
System Requirements Specification Index

For

School Management Application (MS SQL)

Version 4.0

IIHT Pvt. Ltd.

IIHT Ltd, No: 15, 2nd Floor, Sri Lakshmi Complex, Off MG Road, Near SBI LHO,
Bangalore, Karnataka – 560001, India

fullstack@iiht.com

SCHOOL MANAGEMENT APPLICATION SYSTEM

System Requirements Specification

1. PROJECT ABSTRACT

1.1 PROBLEM STATEMENT:

The purpose of this application is to allow the visitors and school management to view Notice, search Students by name, add See Library in the book and list of teachers or many more functions.

1.2 Following is the requirement specifications:

	School Management Application
School Controller	
2	Find a notice by name.
3	See all books in the library.
4	Find student and student list.
5	Show list of enrolled teachers.
6	Borrow a Book From Library, and get Borrow book info while collecting book from Library.

2. TEMPLATE CODE STRUCTURE

2.1 Package: Schoolmanagement

Resource

Names	Resource	Remarks	Status
Package Structure/Project			
controller	SchoolController	School controller handle all action method for crud operations.	Partially Implemented
appsettings.json		Contains connection string for database.	Already Implemented
Properties	launchsettings.json	Url settings for api	Already Implemented

2.2 Package: Schoolmanagement.BusinessLayer

Resource

Names	Resource	Remarks	Status
Package Structure			
Interfaces	Interface directory contain all interface for Services class	Inside this directory contains all business logic code and CURD Operation Logic method.	Already Implemented
Services, Repository	SchoolServices, SchoolRepository, ISchoolRepository cs file for Method and business logic	Using this all cs file performed all CRUD operations.	Partially Implemented
ViewModels	Cs file for represent all view entities	All view entities setting class	Already Implemented

2.3 Package: Schoolmanagement.DataLayer

Resource

Names	Resource	Remarks	Status
Package Structure			
SchoolDbContext	Contain all business logic for data set and Dbset setting	Using this cs file performed all Data related settings operations.	Already Implemented

2.4 Package: Schoolmanagement.Entities

Resources

Names	Resource	Remarks	Status
Package Structure			
Entities Class	Teacher, Student, Notice, Library, Class List cs file	Contain all entities property for application	Already Implemented

2.5 Package: Schoolmanagement.Test

Resources

The Schoolmanagement.Test project contains all test case classes and functions for code evaluation. Don't edit or change anything inside this project.

3. REST ENDPOINTS

Rest End-points to be exposed in the controller along with method details for the same to be created

3.1 SchoolController

URL Exposed		Purpose
/get-all-books		See all book in library
Http Method	GET	
Parameter 1	-	
Return	<IEnumerable<Library>>	
/borrow-book		Borrow a Book from Library
Http Method	GET	
Parameter 1	BookId	
Parameter 2	BookBorrow borrow	
Return	<Book>	
/find-student-byname		Find a student by name or show all student
Http Method	GET	
Parameter 1	String(name)	
Return	<IEnumerable<Student>>	
/find-notice-byname		Find a notice by name
Http Method	GET	

Parameter 1	String(name)	
Return	<IEnumerable<Notice>>	
/get-all-teachers		Show list of teachers
Http Method	GET	
Parameter 1	-	
Return	<IEnumerable<Teacher>>	
/get-borrowbook-info		Book Borrow Information
Http Method	GET	
Parameter 1	BorrowId	
Return	<BookBorrow>	

4. BUSINESS VALIDATIONS

4.1 Entities:

- All the value for Student details: must not be null and min of 4 chars
- All the Library value: must not be null and min of 3 chars
- All the Teachers: must not be null and min of 3 chars.
- All the Notice: must not be null.
- All the Class: must not be null.

4.2 Common Constraints:

- For all receiving Url parameter, validation check must be done and must throw custom exception if data is invalid
 - Must not go and touch the test resources, as they will be used for Auto-Evaluation.
-

5. EXECUTION STEPS TO FOLLOW

1. All actions like build, compile, running application, running test cases will be through Command Terminal.
2. To open the command terminal the test takers need to go to the Application menu (Three horizontal lines at left top) Terminal → New Terminal.
3. On command prompt, cd into your project folder (**cd <Your-Project-folder>**).
4. (Schoolmanagement/**sqlcmd -S localhost -U sa -P pass@word1**)
 - To create database from terminal -
 1. **Create Database SchoolManagement_Db**
 2. **Go**
5. Steps to Apply Migration(Code first approach):
 - Press **Ctrl+C** to get back to command prompt
 - Run following command to apply migration- (Schoolmanagement /**dotnet-ef database update**)
6. To check whether migrations are applied from terminal:
(Schoolmanagement /**sqlcmd -S localhost -U sa -P pass@word1**)
 - 1> **Use SchoolManagement_Db**
 - 2> **Go**
 - 1> **Select * From __EFMigrationsHistory**
 - 2> **Go**
7. To build your project use command:
(Schoolmanagement /**dotnet build**)
8. To launch your application, Run the following command to run the application:
(Schoolmanagement /**dotnet run**)
9. This editor Auto Saves the code.
10. To test any Restful application, the last option on the left panel of IDE, you can find ThunderClient, which is the lightweight equivalent of POSTMAN.

11. To test any UI based application the second last option on the left panel of IDE, you can find Browser Preview, where you can launch the application.
 12. To run the test cases in CMD, Run the following command to test the application:
(You can run this command multiple times to identify the test case status, and refactor code to make maximum test cases passed before final submission)
(Schoolmanagement.Test/**dotnet test --logger "console;verbosity=detailed"**).
 13. If you want to exit(logout) and continue the coding later anytime (using Save & Exit option on Assessment Landing Page) then you need to use CTRL+Shift+B - command compulsorily on code IDE. This will push or save the updated contents in the internal git/repository. Else the code will not be available in the next login.
 14. These are time bound assessments the timer would stop if you logout and while logging in back using the same credentials the timer would resume from the same time it was stopped from the previous logout.
 15. You need to use CTRL+Shift+B - command compulsorily on code IDE, before final submission as well. This will push or save the updated contents in the internal git/repository, and will be used to evaluate the code quality.
-