**TECHTRONIX CORPORATION**

Techtronix Corp.

Software Requirement Specification for Books Directory Node app

**Submitted By**

Syed Sheraz Haider Naqvi

**Submitted To**

Tahir Mehmood

# Introduction

### Purpose

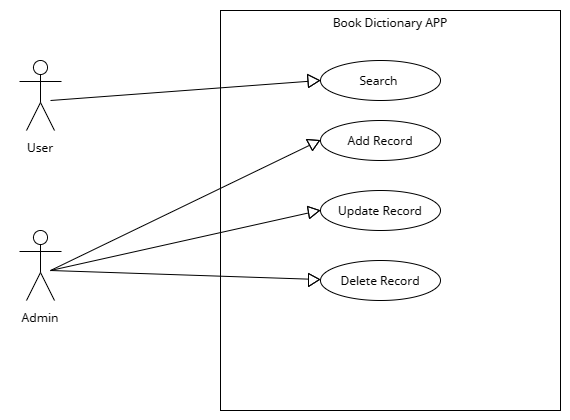
The purpose of this document is to present description of the Book Dictionary app. It will explain the features of the system, what the system will do, the constraints under which it must operate.

### Project Scope

The Book Dictionary app is an implementation of a system which allows user to search for various books available in inventory by book name, author and genre. This project covers features like registration of the users and admin, authentication and authorization.

System also allows adding, deleting or updating books inventory but those operations are restricted to admin role only to ensure data integrity.

# Product Use Case



# Functional Requirements

### 3.1 User Management:

* The system should allow users to register and create accounts.
* Users should be able to log in and log out of the system.
* User authentication should be implemented to ensure secure access to the app.
* User roles should be defined, such as user and administrator.

# Add Book:

* Administrators should have the ability to add a new book to the directory.
* The "Add Book" functionality should allow administrators to enter the book details, such as name of book, authors, and genres.
* The system should store the new book record in the database.

# Search Books:

* Users should be able to search for books by name, author, genre.
* The system should display a list of books that match the search criteria.
* Apis will only shows 10 books at a time.

# Update Book:

* Administrators should have the ability to update the details of a book in the directory.
* The system should update the book record in the database with the new details.

# Delete Book:

* Administrators should have authority privilege to delete a book from the directory.
* Once deleted, the book record should be permanently removed from the database.

# Non-functional requirements

### 4.1 Error Handling:

Proper error handling mechanisms should be implemented to provide meaningful error messages for invalid requests or errors during API calls.

### Performance Requirements

The system should provide fast and responsive performance for users.

Book search and retrieval should be efficient, even with a large number of books in the directory.

### Security Requirements

The system should apply proper authentication and authorization mechanisms to ensure secure access.

API endpoints related to adding, updating, and deleting books should be accessible only to administrators.

Regular users should have read-only access to search and view book details.

### 4.4 Maintainability:

The system should be built using well-structured code.

Code should follow coding standards and best practices to ensure readability.

# Database:

The database should contain the following tables:

### User:

* This table will store the information of users. i.e Admin, User
* The information contains ID, NAME, EMAIL, ROLE of user.

### Books:

* This table will store the information related to books
* The information contains ID, NAME, AUTHORS, GENRE