search input.

9. **Get Searched News by Email**

   Once a user searches for specific news, they can opt to receive similar articles daily via email. This feature ensures users are regularly updated on their chosen topics.

10. **Infographics**

    To enhance user experience, this module presents visual representations of data such as pie charts, bar graphs, and line charts. This is particularly useful for visual learners and helps in quickly understanding

news trends.

11. Graph (Using Matplotlib)

Matplotlib, a Python plotting library, is used in the backend to generate graphs for the Infographics module. These visual tools summarize data patterns from the news in an easy-to-understand format.

12. Profile

The user profile module allows users to manage their account settings and personalize their news feed.

- **Bookmarks**: This sub-module lets users save their favorite articles for easy access later.
- **My News**: Displays a history of previously searched news topics and keywords, enabling users to revisit or track their reading trends.

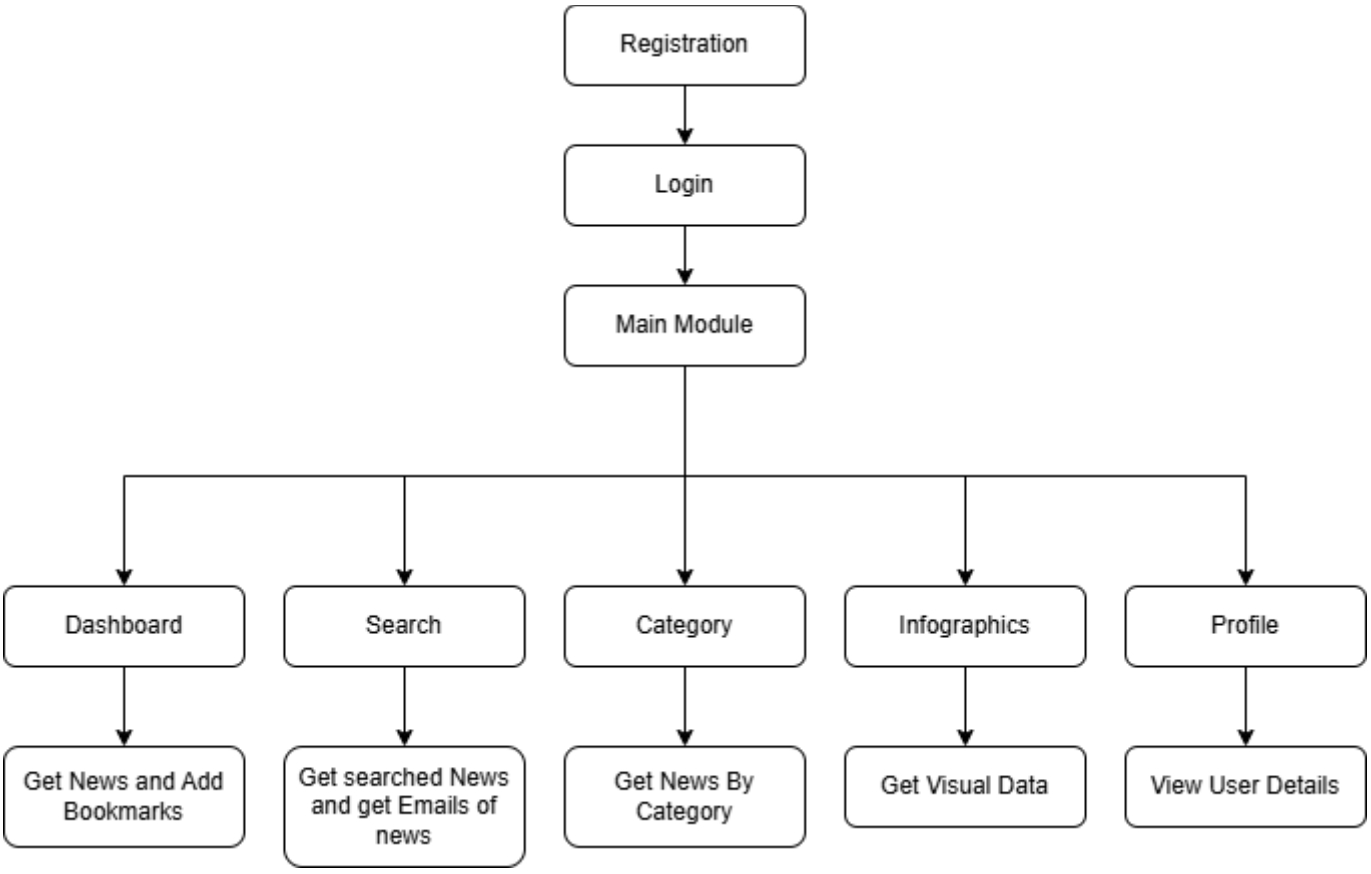

Figure 5.1 – System Architecture of BuzzFeed application

The software design for the news platform application involves the systematic process of defining the software's structure and behavior, ensuring it aligns with user requirements. This high-level design creates a blueprint for translating abstract requirements into specific data models and functionalities. The interface design encompasses how the platform communicates internally, with external news APIs, and with user behaviors, ensuring seamless interactions and a user-friendly experience.

Chapter 6

Project Scheduling

In the context of the BuzzFeed, scheduling plays a vital role in organizing and managing the development process as shown in Figure 6.1. The project schedule comprises a comprehensive list of milestones, tasks, and deliverables, serving as a roadmap for the project's execution. Project scheduling is a critical aspect of ensuring the timely and efficient completion of any software development endeavor. In the context of the described project, which involves creating a news application with personalized features and a graphical user interface (GUI), a structured scheduling approach can help manage tasks effectively and meet project milestones. Here's how the project scheduling could be outlined.

To visualize this schedule, a Gantt chart is employed as shown in Figure 6.1, providing a graphical representation of task durations, start and finish dates, and interactivity. Additionally, Gantt charts help illustrate the project's work breakdown structure and the relationships between activities, ensuring effective project management and progress tracking.

The timeline of the project was 12 weeks long where we work in following stages:

Group formation and Topic finalization: It's stage where we form a group of 4 members that have specific skillset to complete the any project and the scope and objectives where identified that ensures the application and outcome of the project. This stages took 4 days to complete from 6th January to 9th January. Here we don't faces any difficulties and finalized the topic for project

Identifying the functionalities: In this stage we discussed the final outcomes and the basic workflow of selected topic , the duration of stage was from 8th to 14th of January. Basic block diagram , flowchart and core functionalities are discussed among the group members with overlap of topic finalization stage.

Discussing the project topic with the help of paper prototype: This stage was heavily based on end user experience towards the project and the placements of UI component. A paper prototype is developed how has a basic UI structure of project. In this process we discovered that using sidebar for navigation would be better option. The duration of this stage was from 15th January to 10th of February.

Designing the Graphical User Interface: In this stage Sujal and Keval developed the basic pages for application using pythons tkinter library we just placed the UI elements in the stage the alignment and styling was not done. This stage ovelaped with above stage as it required continuous feedbacks and suggestion from group members. The duration of this stage was from 5th February to 10th of February.

Presentation 1: We presented our project application with basic UI, Database Connectivity and basic

messaging demonstration also it's scope and objectives to panel of project co-ordinator and mentor. We got suggestions and improvements calls from panel about UI improvements and bulk messaging features

Database Design: Here Keval and Sujal designed the schema of project which includes the required tables. The time period of this stage was from 6th February to 13th of February.

Database Connectivity: Here Aayush and Keval connected the schema of database to the UI pages of project. The stage was from 10th of February to 21th of February

Integration of all modules and Report Writing: As this was a final stage we implemented the suggested changes given in Presentation1. The report work is done simultaneously while developing the application for project.

We did meeting with project mentor to ensure all the objectives are achieved on 18th of March and implemented the suggestions in next days. The time duration of this stage was from 24th February to 1st April.

Presentation 2: Presented the Project with demonstration of project application.

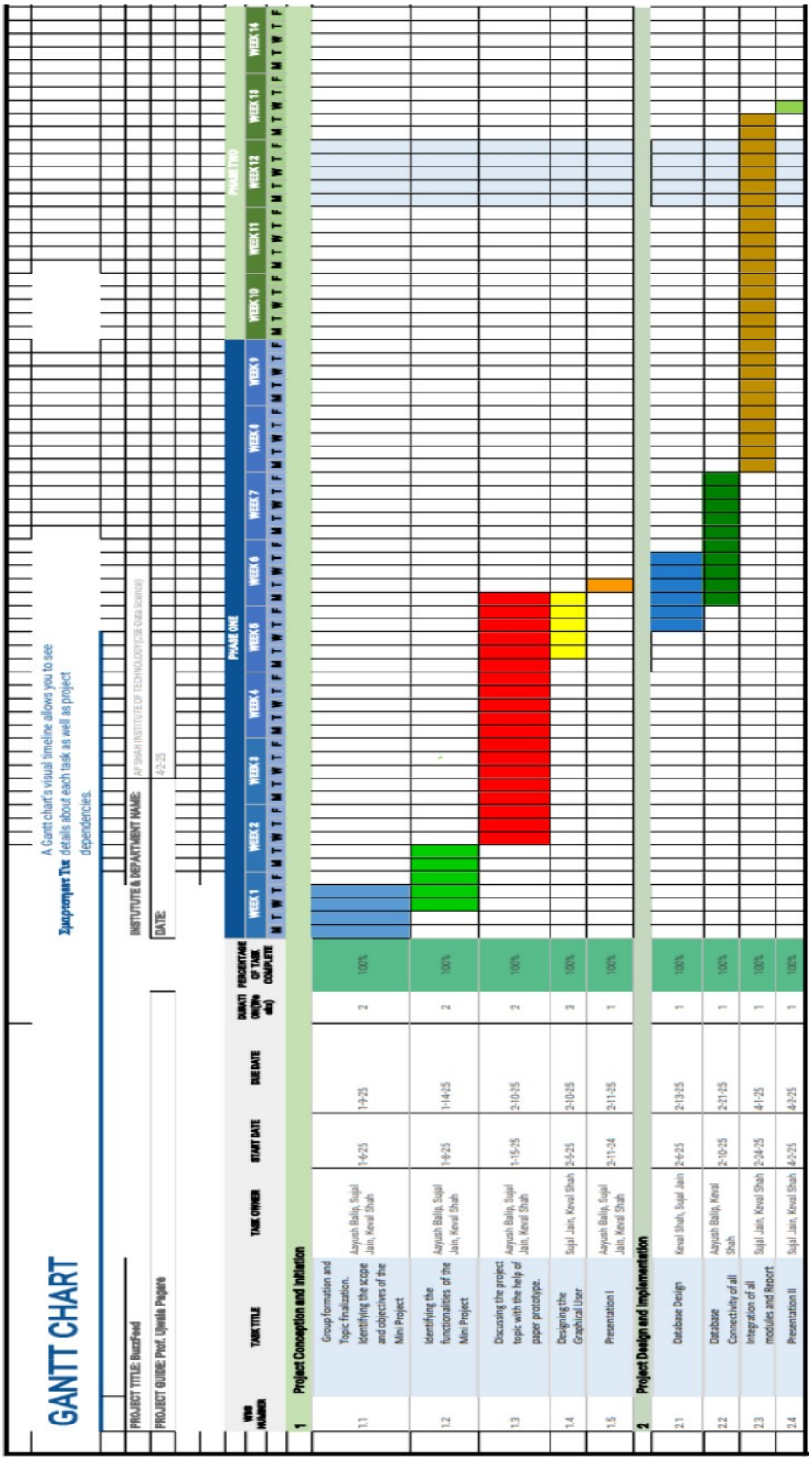

Figure 6.1-Gant Chart of BuzzFeed

Chapter 7

Results

The BuzzFeed application is designed to redefine the way users interact with and consume digital news. In an age where information overload is a real challenge, the platform offers a streamlined, intuitive, and personalized experience that empowers users to access the most relevant, timely, and engaging content based on their interests and preferences. By focusing on user-centric design and leveraging cutting-edge technologies, BuzzFeed aims to enhance information accessibility, comprehension, and engagement.

Here we will see the real time GUI Pages in full display and User end interface on how it would be visible to user and flow of the project. The visual flow illustrates how end-users interact with the app, highlighting each screen and its purpose.

1) Registration Page

As shown in Figure 5.1, illustrates the User Registration Page, where new users are required to input the Full Name, Email Address, Password. Upon submitting the form, the entered data is securely stored in the MySQL database for future login sessions. This ensures a personalized experience and allows for data persistence across user sessions.

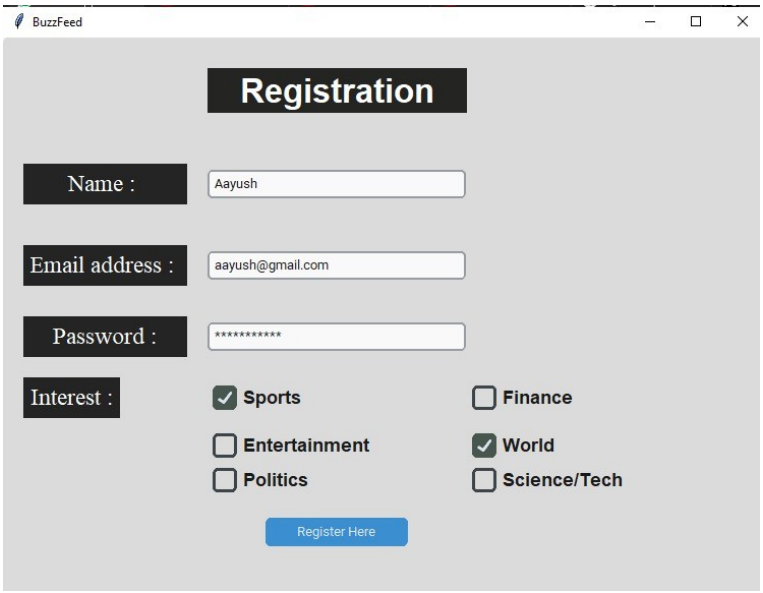

Figure 5.1 –Registration Page

2) Login Page

As shown in figure 5.2, the Login Page allows existing users to access their accounts by entering valid email and password credentials. The login system includes form validation, ensuring that incorrect or incomplete entries prompt user-friendly error messages. A successful login redirects users to the Dashboard.

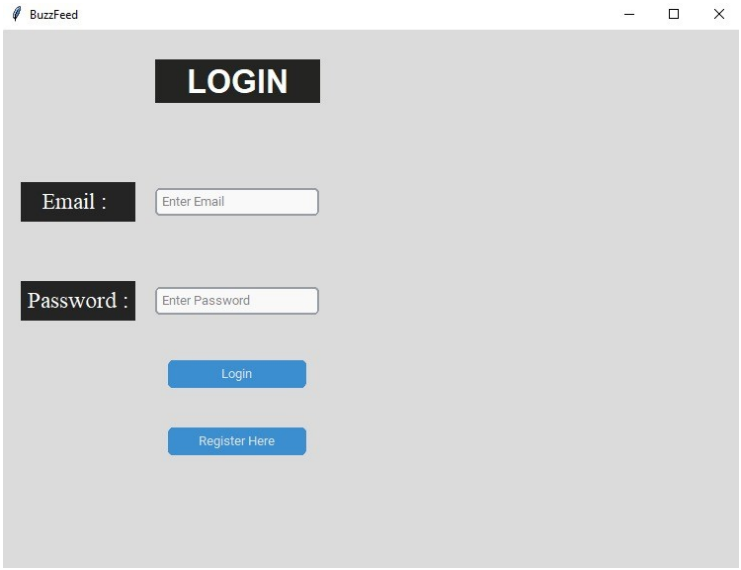

Figure 5.2 -Login Page

3) Dashboard Page

In Figure 5.1, the Dashboard is the landing page post-login, offering a clean, user-friendly interface with four main action buttons: Search News, News by Category, Infographics, My Profile

These buttons are located on the left sidebar, allowing users to swiftly navigate through the application. A first-time user typically starts with the Daily News section, which displays up-to-date news articles fetched via web scraping. This centralized dashboard simplifies access to all essential functionalities from one place.

Figure 5.3 –Main Dashboard

4) Bookmark Section

As shown in figure 5.4, the Bookmark button is prominently positioned in the top-right corner of each article display. Users can click this icon to save news articles of interest. These saved articles are automatically added to the Bookmarks tab inside the My Profile section. This feature is ideal for readers who wish to revisit stories later without re-searching.

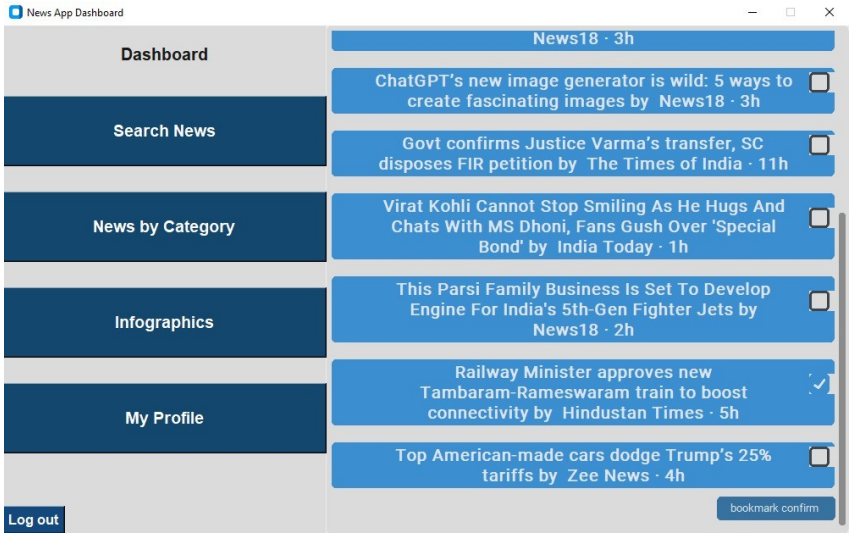

Figure 5.4 –Bookmark section

5) News Category Page

As shown in Figure 5.5 displays the News Category interface, where users can choose from six predefined news categories as Sports, Finance, Politics, World, Entertainment, Sci/Tech. By selecting a category, the application fetches real-time news articles under that theme. This segmentation helps users focus on their areas of interest without the clutter of unrelated news.

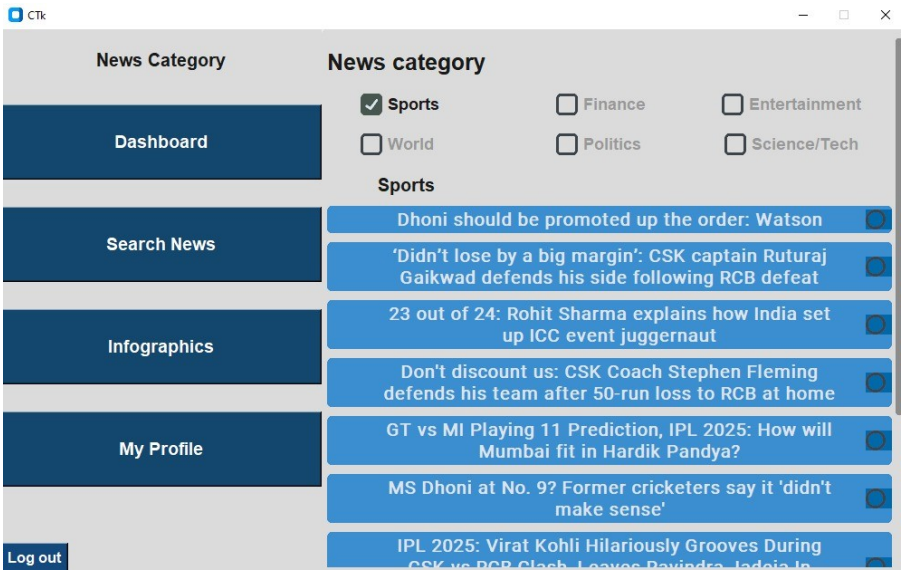

Figure 5.5 –News Category

6) Search News

As shown in Figure 5.6, introduces the Search Functionality, which allows users to input specific keywords or topics in a search bar. The app dynamically displays articles related to the searched topic, offering a customized news feed. This module enhances the app's interactivity and personalization.

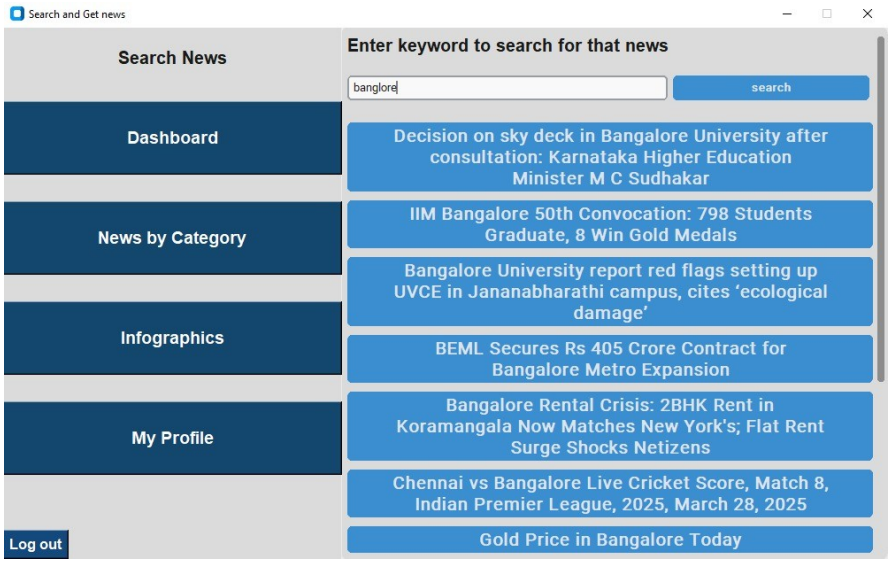

Figure 5.6 –Search News

7) Infographics

As shown in Figure 5.7, the Infographics Module presents news data in a graphical format using charts and visualizations powered by Matplotlib. This page provides Trend graphs, Bar graphs. Such visuals make it easier for users to comprehend complex datasets, especially in categories like Finance or World News.

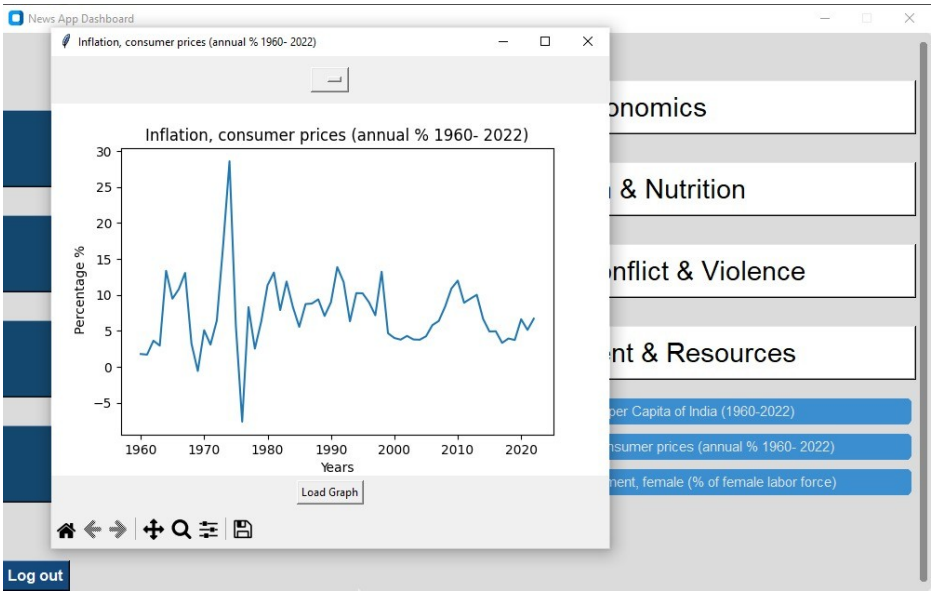

Figure5.7–Infographics (Information page)

8) My Profile

As shown in figure5.8, Figure 5.8 showcases the user's personalized profile interface, which contains Personal Info (name, email, preferences)

1. My News: Displays recent searches and allows quick access to related news
2. Bookmarks: Lists all bookmarked articles saved by the user

This page empowers users to manage and personalize their news consumption experience with ease.

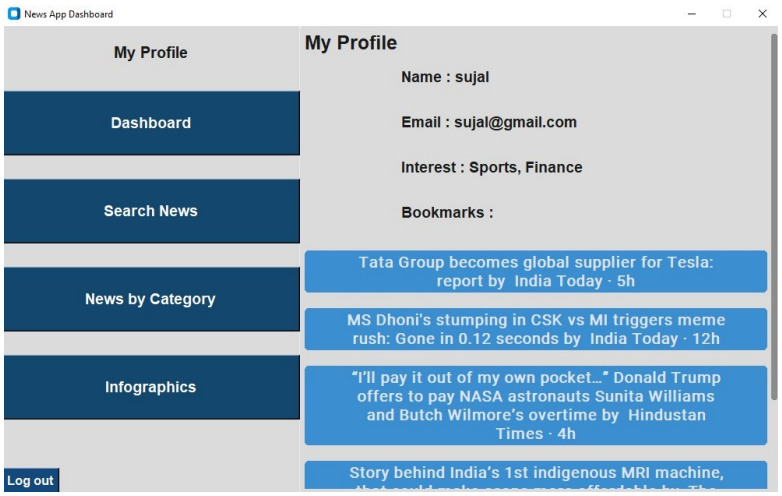

Figure 5.8 –My Profile

9) My News

As shown in Figure 5.9 demonstrates a specific example where a user has searched for “IPL” (Indian Premier League). The My News section stores this keyword and fetches all latest articles related to IPL. Users can also subscribe to updates on these topics via email.

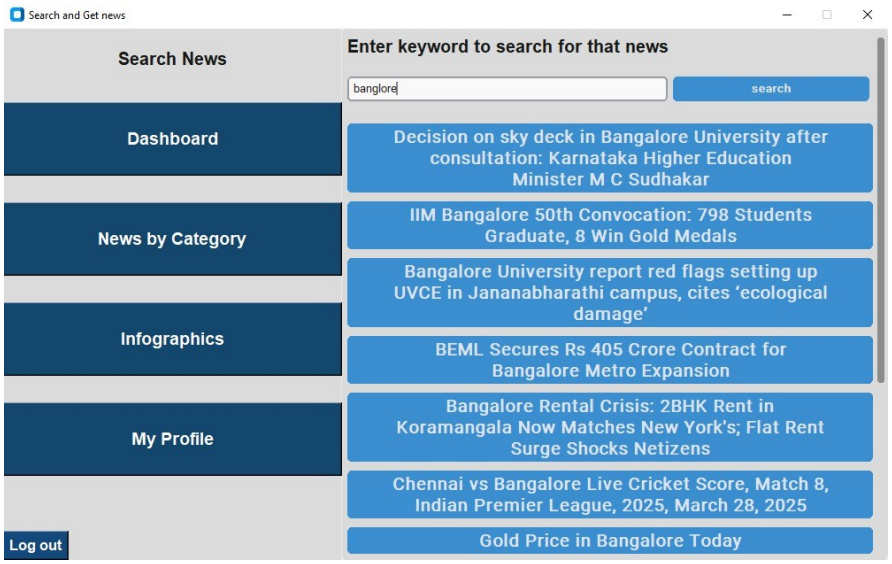

Figure 5.9 -My News(IPL)

Chapter 8

Conclusion

The creation and implementation of the "BuzzFeed" news application marks a pivotal step forward in the evolution of Python-based news platforms. In a digital landscape that is increasingly saturated with information, users are more discerning than ever in seeking news experiences that are not only accurate and real-time but also intuitively designed and personally relevant. BuzzFeed was envisioned as a comprehensive remedy to the existing shortcomings within this space—where most news applications fall short in areas such as personalization, interface responsiveness, scalability, and dynamic content presentation.

This project was carefully designed to bridge these gaps through an integrated and user-first approach, harnessing the flexibility and power of Python’s ecosystem. Leveraging Tkinter and CustomTkinter, the application's graphical user interface is crafted to be aesthetically pleasing and simple to navigate. Unlike many traditional news apps with static layouts and limited interactivity, BuzzFeed enables a fluid, engaging, and visually cohesive user experience. The choice of BeautifulSoup (bs4) as the web scraping tool allows the app to pull news articles from a variety of trusted sources in real time, ensuring that users receive up-to-date headlines and stories across a range of topics.

The core innovation lies not only in gathering information but in presenting it meaningfully. By integrating Matplotlib, the app introduces an infographics module that visually synthesizes complex datasets—news trends, statistics, and analytics—into digestible formats. Interactive bar graphs, pie charts, and line charts are not merely decorative but serve as essential tools for enhancing user understanding and offering deeper insights. This feature sets BuzzFeed apart by transforming raw data into engaging narratives, enriching the reader’s understanding of unfolding events.

From a functional standpoint, BuzzFeed offers a feature-rich environment tailored to individual needs. Whether a user wants to browse news by predefined categories, search for specific topics, or bookmark important articles for later, the platform caters to diverse user habits. The “My Profile” section personalizes the journey even further by storing preferences, recent searches, and saved content, creating a sense of continuity every time a user logs in. Additionally, the email newsletter system, curated based on keywords and preferences, ensures that users remain connected to their interests even when they’re not actively using the app—bridging the gap between accessibility and personalization.

Importantly, BuzzFeed has been built with scalability and future enhancement in mind. Its modular architecture allows for the integration of additional features without compromising performance. For

instance, potential future updates could include machine learning models to recommend news based on reading history, sentiment analysis of trending topics, or language translation features to expand accessibility to non-English speaking audiences. Moreover, the foundational backend structure, supported by MySQL, ensures that user data and news metadata are handled efficiently, securely, and reliably.

From a development perspective, the use of Git for version control and VS Code as the IDE allowed for streamlined collaboration, robust debugging, and rapid iteration. These tools facilitated smooth workflow cycles, ensuring the application remained aligned with its initial goals while being flexible enough to incorporate iterative feedback and evolving feature sets.

In a broader context, the success of BuzzFeed reflects a growing need for user-empowered digital media tools. In an age where misinformation, content fatigue, and information overload are pressing issues, applications like BuzzFeed represent a step towards more intentional, transparent, and empowering news experiences. It transforms users from passive content consumers into active participants, allowing them to shape their news intake based on real interests and needs.

In conclusion, BuzzFeed is more than a software application—it is a complete digital solution that reimagines how news can be discovered, consumed, and understood. By combining technical innovation with thoughtful design and user engagement strategies, it sets a new standard for what news platforms can achieve. It stands as a testament to the power of interdisciplinary thinking, drawing from computer science, journalism, user experience design, and data visualization to build a tool that is both technically sound and socially impactful. As the digital media landscape continues to evolve, applications like BuzzFeed will play an increasingly vital role in shaping how information flows in modern society.

References:

[1] Haixia Lv, “Design and Implementation of Domestic News Collection System Based Python,” Shandong University of Finance and Economics, 2021.

[2] Vidhi Singrodia, Amity University, Newtown, Kolkata, “A Review on Web Scraping and Its Applications,” International Conference on Computer Communication and Informatics (ICCCI), 2019.

[3] Moaiad Ahmad Khder, “Web Scraping or Web Crawling: State of Art, Techniques, Approaches and Application,” *Int. J. Advance Soft Compu. Appl.*, Vol. 13, No. 3, Applied Science University, Bahrain.