Software Requirement Specification (SRS)

Technogrowth Knowledge Base Web Application

1. Introduction

1.1 Purpose

The purpose of this document is to define the requirements for the development of the Technogrowth Knowledge Base Web Application. This application will serve as a platform for newly joined trainee engineers or resources to access training materials, assignments, interview questions, practical assignments, and notes related to various technologies (Angular, Node.js, .NET, Java, React, etc.).

1.2 Scope

This system will provide:

- User Authentication (Login/Registration)
- Dashboard for accessing study material categorized by technology
- Uploading and managing materials (Admin only)
- Assignments, Notes, Interview Questions Management
- Technology-specific materials (with images)
- Tracking completion status (Optional future scope)
- Admin Panel for content management

The application will be built using:

- Frontend: Angular
- Backend: NestJS
- Database: MongoDB or PostgreSQL

2. Overall Description

2.1 Product Perspective

The application is a new, independent web-based platform intended for internal organizational use. It will consist of two main user roles:

- Admin: Manage technologies, upload and manage study materials, assignments, questions, notes.
- Trainee User: View and consume the available resources.

2.2 Product Functions

- User Registration and Login
- Dashboard view of available technologies with images
- Viewing and downloading study material
- Viewing assignments and interview questions
- Upload and organize content based on technologies (Admin)
- User Profile Management
- (Future scope) Completion tracking, quiz assessment

2.3 User Classes and Characteristics

- Admin: Power user, Can add/update/delete study materials, manage technologies, view trainee activity
- Trainee: View-only access to materials, Can mark materials as read/completed

2.4 Operating Environment

- Web Browser (Google Chrome, Firefox, Edge)
- Responsive Design (Desktop, Tablet, Mobile)
- Hosted on Cloud (AWS / Azure / On-Premise server)

2.5 Design and Implementation Constraints

- Angular 18 (or latest stable)
- NestJS (latest)
- MongoDB or PostgreSQL
- JWT Authentication
- REST APIs

- SCSS / TailwindCSS for styling
- CI/CD for deployment (future scope)

2.6 Assumptions and Dependencies

- Users have basic familiarity with web applications.
- Admin content uploads are valid and reviewed.
- Application will have organization-controlled access (no public access).

3. Specific Requirements

3.1 Functional Requirements

Authentication Module:

- User Registration
- Login with Email and Password
- JWT Token-based Authentication
- Forgot Password (optional)

User Dashboard:

- List of available technologies with images
- Material Details View Page
- Search and Filter materials

Admin Dashboard:

- Technology Management (Add/Edit/Delete with image upload)
- Upload Study Materials
- Upload Assignments
- Upload Interview Questions
- Upload Notes
- View list of registered users

Content Management:

- Study Materials, Assignments, Interview Questions, Notes
- Fields: Title, Description, File Upload, Technology Association, Upload Date, Uploaded By

Profile Management:

- Update basic user profile details

Notifications (Optional Phase 2):

- Admin notifications for new uploads
- User reminders for pending materials

3.2 Non-Functional Requirements

- Performance: Application should load within 3 seconds on a 4G network.
- Security: Passwords hashed using bcrypt, Role-based access control.
- Scalability: System should be scalable.
- Maintainability: Modular codebase for easy future upgrades.
- Responsiveness: Should work seamlessly on all devices.

4. Database Design Overview

Collections/Tables:

- users: id, name, email, password, role (Admin/Trainee), profilePic, createdAt, updatedAt
- technologies: id, name, description, imageUrl, createdAt
- materials: id, title, description, fileUrl, materialType, technologyId, createdBy, createdAt
- user_material_status (optional): id, userId, materialId, status (pending/completed)

5. APIs Overview (Backend - NestJS)

- POST /auth/register: User Registration
- POST /auth/login: User Login
- GET /technologies: List Technologies with images
- POST /technologies: Create Technology
- PUT /technologies/:id: Update Technology
- DELETE /technologies/:id: Delete Technology
- GET /materials: List Materials
- POST /materials: Upload Material
- PUT /materials/:id: Update Material
- DELETE /materials/:id: Delete Material
- GET /profile: Get User Profile
- PUT /profile: Update User Profile

6. Wireframe (High-level Screens)

- Login Page
- User Dashboard with technology cards (with images)
- List of Materials per Technology
- Material Details Page
- Admin Dashboard
- Technology Management Page
- Material Upload Form
- User Profile Page

7. Future Enhancements (Phase 2+)

- Quiz and MCQ Assessments
- Auto Certification after completion
- Video Streaming
- Track User Progress
- Leaderboard for Trainees
- Notifications / Email Alerts