A

PROJECT REPORT

On

NEWS AGGREGATOR

Submitted in partial fulfilment of the
requirements for the award of the degrees

of

BACHELOR OF TECHNOLOGY

in

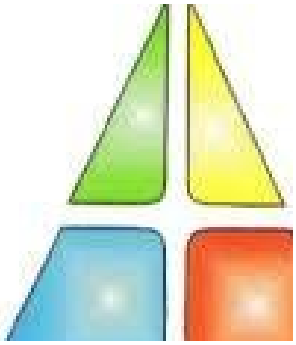
INFORMATION TECHNOLOGY

Submitted by:

Vaibhav Kumar Sahu, Suryanarayan Singh & Amritansh Tripathi

Guided by:

Dr Ani Thomas
Head Of Department, Information Technology Branch



BHILAI INSTITUTE OF TECHNOLOGY DURG
DEPARTMENT OF INFORMATION TECHNOLOGY

UGC Autonomous Institution

(Affiliated to CSVTU, Approved by AICTE, NBA &NAAC ACCREDIATED)

DURG– 491001, CHHATTISGARH, INDIA
www.bitdurg.ac.in

SESSION: 2023-24

CANDIDATE'S DECLARATION

We hereby declare that the project entitled **"News Aggregator"** submitted in partial fulfilment for the award of the degree of Bachelor of Technology in Information Technology completed under the supervision of **Dr Ani Thomas, Head Of Department, Information Technology Branch** BIT DURG is an authentic work.

Further, I/we declare that I/we have not submitted this work for the award of any other degree elsewhere.

Vaibhav Kumar Sahu

Suryanarayan Singh

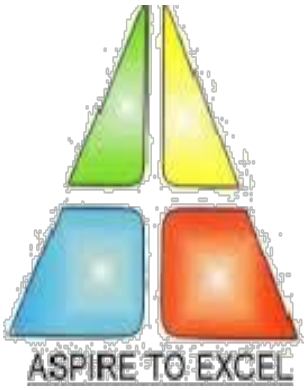
Amritansh Tripathi

Signature and name of the student(s) with date

CERTIFICATE by PROJECT Guide(s)

It is certified that the above statement made by the students is correct to the best of my/our knowledge.

Signature of BTP Guide(s) with dates and their designation



BHILAI INSTITUTE OF TECHNOLOGY DURG

DEPARTMENT OF INFORMATION TECHNOLOGY

UGC Autonomous Institution

(Affiliated to CSVTU, Approved by AICTE, NBA &NAAC ACCREDIATED)

DURG– 491001, CHHATTISGARH, INDIA

Department of Information Technology

CERTIFICATE BY THE EXAMINERS

This is to certify that the Major Project work entitled "**News Aggregator**" is carried out by **Vaibhav Kumar Sahu (300103321006), Suryanarayan Singh (300103321036), Amritansh Tripathi (300103321042)** in partial fulfilment for the award of degree of **Bachelor of Technology** in **Information Technology**, **Bhilai Institute Of Technology**, **Durg** during the academic year 2023-2024.

Mr Amrendra Kumar Singh External Examiner Prof. Dr. Ani Thomas

Internal Guide HOD

ACKNOWLEDGEMENTS

We wish to acknowledge with a deep sense of hearty gratitude and indebtedness to **Mr Amrendra Kumar Singh** Information Technology, who gave us this opportunity to experience project work & his valuable suggestion during this project have been invaluable.

We take this opportunity to voice & record our sincerest gratefulness towards our esteem Supervisor **Dr Ani Thomas** under whose able guidance the project work has been brought to completion.

Our heart leaps up in thankfulness for his benevolence & time to time help, valuable suggestions, constructive criticism & active interest in the successful completion of this project work.

We are also thankful to all our honourable teachers of the Information Technology Department and our parents whose valuable support helped us and kept us motivated all through.

Vaibhav Kumar Sahu

Suryanarayan Singh

Amritansh Tripathi

B.Tech. III Year
Discipline of Information Technology
BIT DURG

TABLE OF CONTENTS

ABSTRACT

This document describes the creation of a news aggregator application that leverages Java, JavaFX, and MySQL. The primary goal is to offer users a centralized platform for conveniently accessing news content from multiple sources. The document covers the project's architecture, implementation, testing, and potential future improvements, highlighting its core features and the user experience it provides.

In a fast-paced world, news aggregators play a vital role in helping users access diverse news content from different outlets. This project's use of Java, JavaFX, and MySQL enables the creation of a user-friendly and efficient news aggregation platform. The document details the system's components, code snippets, testing methodologies, and suggests areas for future enhancement to provide a more comprehensive and engaging news browsing experience.

INTRODUCTION

Overview of the Project:

The news aggregator project is designed to gather, organize, and display news content from various sources into a single platform. This application aims to provide users with a centralized location to access and view diversified news articles, enhancing user convenience and information accessibility.

Objectives and Scope:

The primary objective of this project is to develop a functional news aggregator utilizing Java programming language, JavaFX for the user interface, and MySQL for database management. The project focuses on integrating these technologies to create a seamless and user-friendly news aggregation platform. The scope involves the aggregation of news articles from multiple sources, categorizing them, and presenting them in an organized manner to the users.

Importance of News Aggregators:

News aggregators play a crucial role in today's fast-paced world, where information is constantly being generated from various sources. They offer users the ability to access news content from different outlets in one location, saving time and providing a comprehensive view of current events. Additionally, they facilitate personalized content delivery, enhancing user experience and engagement.

Background Information on Java, JavaFX, MySQL, and JDBC:

Java: Java is a widely used programming language known for its platform independence and object-oriented structure. Its robustness and versatility make it suitable for diverse applications.
JavaFX: JavaFX is a software platform used for creating and delivering desktop applications, particularly emphasizing rich user interfaces.
MySQL: MySQL is an open-source relational database management system used for efficiently managing and querying structured data.
JDBC (Java Database Connectivity): JDBC is an API that provides Java methods to interact with databases. It enables connectivity between Java applications and various databases.

PROBLEM IDENTIFICATION

Problem Statement:

In the contemporary digital landscape, the abundance of news sources presents a challenge for users to efficiently access diverse and relevant information. Navigating through multiple websites or applications to stay updated on various news categories can be time-consuming and overwhelming. Existing news platforms may lack user-friendly interfaces or fail to provide a consolidated view of news content. Moreover, the authenticity and timeliness of news articles can be compromised. To address these challenges, there is a need for a robust news aggregator application that seamlessly integrates diverse news sources, offers a user-friendly interface, ensures real-time updates, and enhances the overall user experience. This project aims to fill this gap by developing a news aggregator using Java, JavaFX, and MySQL, providing a centralized platform for users to conveniently access and navigate through a variety of news categories.

METHODOLOGY

Project Design and Architecture:

System Architecture and Components:

The news aggregator system is comprised of several key components that collaborate to deliver a comprehensive news browsing experience.

User Interface (UI):

Utilizes JavaFX to create the graphical interface for user interaction.
Scenes such as the login/register interface and news category selection screen are developed to seamlessly guide users through the application.

Backend and Data Management:

Java serves as the backend language handling data aggregation, user interactions, and external source connectivity.
The NewsDatabase class acts as a bridge between the application and the MySQL database, retrieving news content categorized by type.

Database Management:

MySQL is employed as the relational database system for storing news articles.
Data is organized into tables that categorize news based on type, enabling efficient retrieval.

Integration of Java, JavaFX, and MySQL:

The integration is fundamental to the functionality of the news aggregator. JavaFX facilitates the creation of a user-friendly interface, seamlessly connecting with the backend Java code. Java uses JDBC (Java Database Connectivity) to communicate with the MySQL database, allowing the storage and retrieval of news content.

Database Schema and Its Relevance:

The database schema represents the organization of the MySQL database. It is crucial for efficiently managing and storing news content.

Tables:

News categories: Store different types of news (sports, international, technology, health).
News content: Store actual news articles or their references, categorized under each type.
Relevance:
Provides a structured framework for storing, retrieving, and displaying news content to uers based on their preferences and selected categories.
The effective design and interaction between these components form the backbone of the news aggregator, ensuring a seamless and user-centric experience. The next section will delve into the implementation details, highlighting code snippets and the methodology employed for content aggregation

Implementation：

Java Code Snippets for News Aggregation:

The provided Java code implements a simple news aggregator application using JavaFX for the user interface and JDBC for database connectivity. The App class serves as the main structure for managing scenes and user interactions.

Login and News Category Scenes:

createLoginScene() and createNewsCategoryScene() functions initialize the scenes for user login and news category selection.

JavaFX layout components like VBox, BorderPane, Label, and Button are used to structure and design the interface.

Displaying News:

displayNews() method updates the label in the news category scene with the selected category's news content.

Typing animation is simulated using createTypingAnimation() to display news content letter by letter.

User Interface Development with JavaFX:

Login Scene:

In the login scene, users are prompted to input their name via a text field. Upon clicking the "Login" button, the user transitions to the news category selection scene.

News Category Scene:

This scene presents various category buttons such as Sports, International, Technology, and Health news. Upon selection, the news content is displayed dynamically.

Typing Animation:

News content is displayed with a typing animation effect, providing an engaging user experience.

Database Connectivity using JDBC:

The NewsDatabase class manages database connections and queries to retrieve news content based on category. The getNews(String category) method retrieves news content from the MySQL database using JDBC.

Connection Establishment:

Connection parameters such as URL, username, and password are defined.
The getConnection() method establishes a connection to the MySQL database.

Retrieving News Content

SQL queries are executed to retrieve news content based on the selected category. This content is then returned to the application for display.
Algorithms or Methodologies for Content Aggregation:
The application utilizes a simple approach to aggregate and display news content. Content is retrieved from the MySQL database based on the selected category and is dynamically displayed using a typing animation.

Challenges Faced and Resolutions:

Database Connectivity:

Ensuring a secure and efficient database connection while handling exceptions related to connection establishment and query execution.

Dynamic Content Display:

Implementing a smooth and visually appealing content display mechanism while ensuring an engaging user experience.

The provided code snippets demonstrate the implementation of a basic news aggregator application using Java, JavaFX, and MySQL database connectivity. However, further enhancements and error handling might be necessary for a complete and robust system. The next section will cover the detailed functionalities of the news aggregator and how different features were implemented using the JavaFX UI and backend Java code.

Features and Functionality:

The provided code showcases a basic news aggregator application developed using Java, JavaFX for the user interface, and MySQL for storing and retrieving news content. The functionalities focus on user interactions, category-based news display, and a typing animation effect. Here's an outline for the detailed features and functionality overview:

Detailed Functionalities of the News Aggregator:

1. User Login and News Category Selection:

- Login Scene:

  Users can input their name in the text field and proceed by clicking the "Login" button, transitioning to the news category selection scene.

- News Category Scene:

  Presents various category buttons such as Sports, International, Technology, and Health news.

2. News Display and Typing Animation:

- Content Display:

  Upon selecting a news category, the corresponding news content is displayed in the center panel.

- Typing Animation:

  An engaging feature where news content is displayed letter-by-letter using a typing animation effect for an interactive user experience.

3. Category-based News Retrieval from Database:

- Integration with Database::

  The NewsDatabase class uses JDBC to query a MySQL database and fetch news content based on the selected category.

User Interactions and System Responses:

1. User Interface:

- Provides a clean and simple user interface utilizing JavaFX layout components.
- Enables intuitive interactions by presenting category buttons for news selection.

2. Content Display and Animation:

- Dynamic content display offering news in various categories.
- The typing animation engages users by gradually revealing the news content in an animated manner.

Implementation Using Java, JavaFX, and MySQL:

1. Java and JavaFX:

- Utilizes JavaFX to design the UI, manage scenes, and handle user interactions.
- Implements event-driven actions for button clicks to switch scenes and display news.

2. MySQL Database Connectivity:

- Connects to a MySQL database using JDBC for retrieving news content.
- Executes SQL queries to fetch news articles based on the selected category.

The aggregation and retrieval of news content are handled through the NewsDatabase class, allowing users to view various news categories dynamically.

The seamless integration of Java, JavaFX, and MySQL results in a functional news aggregator with basic yet engaging functionalities. However, the system might benefit from additional features like user authentication, error handling, and a more extensive database structure for a comprehensive news browsing experience.

This section highlights the core functionalities and system responses provided by the application. The subsequent section will cover testing methodologies, performance metrics, and an evaluation of the system's performance and accuracy.

Testing and Results:

The testing phase of the news aggregator application involves various methodologies and metrics to evaluate its performance and accuracy. Here is an outline for this section:

Testing Methodologies Employed:

1. Unit Testing:

- Validating individual components of the application like button actions, database connectivity, and text animations to ensure they function as expected.

2. Integration Testing:

- Testing the interaction between different modules like JavaFX UI components, Java logic, and the MySQL database.

3. Usability Testing:

- Involve potential end-users to test the user-friendliness and ease of navigation within the application.

Performance Metrics and Test Cases:

1. User Interaction and Responsiveness:

- Measure the time taken for the application to respond to user actions such as button clicks and scene transitions.

2. Database Connectivity Performance:

- Evaluate the speed of fetching news content from the MySQL database for different categories.

3. Typing Animation Accuracy:

- Check the accuracy and smoothness of the typing animation effect in displaying news content.

Evaluation of System's Performance and Accuracy:

1. Functional Validation:

- Verify that the system accurately displays news content based on the selected category.

2. Error Handling:

- Verify that the system accurately displays news content based on the selected category.

3. Scalability:

- Verify that the system accurately displays news content based on the selected category.

4. User Experience:

- Verify that the system accurately displays news content based on the selected category.

Result Analysis:

1. Challenges Faced:

- Outline challenges encountered during testing, like unexpected UI behavior, database connection issues, or performance bottlenecks.

2. Resolutions and Enhancements:

- Detail the resolutions applied to overcome challenges, whether through code improvements, error handling, or database optimization.

3. Areas for Future Improvement:

- Identify aspects that could be further improved, such as enhanced user authentication, dynamic content updates, or additional features for a more comprehensive user experience.

The testing phase aims to ensure the reliability, responsiveness, and accuracy of the news aggregator application while identifying areas for enhancement. It is essential to validate the application under various scenarios to provide a robust and user-friendly news browsing experience.

Conclusion and Future Work

Summary of the Project's Outcomes:

The News Aggregator project aimed to create a Java application using JavaFX for the user interface, JDBC for database connectivity, and MySQL for storing news data. The application successfully provides users with different categories of news, allowing them to browse through Sports, International, Technology, and Health news. Each category presents a typing animation feature to display news content dynamically.

Achievements and Limitations:

Achievements:

Functional Validation:

- User-Friendly Interface: The user interface offers simplicity and ease of navigation for selecting and viewing news categories.
- Dynamic Content Display: The typing animation creates an engaging experience by displaying news content character-by-character.
- Database Connectivity: Successful integration with a MySQL database to fetch news data based on user category selection.

Limitations:

- Static Sample Data: The project currently uses hardcoded sample news for different categories, limiting the variety and authenticity of the news content.
- Basic User Interaction: The application lacks user authentication and personalization features.

Future Enhancements and Possible Expansions:

1. Authentic Data Integration:

Integrate with real-time news APIs to fetch genuine and updated news content for each category.

2. User Authentication and Preferences:

Implement user authentication to allow personalization, like saving preferences or bookmarking news

articles.

3. Improved User Experience:
Enhance the user interface by introducing graphics, images, or video content for a richer news browsing experience.

4. Content Filtering and Search:
Implement a search functionality and filtering options for more precise news category selection.

5. Mobile Application Development:
Expand the project by developing a mobile application to reach a broader audience.

6. Performance Optimization:
Optimize database queries and UI rendering to enhance application performance and responsiveness.

Final Thoughts:

The News Aggregator project provides a solid foundation for a news browsing application. While it currently serves the purpose of displaying news in different categories, there is substantial potential for expansion and enhancement to deliver a more comprehensive, user-friendly, and authentic news browsing experience. With further development and feature additions, it could evolve into a powerful and widely-used news platform.

Code Snippets

JavaFX User Interface Creation:

The code creates a JavaFX application with two scenes: the login scene and the news category scene. It sets up the UI components for user interaction.

```
@Override
public void start(Stage primaryStage) {
    createLoginScene(primaryStage);
    createNewsCategoryScene(primaryStage);
    primaryStage.setScene(loginScene);
    primaryStage.setTitle("News Aggregator");
    primaryStage.show();
}

private void createLoginScene(Stage primaryStage) {
    BorderPane loginRegisterRoot = new BorderPane();
        loginScene = new Scene(loginRegisterRoot, 600, 400);

        VBox loginRegisterMenu = new VBox(10);
        loginRegisterMenu.setPadding(new Insets(10));
        loginRegisterMenu.setStyle("-fx-background-color: #EDEDED");

        TextField usernameTextField = new TextField();
        usernameTextField.setPromptText("Enter your name");

        Button loginButton = new Button("Login");

        loginRegisterMenu.getChildren().addAll(usernameTextField, loginButton);
        loginRegisterRoot.setCenter(loginRegisterMenu);

        VBox loginRegisterContent = new VBox(10);
        loginRegisterContent.setPadding(new Insets(10));
```

```
        loginRegisterContent.getChildren().add(new Label("Welcome to News Aggregator!"));

        loginRegisterRoot.setLeft(loginRegisterContent);

        // Button actions
        loginButton.setOnAction(event -> switchToNewsCategoryPage(primaryStage));
}

private void createNewsCategoryScene(Stage primaryStage) {
   BorderPane newsCategoryRoot = new BorderPane();
        newsCategoryScene = new Scene(newsCategoryRoot, 600, 400);

        VBox newsCategoryMenu = new VBox(10);
        newsCategoryMenu.setPadding(new Insets(10));
        newsCategoryMenu.setStyle("-fx-background-color: #EDEDED");

        Button sportsNewsButton = new Button("Sports News");
        Button internationalNewsButton = new Button("International News");
        Button technologyButton = new Button("Technology News");
        Button healthButton = new Button("Health News");
Button logoutButton = new Button("Logout"); // Added the "Logout" button

        newsCategoryMenu.getChildren().addAll(sportsNewsButton, internationalNewsButton,
technologyButton, healthButton, logoutButton);
        newsCategoryRoot.setLeft(newsCategoryMenu);

        // Create a label for displaying news content
        Label newsContentLabel = new Label("Select a category to view news.");
        newsContentLabel.setWrapText(true);
        newsCategoryRoot.setCenter(newsContentLabel);

        Define sample news content
        String sportsNews = getSportsNews();
```

```
        String internationalNews = getInternationalNews();
        String technologyNews = getTechnologyNews();
        String healthNews = getHealthNews();

        // Button actions
        sportsNewsButton.setOnAction(event -> displayNews(newsContentLabel, "Sports News",
sportsNews));
        internationalNewsButton.setOnAction(event -> displayNews(newsContentLabel, "International News",
internationalNews));
        technologyButton.setOnAction(event -> displayNews(newsContentLabel, "Technology News",
technologyNews));
        healthButton.setOnAction(event -> displayNews(newsContentLabel, "Health News", healthNews));}
```

Displaying News Content with Typing Animation:

The code initializes a typing animation for displaying news content character-by-character when a news category button is clicked.

```
private void displayNews(Label newsContentLabel, String category, String news) {
    newsContentLabel.setText(category + "\n\n");
    createTypingAnimation(news, newsContentLabel).play();
}

private Timeline createTypingAnimation(String text, Label label) {
    Timeline timeline = new Timeline();
    KeyFrame[] keyFrames = new KeyFrame[text.length() + 1];

    for (int i = 0; i <= text.length(); i++) {
        final int index = i;
        KeyFrame keyFrame = new KeyFrame(
            Duration.millis(50 * i),
            event -> label.setText(text.substring(0, index))
        );
        keyFrames[i] = keyFrame;
```

```
    }
    timeline.getKeyFrames().addAll(keyFrames);
    return timeline;
}
```

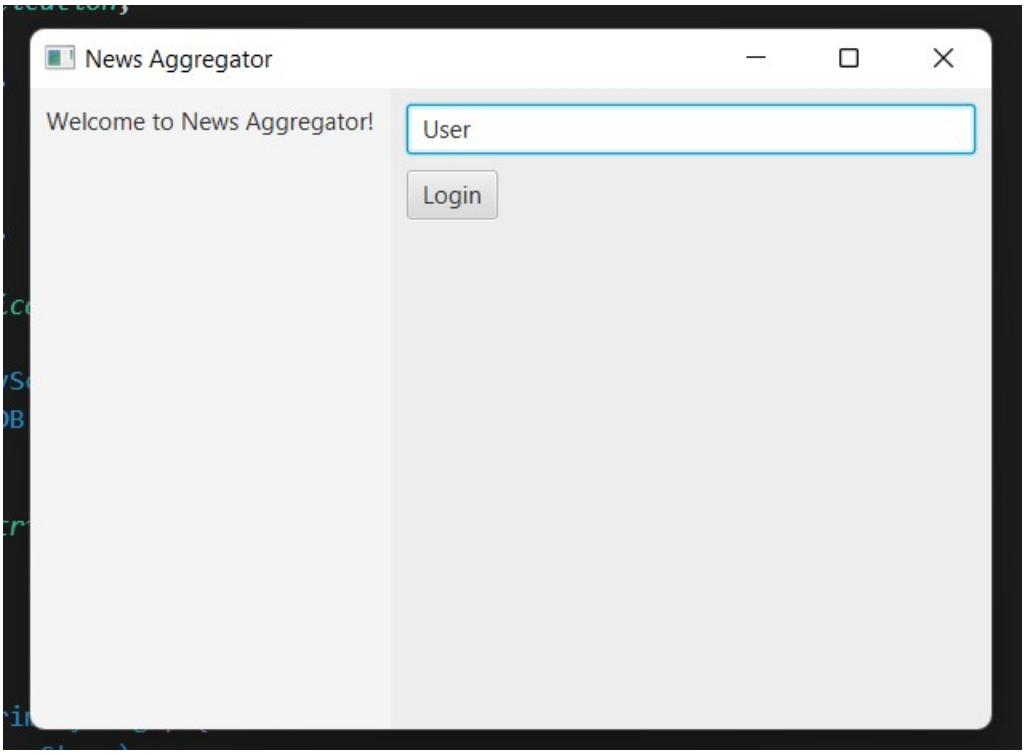
JDBC and newsDB class:

The newsDB class handles the retrieval of news content using JDBC to communicate with a MySQL database.

```
public class newsDB {
    private static final String url = "jdbc:mysql://localhost:3306/newsDB";
    private static final String userName = "root";
    private static final String password = "babu";

    public static String getNews(String category) {
        String news = "";
        try (Connection connection = DriverManager.getConnection(url, userName, password);
             Statement statement = connection.createStatement()) {
            String query = "SELECT content FROM News WHERE category = '" + category + "'";
            try (ResultSet resultSet = statement.executeQuery(query)) {
                if (resultSet.next()) {
                    news = resultSet.getString("content");
                }
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
        return news;
    }
}
```

Screenshots of the User Interface:



BIBLOGRAPHY

1.	JAVA Language	https://docs.oracle.com/javase/8/docs/technotes/guides/language/index.html
2.	JAVA FX (JFX)	https://docs.oracle.com/javafx/2/
3.	Java Database Connectivity (JDBC)	https://docs.oracle.com/javase/8/docs/technotes/guides/jdbc/
4.	MySQL	https://dev.mysql.com/doc/