



COLLEGE CODE: 9528

**COLLEGE NAME: SCAD COLLEGE OF ENGINEERING
AND TECHNOLOGY**

**DEPARTMENT : COMPUTER SCIENCE AND
ENGINEERING**

STUDENT NM ID: 172747DBD1D6C799686466A00132E028

Roll no : 952823104116

DATE : 24.10.2025

TECHNOLOGY PROJECT NAME :

TO DO LIST APPLICATION

SUBMITTED By,

NAME: NAMAKANI A.

MOBILE NO:6381212401

Phase 5 — Project Demonstration & Documentation (Deadline – Week 10)

1. Final Demo Walkthrough

The final demonstration showcases the complete functionality of the To-Do List Application, emphasizing both user experience and technical robustness.

Key steps:

Introduction: Briefly explain the purpose, core features, and tech stack.

Live Demo: Walk through user operations such as task creation, updating, deletion, setting priorities, and managing deadlines.

Feature Highlights: Demonstrate added functionalities like notifications, collaboration, and analytics (if implemented).

Backend Overview: Display API endpoints using Postman or Swagger.

Admin Panel / Analytics (if applicable): Showcase any administrative or dashboard features.

Q&A Segment: Address questions about design choices, architecture, and performance improvements.

2. Project Report

A detailed project report provides complete technical and functional documentation.

Report Sections Include:

1. **Project Overview:** Objective, scope, and motivation.
2. **System Architecture:** High-level and detailed architectural diagrams.
3. **Technology Stack:** Tools, libraries, and frameworks used.
4. **Database Schema & API Design:** Entity relationships, sample API schema, and request-response models.

- 5. Implementation Details:** Module-wise explanation (Frontend, Backend, Database).
- 6. Testing Results:** Summary of unit, integration, and UAT outcomes.
- 7. Deployment Process:** Hosting details (Netlify/Vercel/Cloud) and CI/CD workflow.
- 8. Future Enhancements:** Planned or potential improvements.
- 9. Conclusion:** Achievements and learning outcomes.

3. Screenshots / API Documentation

To validate development and testing:

Include UI Screenshots showing different features (Dashboard, Task Creation, Task Editing, etc.).

Include API Documentation generated via Swagger / Postman Collections, listing:

- Endpoint names
- Request methods (GET, POST, PUT, DELETE)
- Parameters and response structures
- Authentication details (JWT tokens, etc.)

4. Challenges & Solutions

During development, several challenges may arise — this section explains how they were addressed.

Challenge

Managing state efficiently in frontend

Slow API response under heavy load

Data synchronization between users

Security vulnerabilities

Deployment configuration issues

Solution

Implemented React Context API / Redux for centralized state management

Added caching and optimized database queries

Integrated WebSockets for real-time updates

Applied JWT authentication and validated all API inputs

Used Docker & environment variables for smoother setup

5. GitHub README & Setup Guide

A comprehensive `README.md` ensures anyone can understand, set up, and contribute to your project.

README Contents:

Project Title & Description

Features Overview

Tech Stack

Project Architecture Diagram

Installation Steps:

```
git clone https://github.com/username/todo-list-app.git  
cd todo-list-app
```

```
npm install
```

```
npm run dev
```

Environment Setup:

- Example .env file (without exposing secrets)
- API base URL configuration

API Reference Link (Swagger / Postman)

Screenshots / Demo Video Link

Deployed App Link (Netlify / Vercel)

Contributors & License Information

6.Final Submission (Repository + Deployed Link)

For final grading and evaluation:

GitHub Repository Link — must contain source code, documentation, and README.

Live Deployed Link — hosted on Netlify, Vercel, or Cloud platform (backend + frontend integrated).

Documentation Folder — should include:

- Project Report (PDF / DOCX)
- Screenshots folder

- API documentation JSON or HTML export
- Presentation slides (if applicable)

GITUPID:<https://github.com/namakani/To-do-list-application-.git>