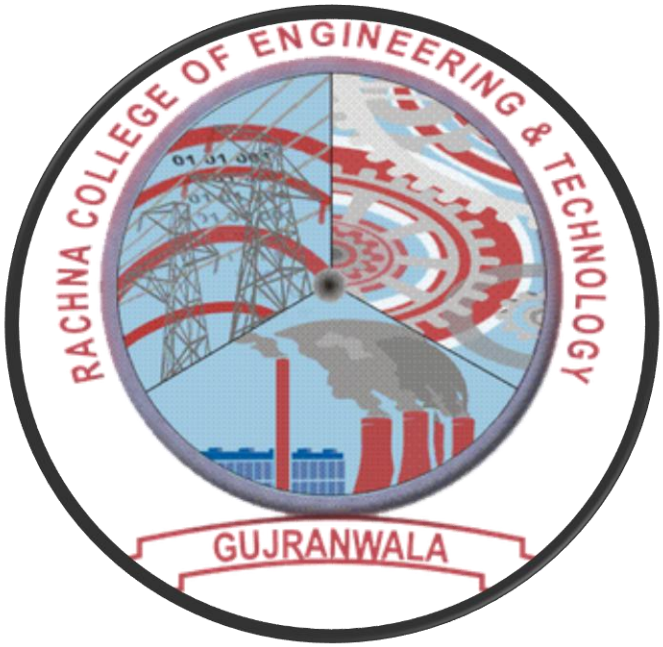

Software Requirements Specification (SRS) for TaskMate

(Based on the Project Proposal: TaskMate – All-in-One Productivity Platform)

Submitted By:

• Adil Ur Rehman (2024-CS-760)

• Atika Asif (2024-CS-498)

• Shawal Ubaid (2024-CS-484)

• Meerab Ihtisham (2024-CS-459)

Submitted To:

• Dr. Natasha Nigar

Department of Computer Science

Rachna College of Engineering & Technology (RCET)

Contents

1. Introduction

1.1 Purpose

This Software Requirements Specification (SRS) describes the requirements for TaskMate, a web-based productivity platform that consolidates multiple document-generation tools into a single application. It defines functional and non-functional requirements, intended for developers, testers, and stakeholders.

1.2 Scope

TaskMate provides CV, Resume, Invoice, and Business Card Generators, supports user-contributed custom tools, and uses Node.js for backend logic, MySQL for persistent data storage, and JSON files for temporary data handling.

1.3 Definitions, Acronyms, and Abbreviations

- WBP – Web-Based Platform
- JSON – JavaScript Object Notation
- UI/UX – User Interface / User Experience
- DBMS – Database Management System

1.4 References

Project Proposal: TaskMate – All-in-One Productivity Platform.

2. Overall Description

2.1 Product Perspective

TaskMate is an independent web application built with HTML, CSS, JavaScript, Node.js, and MySQL. It integrates multiple productivity tools and allows future expansion through community contributions.

2.2 Product Functions

Main system functions include:

- Generate CV/Resume
- Create invoices
- Create business/visiting cards

• Browser-based interface.

• REST API communication between frontend and backend. • Provide real-time previews through JSON temporary storage

• Allow contributors to create and add new tools

• Display contributors' names with their tools

2.3 User Classes and Characteristics

• General Users – Create documents using available tools. • Contributors – Develop new tools and submit them for approval. • Admin – Reviews and verifies submitted tools.

2.4 Operating Environment

TaskMate runs in any modern browser (Chrome, Firefox, Edge). Backend runs on Node.js. MySQL provides persistent storage.

2.5 Design & Implementation Constraints

• Must use HTML, CSS, JS on the front end. • Must use Node.js on the backend. • Must use MySQL for main storage. • JSON must be used for temporary data. • Deployment environment limited to GitHub Pages/Netlify for frontend.

2.6 Assumptions and Dependencies

• Users have internet access.

• MySQL database remains operational.

• Contributors submit valid and original tools.

3. Specific Requirements

3.1 Functional Requirements

• User login & registration using MySQL.

• Document generation (CV, Resume, Invoice, Card).

• Live previews using JSON temporary storage.

• Tool contribution module.

• Admin verification of submitted tools.

• Add approved tools to main platform.

3.2 External Interface Requirements

• UI must be simple, clean, and Figma-designed.

3.3 Non-Functional Requirements

• Performance – Pages load within 3 seconds.

• Security – Data validation and secure database queries.

• Usability – User-friendly UI.

• Scalability – Support for additional tools. • Maintainability – Clean modular code.

4. System Models

4.1 Use Case Summary Actors:

• User – Generates documents, submits tools.

• Admin – Verifies and approves contributed tools.

Use Cases:

• Create document

• Preview document

• Save document

• Submit a tool

• Approve tool

5. Other Requirements

• Contributors must submit original tools.

• No copyrighted templates may be used without permission.

• System must remain accessible and stable during expected traffic levels.