

# Software Requirements Specification

Version 1.0



## PathSeeker

**Discover What Fits You Best...**

**Theme: Career Passport**

**Category: Full-Stack Application Development**

# TABLE OF CONTENTS

**1.1 Background and Necessity for the Full-Stack Web Application**

**1.2 Proposed Solution**

**1.3 Purpose of this Document**

**1.4 Scope of Project**

**1.5 Constraints**

**1.6 Functional Requirements**

**1.7 Non-Functional Requirements**

**1.8 Interface Requirements**

➤ **Hardware**

➤ **Software**

➤ **Database Design**

**1.9 Project Deliverables**

## 1.1

## Background and Necessity for a Full-Stack Web Application

In today's competitive world, students, graduates, and professionals often struggle to choose career paths that align with their skills and interests. Existing resources are either too generic, difficult to navigate, or not tailored to individual user requirements.

There is a growing demand for a fully functional, interactive Web application that simplifies career exploration and personalizes guidance. An application is required to fulfill this demand by offering a dynamic and responsive platform which provides:

- Tailored career options based on user type
- Interest-based quizzes and recommendations
- Multimedia learning and downloadable resources

Built with modern front-end technologies and static data files, it should ensure seamless performance even without backend support — making it ideal for institutions, individuals, and offline-ready deployment.

## 1.2

## Proposed Solution

To address the gap in accessible, personalized career guidance, propose development of a fully functional Web application named **PathSeeker** a responsive, interactive **Career Passport platform** designed for students, graduates, and working professionals.

The application offers a seamless user experience by segmenting content based on the user's academic or professional stage. Each user type is guided through tailored career paths using intuitive UI, interactive tools, and visual aids after they have registered and duly logged in.



### 1.3

## Purpose of the Document

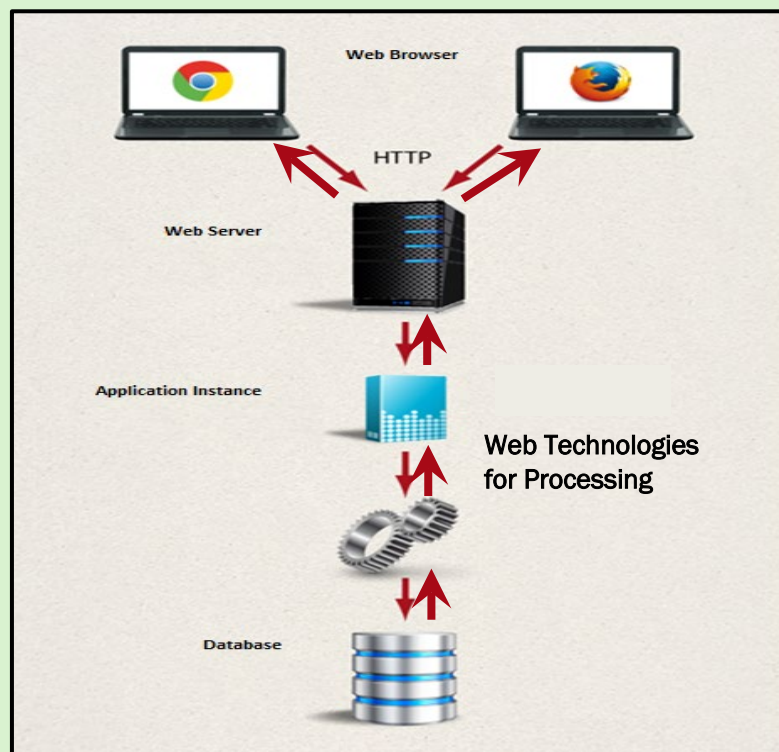
This document presents a detailed description of the **PathSeeker** application, explaining its features, purpose, scope, and limitations. It is intended for both stakeholders and developers of the application.

### 1.4

## Scope of the Project

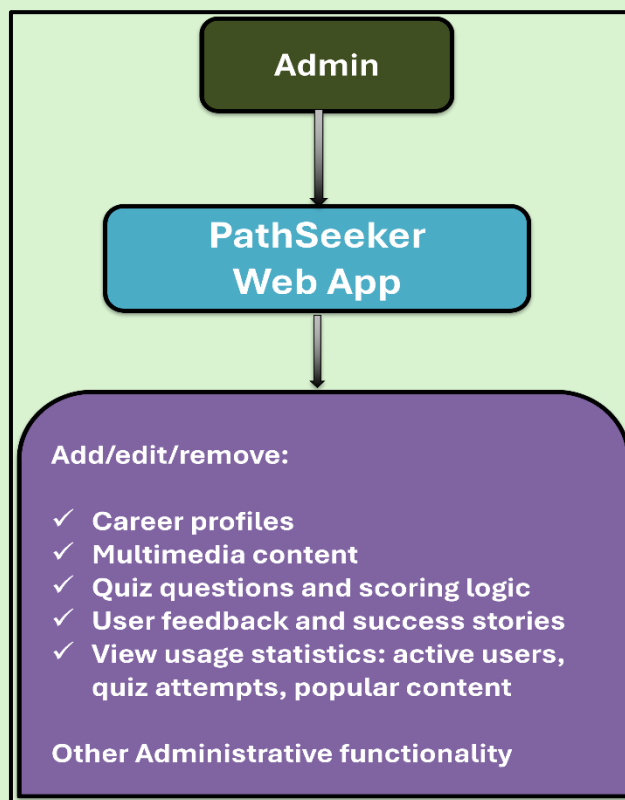
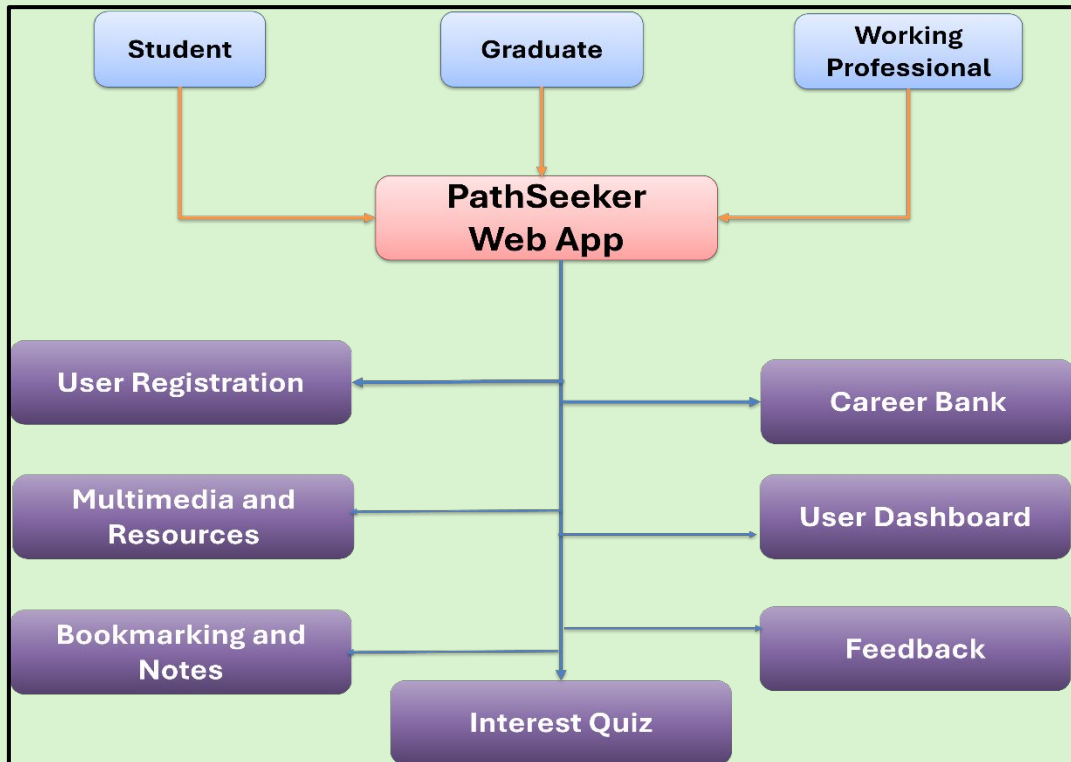
**PathSeeker** is a full-stack Web application designed to guide students, graduates, and working professionals in making informed career decisions. It offers a personalized experience through user registration and login, an interest-based quiz, a dynamic career bank comprising global job roles, and a customizable dashboard. The application includes a backend system to support data storage, user session management, feedback submission, and content updates. Users can explore careers, bookmark resources, view expert videos, and download career materials. Built with modern technologies, the platform is scalable, responsive, and adaptable for institutions, career fairs, and personal use, offering an interactive Career Passport solution. Administrative functionality will also be part of the application, giving full control over various features, including user management and so on.

### Architecture Diagram for PathSeeker



## Flow Diagrams

Sample flow diagrams depicting interaction between various entities and the application is shown here:



## 1.5

### Constraints

Development of the **PathSeeker** Web application must adhere to several constraints to ensure its successful implementation and operation. Technically, the application must be compatible with major Web browsers and responsive across various devices. You may encounter constraints related to data storage, data synchronization, and backup procedures. The usage of various images and videos may be subject to licensing agreements and copyright restrictions. It is important to understand and comply with these constraints to avoid legal issues. The Web application will not have any functionality for actual validation of resume submitted, other data submitted, or payment gateways. These are beyond the scope of the application.



## 1.6

### Functional Requirements

The **PathSeeker** Web application will offer a complete, role-based career exploration experience with dynamic frontend features and robust backend support. It will not only cover essential functionality but also implement advanced capabilities to enhance personalization, scalability, and interactivity.



#### User Authentication and Management

- Role-based registration and login (Student, Graduate, or Professional) - Admin will have direct access to login feature
- Secure session management
- Forgot password/Reset password feature and email verification via One Time Password (OTP) or tokenized link
- User profile creation with editable education, skills, and interest fields and work experience, if applicable
- Optionally, file upload feature to allow users to upload their resume to the portal



#### Personalized Dashboard

- Displays personalized greeting, recent activity, quiz results, and bookmarked items
- Recommends careers, content, and videos based on user interaction history
- Displays Dynamic widgets: 'Trending Careers', 'Top Picks for You', and so on.



### Career Bank (with Advanced Filters)

- Fetches and displays careers/job roles from a backend database
- Multi-level filtering (domain, skill match, expected salary, or job demand)
- Smart Search with autocomplete and spell-check using Elasticsearch or similar
- Save search filters and preferences (for example, show only tech or healthcare roles)



### AI-Powered Interest Quiz (Optional)

Use AI tools to prepare quiz questions with following features:

- Multi-step quiz with timed questions, sliders, and Likert scale ratings
- Quiz history stored for progress tracking
- Auto-suggestion of streams and job roles based on current trends and user performance in the quiz



### Interactive Multimedia Center

- Stream embedded videos, audio podcasts, and animated explainers
- Video player with transcript toggle, playback control, and related content suggestions
- Admin-controlled tagging and categorization
- User feedback/rating on videos (5-star or thumbs-up/down system)



### Success Stories Hub

- Card-based success stories with domain-based filtering
- Timeline-style storytelling (educational path → challenges → outcome)
- Option for users to submit their own stories (admin approval required)



### Document Resource Library

- Downloadable PDFs, checklists, and infographics grouped by type and target audience
- Auto-preview for documents with popups or modals
- Backend-driven tagging (Example: “Beginner,” “Scholarship,” “Skill-Building”)
- Admin can track download count and popularity



### Feedback and Analytics

- Dynamic feedback form with type categorization (bug, suggestion, or query)
- Admin dashboard to view feedback analytics (for example, sentiment summary and response type stats)

- In-app notification center for feedback responses or announcements



### **Bookmarking, Notes, and Sharing**

- Bookmark any career, article, or video
- Add sticky notes or comments to bookmarks
- Export notes and bookmarks as PDF or share via email/social media
- Auto-suggestion of similar careers or content based on bookmarks



### **Admin Control Panel**

Add/edit/remove:

- Career profiles
- Multimedia content
- Quiz questions and scoring logic
- User feedback and success stories
- View usage statistics: active users, quiz attempts, popular content



### **System Intelligence (Advanced UX)**

- Recently viewed item history (session and persistent)
- Predictive analytics for career trends using stored data (optional)
- Dynamic recommendations based on collaborative filtering or interaction behaviour
- “If you liked this...” suggestion engine for content and careers



### **Accessibility and UI Enhancements**

- Dark mode toggle, font-size adjustment for accessibility
- Breadcrumbs for navigation clarity
- Smooth transitions and loading spinners of media

**Note:** Boilerplate or readymade HTML template can be used, provided it is only for design aspect and not for implementing application functionality.

Do NOT copy content or code from GPTs or other AI tools, although you are permitted to use images generated by AI tools for any visual representation purposes. It is mandatory to mention such tools used in case you add any AI generated images.



## 1.7

## Non-Functional Requirements



There are several non-functional requirements that should be fulfilled by the application. They include:

- **Safe to use:** The application should not result in any malicious downloads or unnecessary file downloads.
- **Accessibility:** The application should have clear and legible fonts, user-interface elements, and navigation elements.
- **User-friendliness:** The application should be easy to navigate with clear menus and other elements and easy to understand.
- **Operability:** The application should be reliable and efficient.
- **Performance:** The application should demonstrate high value of performance through speed and throughput. In simple terms, the application should have minimal load time and smooth page redirection.
- **Scalability:** The application architecture and infrastructure should be designed to handle increasing user traffic, data storage, and feature expansions.
- **Security:** The application should implement adequate security measures such as authentication. For example, only registered users can access certain features.
- **Availability:** The application should be available 24/7 with minimum downtime.
- **Compatibility:** The application should be compatible with latest browsers and various devices.



These are the bare minimum expectations from the project. It is a must to implement the FUNCTIONAL and NON-FUNCTIONAL requirements given in this SRS. Once they are complete, you can use your own creativity and imagination to add more features if required.



## 1.8

## Interface Requirements

### HARDWARE

- Intel Core i5/i7 Processor or higher
- 8 GB RAM or higher
- Color SVGA monitor
- 500 GB Hard Disk space
- Mouse
- Keyboard

**IDE:** Appropriate IDE as per the platform

**Frontend:** HTML5, CSS3, Bootstrap, ReactJS/AngularJS/Angular/TypeScript, JavaScript, jQuery, and XML

**Backend:** Java SDK with Apache NetBeans or Eclipse, Jakarta EE

OR

C# with ASP.NET MVC and ASP.NET MVC Core (optional), Visual Studio IDE

OR

PHP with Laravel Framework

OR

Python with Flask or Django

OR

MongoDB, Express.js, Angular, Node.js

OR

MongoDB, Express.js, React, Node.js

Database: MySQL / SQL Server

For local hosting (optional):

XAMPP latest version

### SOFTWARE

## Database Design

Based on the given specifications, you will define suitable entities, attributes for these entities, and identify relationships between the entities. For example, some entities along with their attributes can be identified as follows:

<table><tr><th>Admins</th></tr><tr><td>admin_id (PK)</td></tr><tr><td>name</td></tr><tr><td>email</td></tr><tr><td>password_hash</td></tr></table>	Admins	admin_id (PK)	name	email	password_hash	<table><tr><th>Careers</th></tr><tr><td>career_id (PK)</td></tr><tr><td>title</td></tr><tr><td>description</td></tr><tr><td>domain</td></tr><tr><td>required_skills</td></tr><tr><td>education_path</td></tr><tr><td>expected_salary</td></tr></table>	Careers	career_id (PK)	title	description	domain	required_skills	education_path	expected_salary	<table><tr><th>Users</th></tr><tr><td>user_id (PK)</td></tr><tr><td>uname</td></tr><tr><td>email (unique)</td></tr><tr><td>password_hash</td></tr><tr><td>role (student/graduate /professional)</td></tr></table>	Users	user_id (PK)	uname	email (unique)	password_hash	role (student/graduate /professional)						
Admins																											
admin_id (PK)																											
name																											
email																											
password_hash																											
Careers																											
career_id (PK)																											
title																											
description																											
domain																											
required_skills																											
education_path																											
expected_salary																											
Users																											
user_id (PK)																											
uname																											
email (unique)																											
password_hash																											
role (student/graduate /professional)																											
<table><tr><th>Resources</th></tr><tr><td>resource_id (PK)</td></tr><tr><td>title</td></tr><tr><td>category</td></tr><tr><td>description</td></tr><tr><td>file_url</td></tr><tr><td>tag</td></tr><tr><td>views_count</td></tr><tr><td>created_by (FK)</td></tr></table>	Resources	resource_id (PK)	title	category	description	file_url	tag	views_count	created_by (FK)	<table><tr><th>SuccessStories</th></tr><tr><td>story_id (PK)</td></tr><tr><td>rname</td></tr><tr><td>domain</td></tr><tr><td>story_text</td></tr><tr><td>image_url</td></tr><tr><td>submitted_by (FK)</td></tr><tr><td>approved_by (FK)</td></tr><tr><td>approved_at</td></tr></table>	SuccessStories	story_id (PK)	rname	domain	story_text	image_url	submitted_by (FK)	approved_by (FK)	approved_at	<table><tr><th>UserProfiles</th></tr><tr><td>profile_id (PK)</td></tr><tr><td>user_id (FK)</td></tr><tr><td>education_level</td></tr><tr><td>interests</td></tr><tr><td>profile_image</td></tr><tr><td>updated_at</td></tr></table>	UserProfiles	profile_id (PK)	user_id (FK)	education_level	interests	profile_image	updated_at
Resources																											
resource_id (PK)																											
title																											
category																											
description																											
file_url																											
tag																											
views_count																											
created_by (FK)																											
SuccessStories																											
story_id (PK)																											
rname																											
domain																											
story_text																											
image_url																											
submitted_by (FK)																											
approved_by (FK)																											
approved_at																											
UserProfiles																											
profile_id (PK)																											
user_id (FK)																											
education_level																											
interests																											
profile_image																											
updated_at																											
<table><tr><th>Multimedia</th></tr><tr><td>media_id (PK)</td></tr><tr><td>title</td></tr><tr><td>type</td></tr><tr><td>url</td></tr><tr><td>tags</td></tr><tr><td>transcript</td></tr><tr><td>rating_avg</td></tr><tr><td>rating_count</td></tr></table>	Multimedia	media_id (PK)	title	type	url	tags	transcript	rating_avg	rating_count	<table><tr><th>QuizQuestions</th></tr><tr><td>question_id (PK)</td></tr><tr><td>question_text</td></tr><tr><td>type</td></tr><tr><td>options</td></tr><tr><td>correct_answer</td></tr><tr><td>weightage</td></tr></table>	QuizQuestions	question_id (PK)	question_text	type	options	correct_answer	weightage	<table><tr><th>Feedback</th></tr><tr><td>feedback_id (PK)</td></tr><tr><td>user_id (FK)</td></tr><tr><td>category</td></tr><tr><td>message</td></tr><tr><td>status</td></tr><tr><td>submitted_at</td></tr></table>	Feedback	feedback_id (PK)	user_id (FK)	category	message	status	submitted_at		
Multimedia																											
media_id (PK)																											
title																											
type																											
url																											
tags																											
transcript																											
rating_avg																											
rating_count																											
QuizQuestions																											
question_id (PK)																											
question_text																											
type																											
options																											
correct_answer																											
weightage																											
Feedback																											
feedback_id (PK)																											
user_id (FK)																											
category																											
message																											
status																											
submitted_at																											

Similarly, you can define other entities and relationships between entities and methods representing activities on the entities.

**Note:** These are just examples, you do not have to adhere to these structures and can design your own table structure with different columns.

You will require to design and build the project and submit it along with a complete project report that includes:

- Problem Definition
- Design Specifications
- Diagrams such as Flowcharts for various Activities, Data Flow Diagrams, and so on
- Database Design
- Test Data Used in the Project
- Project Installation Instructions (mandatory)
- **User Credentials for all Types of Users with Passwords**

Documentation is considered as a very important part of the project. Ensure that documentation is complete and comprehensive.

**Documentation should not contain any source code.**

The consolidated project will be submitted as a zip file with a ReadMe.doc file listing assumptions (if any) made at your end and SQL scripts files (.sql) OR schema files containing database and table definitions.

**Note:** Preferably, host the working Web application on a Website and share the URL for evaluation.

**Submit a video (.mp4 file) demonstrating the working of the Web application, including all features under Functional Requirements. This is MANDATORY.**

Over and above the given specifications, you can apply your creativity and logic to improve the system.

**Sitemap:** To understand the flow of **PathSeeker** Web Application, you will have to create a Sitemap and add it to the home page of your application.

\*\*\*\*\*