



**COLLEGE CODE : 9623**

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

**DEPARTMENT : Computer Science and Engineering**

**STUDENT NM-ID : 81972D4D6A7D747AC5A401498935D51B**

**ROLLNO : 962323104011**

**DATE : 11-09-2025**

**Completed the project**

**Name as Phase 2**

**TECHNOLOGY PROJECT**

**NAME: News Feed Application**

**SUBMITTED BY,**

**NAME: AISHWARYA S L**

**MOBILE NO:7305298016**

# PROJECT PHASE 2: News Feed Application

## Solution Design & Architecture

### 1. Tech Stack Selection

To build a scalable, fast, and user-friendly news feed app, the following technologies will be used:

Frontend (User Interface)

- React.js – For building interactive and responsive UI components.
- Tailwind CSS / Bootstrap – For clean, modern UI styling.
- Axios / Fetch API – For making HTTP requests to backend APIs.
- React Router – For smooth navigation between pages (Home, Bookmarks, Categories).

Backend (REST API)

- Node.js + Express.js – Lightweight and fast backend for handling API requests.
- MongoDB (NoSQL Database**)** – For storing user bookmarks and preferences.
- Mongoose ORM – For smooth database interaction.

External APIs (News Data)

- NewsAPI.org or Google News API – To fetch real-time headlines, categories, and sources.

Tools & Others

- Git & GitHub – Version control.
- Postman – API testing.
- JWT Authentication (Future scope) – If login/user personalization is added.

### 2. UI Structure / API Schema Design

**UI Structure**

- Header/Navbar → App name, search bar, and category tabs.
- News Feed → Scrollable cards with image, title, summary, source, timestamp.
- Category Section → Filter news by category.

- Bookmarks Page → List of saved articles.
- Footer → About, Contact (optional).

**API Schema (Backend Design)**

Collections / Models

1. User (future scope)
    - userId
    - name
    - email
    - preferences
2. Bookmarks
    - bookmarkId
    - userId (optional for future login feature)
    - articleTitle
    - articleUrl
    - source
    - timestamp

**3. Data Handling Approach**

1. Frontend → Backend → External API
    - User opens app → React fetches data → Node.js backend calls NewsAPI → Data returned to frontend.
2. Bookmark Handling
    - When user clicks “Save”, article details sent to backend → Stored in MongoDB → Retrieved when opening Bookmarks page.
3. Filtering
    - User selects category (e.g., “Sports”) → Backend makes filtered request → Returns sports news only.

**4. Component / Module Diagram**

Components in Frontend

- App.js → Main entry point.
- Header.js → Navigation bar with search & categories.
- NewsCard.js → Displays individual news (image, title, summary, timestamp).
- NewsFeed.js → Fetches and displays multiple NewsCard.
- Bookmarks.js → Displays saved articles.
- Footer.js → Footer content.

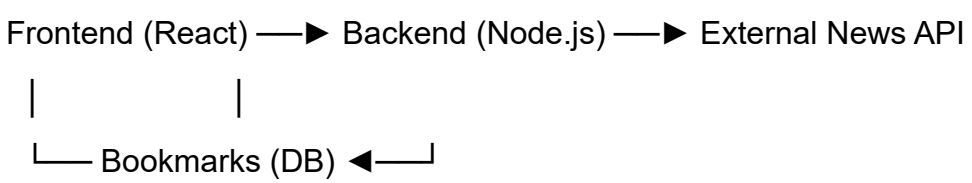
Backend Modules

- server.js → Express server entry point.
- routes/newsRoutes.js → Handles API requests (fetching news, search).
- routes/bookmarkRoutes.js → Handles saving and retrieving bookmarks.
- models/bookmarkModel.js → MongoDB schema for bookmarks.
- controllers/newsController.js → Business logic for fetching & filtering.

Workflow

1. User requests news → NewsFeed.js → Backend → External News API → Response → NewsCard.js.
2. User bookmarks article → Frontend sends POST → Backend → MongoDB → Saved in Bookmarks.js.

Diagram (Text Fo



**5. Basic Flow Diagram**