

AANM & VVRSR POLYTECHNIC

SESHADRI RAO KNOWLEDGE VILLAGE

GUDLAVALLERU

COMPUTER ENGINEERING

PROJECT NAME → NEWS WEBSITE

Members in project:-

SNO	PIN NO	NAME
1	23030-cm-113	G.Sri Raju
2	23030-cm-097	E.Dharma Teja
3	23030-cm-091	D.Sahith Ram
4	23030-cm-094	E.vivek
5	23030-cm-087	D.Ravi Krishna
6	23030-cm-078	D.Mohan Sandeep

Software Requirements Specification

Title Page

Project Name: Todays Voice

Tagline: Dynamic News & Weather Portal

Prepared By: Batch 5

Date: October 2025

GitHub: <https://github.com/dms-2008/simplenews>

Table of Contents

1. Introduction
 2. Overall Description
 3. System Features and Requirements
 4. External Interface Requirements
 5. System Architecture
 6. Future Enhancements
 7. References
-

1. Introduction

1.1 Purpose

Todays Voice provides a dynamic news portal that delivers the latest news, weather updates, and interactive features for users, ensuring fast, reliable, and responsive content.

1.2 Scope

- Dynamic news fetching by category (Tech, Sports, Politics, Latest).
- Weather information retrieval for any city.
- User authentication with secure login and registration.
- Contact form with email functionality.
- Responsive web design with animations and smooth navigation.

1.3 Definitions, Acronyms, Abbreviations

- API: Application Programming Interface
 - SMTP: Simple Mail Transfer Protocol
 - UI: User Interface
-

2. Overall Description

2.1 Product Perspective

- Built with Flask (Python) backend and SQLite database.
- Integrates NewsData.io and OpenWeatherMap.org APIs.
- Responsive web application accessible on desktop and mobile browsers.

2.2 User Classes and Characteristics

User Class	Description	Access Level
General User	Browse news & weather	Public
Registered User	Login to access session	Authentication
Admin (future)	Maintain site content	Admin panel

2.3 Operating Environment

- OS: Windows, Linux, macOS
- Browsers: Chrome, Firefox, Edge, Safari
- Python 3.11+
- Internet connectivity required

2.4 Design Constraints

- API key limits (NewsData.io, OpenWeatherMap.org).
- SQLite for user data (limited concurrency).
- Handle API downtime gracefully.

2.5 User Needs

- Quick access to categorized news.

- Responsive and visually appealing interface.
 - Secure login and registration.
 - Real-time weather information.
 - Easy communication via contact form.
-

3. System Features and Requirements

3.1 Functional Requirements

1. **User Authentication:** Register, login, password hashing, logout.
2. **News Feed:** Fetch & display news, category filter, caching.
3. **Weather Module:** City input, display temp/humidity/wind, styled card.
4. **Contact Form:** Collect name/email/message, send email via SMTP, flash messages.
5. **UI/Navigation:** Sticky header, smooth scrolling, interactive buttons.
6. **Error Handling:** Input validation, API failure handling.

3.2 Non-functional Requirements

- Performance: Load <3s.
 - Security: Hash passwords, protect API keys.
 - Usability: Responsive UI, accessible fonts.
 - Reliability: Use caching for stable API responses.
 - Maintainability: Modular Flask structure with templates and static files.
-

4. External Interface Requirements

4.1 User Interfaces

- Home Page: Hero, news ticker, weather card.
- News Pages: Grid of articles with images.
- Login/Register Pages: Centered forms with validation.
- Contact Page: Form + social links.
- About Page: Team info and project description.

4.2 Hardware Interfaces

- Standard device with internet access.

4.3 Software Interfaces

- Flask backend, SQLite database, NewsData.io & OpenWeatherMap.org APIs, SMTP server.
-

5. System Architecture

Backend: Flask handles routes, authentication, API requests.

Frontend: HTML/CSS/JS templates with animations.

Database: SQLite stores user info securely.

API Integration: NewsData.io & OpenWeatherMap.org with caching.

Architecture Diagram (conceptual)

[User] → [Browser] → [Flask Backend] → [SQLite DB]



[News API] / [Weather API]



[Frontend Templates]

6. Future Enhancements

- Admin dashboard for content management.
 - Personalized news recommendations.
 - Push notifications for breaking news.
 - Migration to MySQL/PostgreSQL for scalability.
 - Multi-language support.
-

7. References

- Flask Documentation: <https://flask.palletsprojects.com/>
- NewsData.io API Docs: <https://newsdata.io/docs>
- OpenWeatherMap.org API Docs: <https://openweathermap.org/api>
- SQLite Documentation: <https://www.sqlite.org/docs.html>