



**COLLEGE CODE : 9623**

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

**DEPARTMENT : Computer Science and Engineering**

**STUDENT NM-ID : D2D7D7C61089879B5AD495BE200716F9**

**ROLLNO : 962323104026**

**DATE : 22-09-2025**

**COMPLETED THE PROJECT NAME AS PHASE 3**

**TECHNOLOGY PROJECT NAME: News Feed Application**

**SUBMITTED BY,**

**NAME: BHAVANA J S**

**MOBILE NO: 72009 02997**

# **PROJECT PHASE 3: News Feed Application**

## **MVP Implementation**

### **1. Project Setup**

#### **Frontend (React.js):**

- Created using create-react-app
- Installed:
  - axios – For HTTP requests
  - react-router-dom – For page routing
  - tailwindcss – For responsive styling
- Folder structure organized by components and pages.

#### **Backend (Node.js + Express.js):**

- Set up with express, mongoose, axios, and cors.
- API routes created for:
  - /api/news – Fetch news
  - /api/bookmarks – Save & retrieve bookmarks
- Connected to MongoDB using mongoose.

#### **Version Control:**

- Git initialized from day one.
- Regular commits made.
- Pushed to GitHub for remote backup and collaboration.

### **2. Core Feature Implementation**

This section outlines how each MVP feature was implemented using your specified tech stack.

#### **2.1 News Feed Display**

- **Frontend:**

- NewsFeed.js fetches articles from backend and displays them in cards.
- Card content includes: headline, image, summary, source, timestamp
- **Backend:**
  - newsController.js uses axios to call NewsAPI.org
  - Articles returned in JSON format to frontend.

## 2.2 Category-Based Filtering

- **Frontend:**
  - Category buttons in Header.js
  - On click, category passed as a query to backend.
- **Backend:**
  - /api/news?category=business → dynamically fetches category-specific news using NewsAPI.

## 2.3 Bookmark / Save Articles

- **Frontend:**
  - Bookmark button sends POST to /api/bookmarks via Axios.
  - Bookmarked items shown on Bookmarks page (Bookmarks.js).
- **Backend:**
  - Mongoose schema bookmarkModel.js stores article info.
  - Routes created to handle saving and retrieving bookmarks.
- **Database:**
  - MongoDB stores bookmarks persistently.

## 2.4 Share News

- **Frontend Only:**
  - Share button uses Web Share API (`navigator.share()`).
  - Fallback: “Copy Link” feature to copy article URL to clipboard.

## 2.5 Search Functionality

- **Frontend:**
  - SearchBar component captures user input and sends query via Axios.
- **Backend:**
  - `/api/news?query=climate` → fetches matching articles via NewsAPI's `q=` parameter.
  - Results dynamically update the news feed.

## 2.6 Responsive UI (Mobile-First)

- Built with Tailwind CSS using mobile-first classes.
- Layout adapts to:
  - Mobile (stacked cards)
  - Tablet (two-column)
  - Desktop (grid of three or more)
- Tested responsiveness manually using Chrome DevTools.

## 3. Data Storage (Local State & Database)

### Frontend (React State):

- `useState` used for managing:
  - News feed data

- Bookmarked articles (temporarily)
- Search input and category selection

### **Backend (MongoDB):**

- Used Mongoose to model and interact with MongoDB.
- Bookmarks are stored with fields like:
  - `articleTitle`, `articleUrl`, `source`, `timestamp`
- Stored persistently in local MongoDB database (`newsFeedDB`).

## **4. Testing Core Features**

### **Frontend Testing:**

- Manual testing in browser:
  - News load
  - Category filtering
  - Bookmark/save action
  - Responsive UI
- Can be extended with Jest + React Testing Library for unit/component tests.

### **Backend Testing:**

- Used Postman to test API endpoints:
  - `GET /api/news`
  - `GET /api/bookmarks`
  - `POST /api/bookmarks`

## **5. Version Control (GitHub)**

- Git initialized at project start.
- Commit history includes:

- Setup commits
  - Feature-specific commits  
(e.g., "Implemented bookmarks feature")
- GitHub repository created with organized branches (optional).
- Example commands used:

```
git init
git add .
git commit -m "Initial setup"
git remote add origin <GitHub repo URL>
git push -u origin main
```