COLLEGE CODE : 9623

COLLEGE NAME : Amrita College of Engineering and Technology

DEPARTMENT : Computer Science and Engineering

STUDENT NM-ID : F2DFE2525B0AB62BB764F5B255AC349B

ROLLNO : 962323104051

DATE :19-10-2025

COMPLETED THE PROJECT NAME AS PHASE 5

TECHNOLOGY PROJECT NAME: News Feed Application

SUBMITTED BY,

NAME: KESAVINI V

MOBILE NO: 93428 65728

PROJECT PHASE 5: News Feed Application

Project Overview

The News Feed App with Database is a complete full-stack application built using the MERN stack (MongoDB, Express, React, Node.js). It serves as a localized news aggregator where articles are managed entirely within a persistent MongoDB database. Core features include a dynamically filtered News Feed, persistent Bookmark management, and a dedicated Statistics dashboard showcasing database metrics. The backend employs a RESTful API for data handling, and the frontend ensures a responsive user experience with real-time updates.

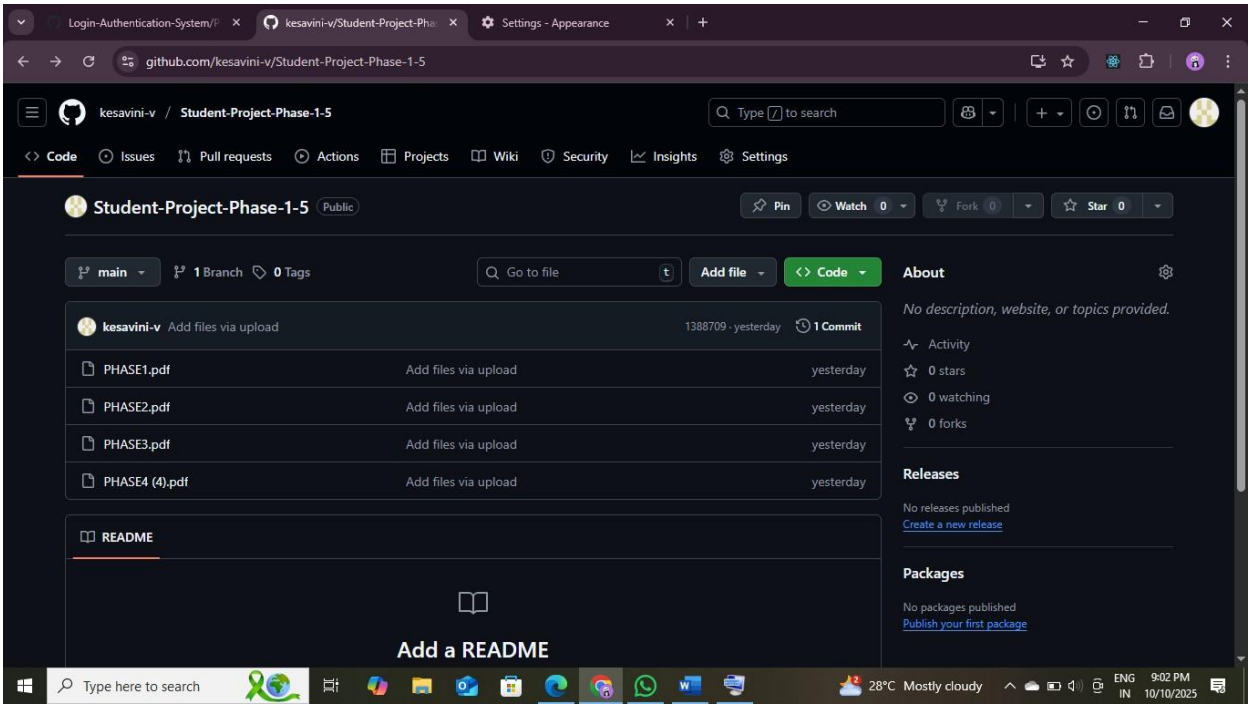
Project Report

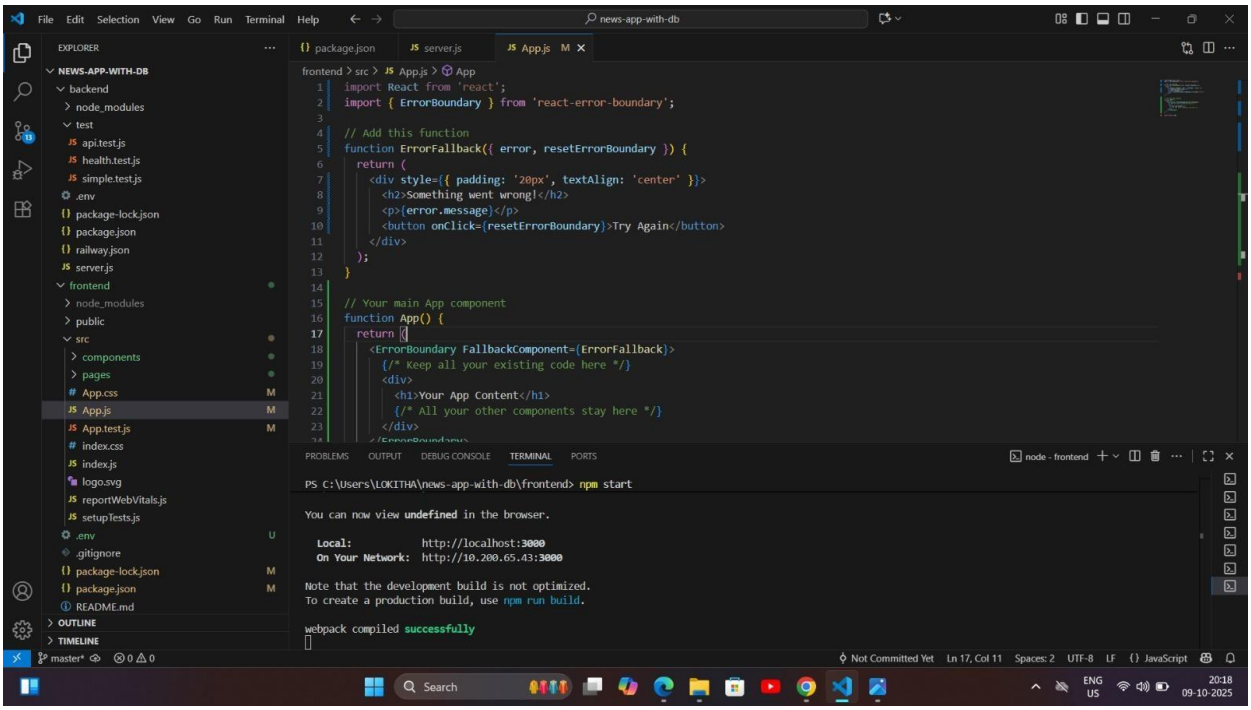
The project successfully implemented a three-schema MongoDB model (Article, Bookmark, User) managed via Mongoose. The Express backend includes advanced routes for:

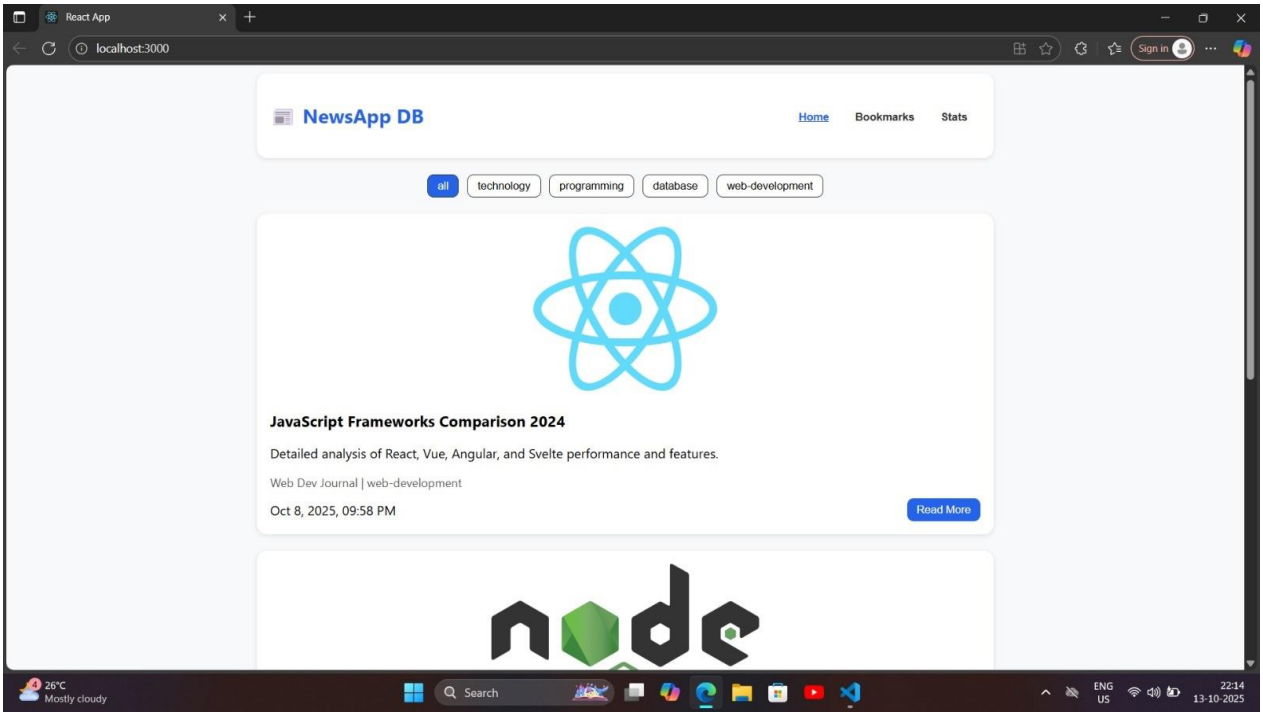
1. News Retrieval: Supports search, category filtering, and pagination on the /api/articles endpoint.
2. Bookmark CRUD:Allows users to save and delete articles, preventing duplicates.
3. Statistics: Uses MongoDB Aggregation to calculate total counts and categorize articles by type.

The React frontend uses axios to consume these APIs. Components like Header (for navigation and filtering via Custom Events) and NewsCard(for bookmark logic) were created. The application successfully loads sample data into the database and provides a functional, data-driven news feed experience.

Screenshots







Challenges & Solutions

Challenge/Difficulty and Solution Applied :
1. Real-time Bookmark Count Update : Used a callback function (onBookmarkUpdate) passed from App.js down to NewsCard.js to trigger a state refresh in the parent component after a successful API call. 2. Decoupled Category Filtering : Implemented Custom DOM Events (window.dispatchEvent) in the Header to communicate the selected category to the listening NewsFeed component.
3. Backend Search & Filtering : Utilized Mongoose queries combining case-insensitive RegExp with the $or operator for flexible search across multiple article fields.

GitHub Link

Github repository link: https://github.com/kesavini-v/Student-Project-Phase1-5