**COLLEGE CODE : **9623

**COLLEGE NAME : **Amrita College of Engineering and Technology

**DEPARTMENT : **Computer Science and Engineering

STUDENT NM-ID : F2DFE2525B0AB62BB764F5B255AC349B

**ROLLNO : **962323104051

DATE :10-10-2025

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.

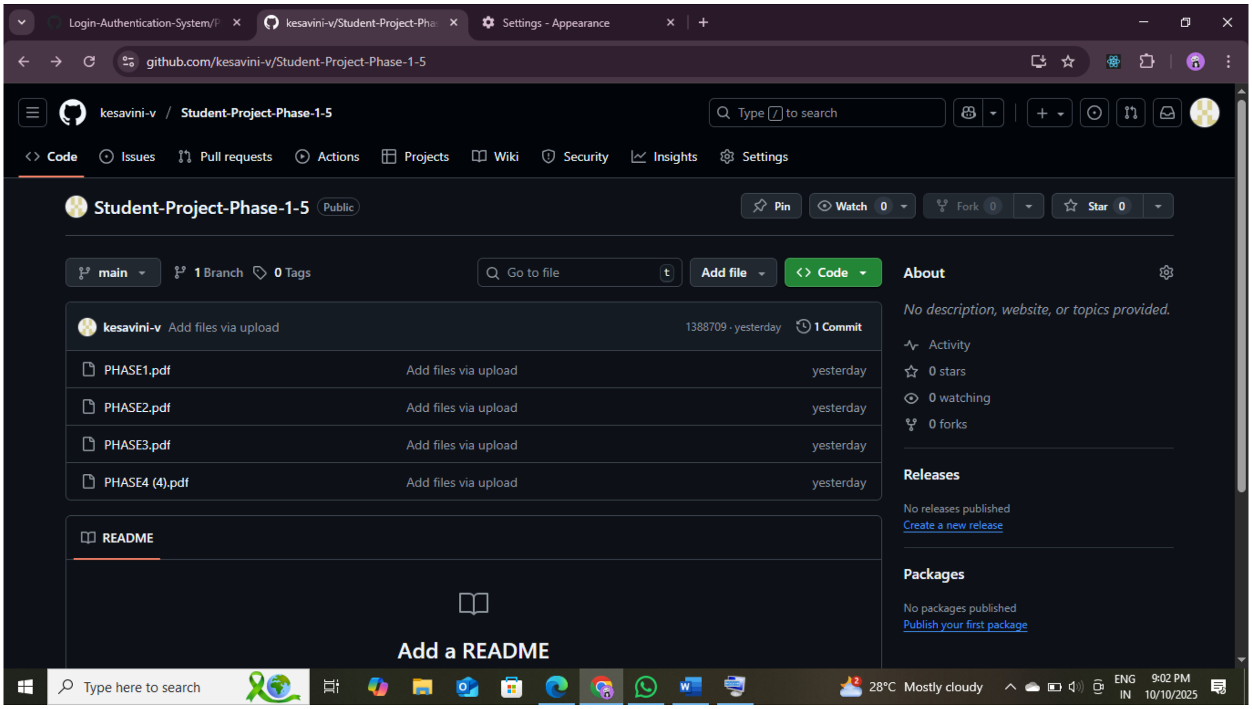
Project Report

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



Student-Project-Phase-1-5 Public

kesavini-v Add files via upload 1388709 · yesterday 1 Commit

PHASE1.pdf	Add files via upload	yesterday
PHASE2.pdf	Add files via upload	yesterday
PHASE3.pdf	Add files via upload	yesterday
PHASE4 (4).pdf	Add files via upload	yesterday

README

Add a README

About

No description, website, or topics provided.

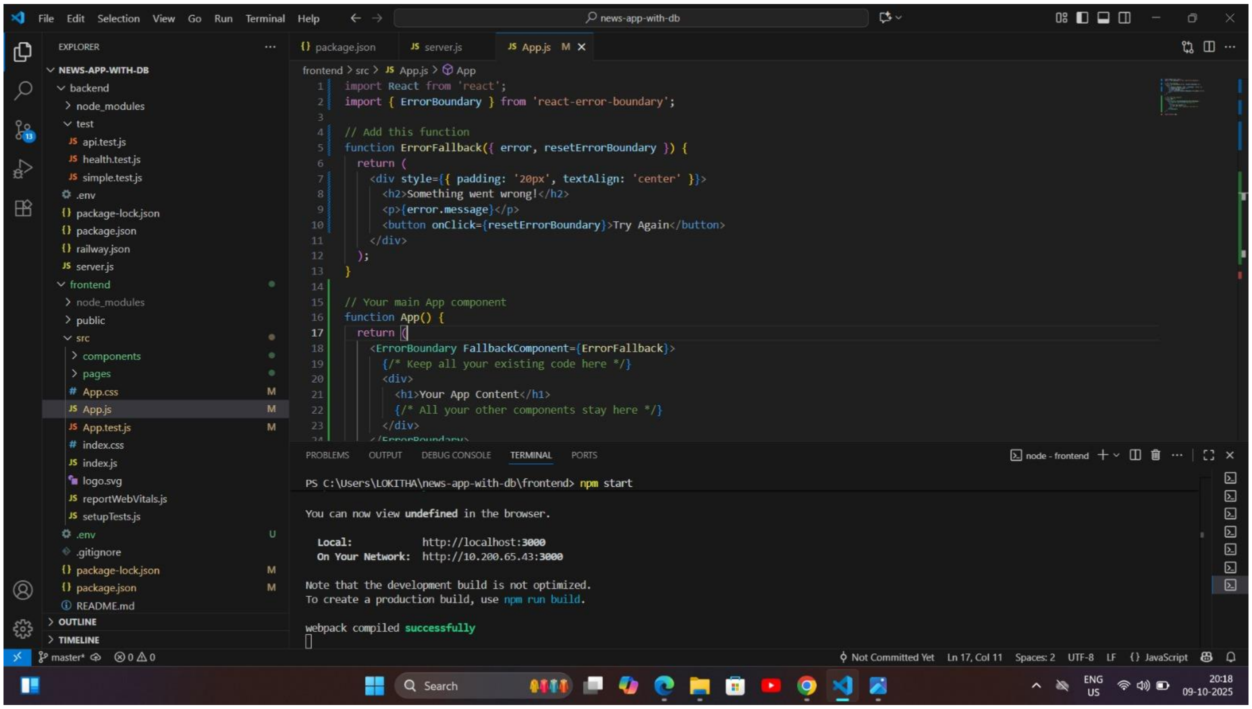
Activity
0 stars
0 watching
0 forks

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package



```
import React from 'react';
import { ErrorBoundary } from 'react-error-boundary';

// Add this function
function ErrorFallback({ error, resetErrorBoundary }) {
  return (
    <div style={{ padding: '20px', textAlign: 'center' }}>
      <h2>Something went wrong!</h2>
      <p>{error.message}</p>
      <button onClick={resetErrorBoundary}>Try Again</button>
    </div>
  );
}

// Your main App component
function App() {
  return (
    <ErrorBoundary FallbackComponent={ErrorFallback}>
      {/* Keep all your existing code here */}
      <div>
        <h1>Your App Content</h1>
        {/* All your other components stay here */}
      </div>
    </ErrorBoundary>
```

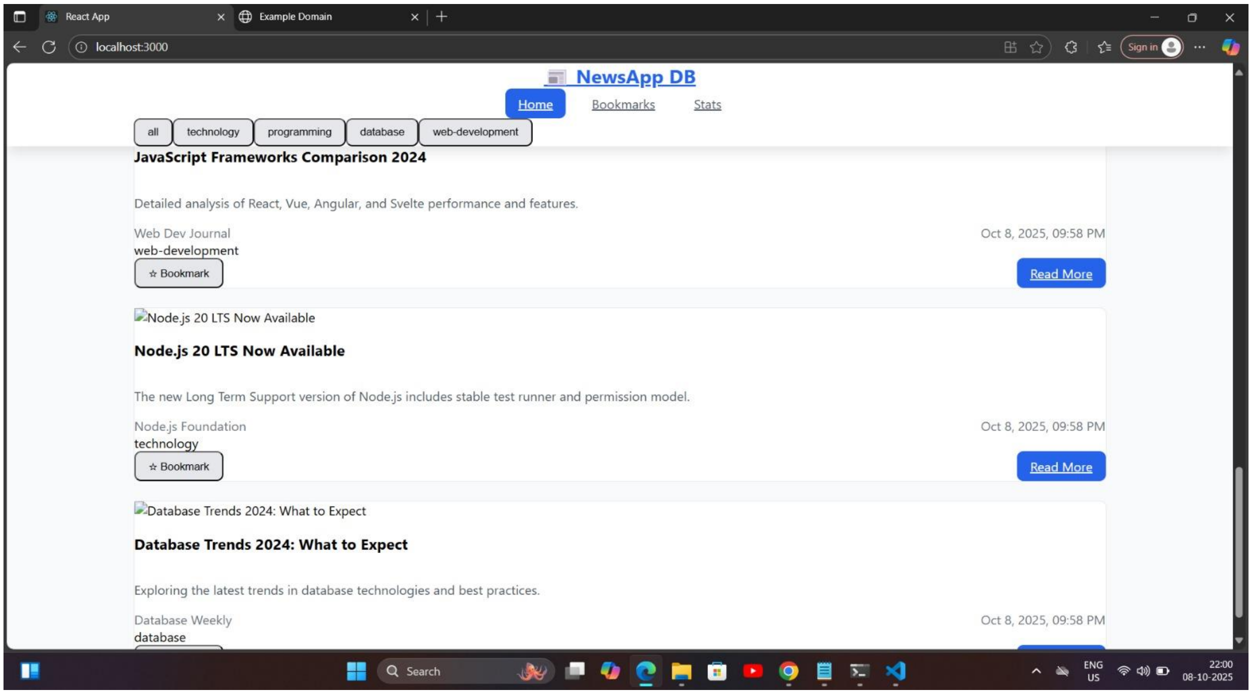
```
PS C:\Users\LOKITHA\news-app-with-db\frontend> npm start

You can now view undefined in the browser.

  Local:            http://localhost:3000
  On Your Network:  http://10.200.65.43:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```



JavaScript Frameworks Comparison 2024

Detailed analysis of React, Vue, Angular, and Svelte performance and features.

Web Dev Journal Oct 8, 2025, 09:58 PM
web-development

Node.js 20 LTS Now Available

Node.js 20 LTS Now Available

The new Long Term Support version of Node.js includes stable test runner and permission model.

Node.js Foundation Oct 8, 2025, 09:58 PM
technology

Database Trends 2024: What to Expect

Database Trends 2024: What to Expect

Exploring the latest trends in database technologies and best practices.

Database Weekly Oct 8, 2025, 09:58 PM
database

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-Phase-1-5