
System Requirements Specification Index

For

School Management Application

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	
1	View All School Upcoming Notice.
2	Find a notice by name and event.
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 while collecting book from Library.

2. TEMPLATE CODE STRUCTURE

2.1 Package: SchoolManagement

Resource

Names	Resource	Remarks	Status
Package Structure/Project			
wwwroot	CSS, JS, Lib	Inside all these directory contains all styling logic code and javascripts file.	Already Implemented
controller	SchoolController	School controller handle all action method for respective user interface view(cshtml file)	Partially Implemented
Models	Error view model cs file	Contain all basic error definition	Already Implemented
Views	School, Home, Shared	All View .cshtml file for user interface.	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 CRUD 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			
DataGenerator	This cs class contain all dummy data for inMemory Operation	Create and initialize a basic data for application alive for demo.	Already Implemented
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.Tests

Resources

The **SchoolManagement.Tests** 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

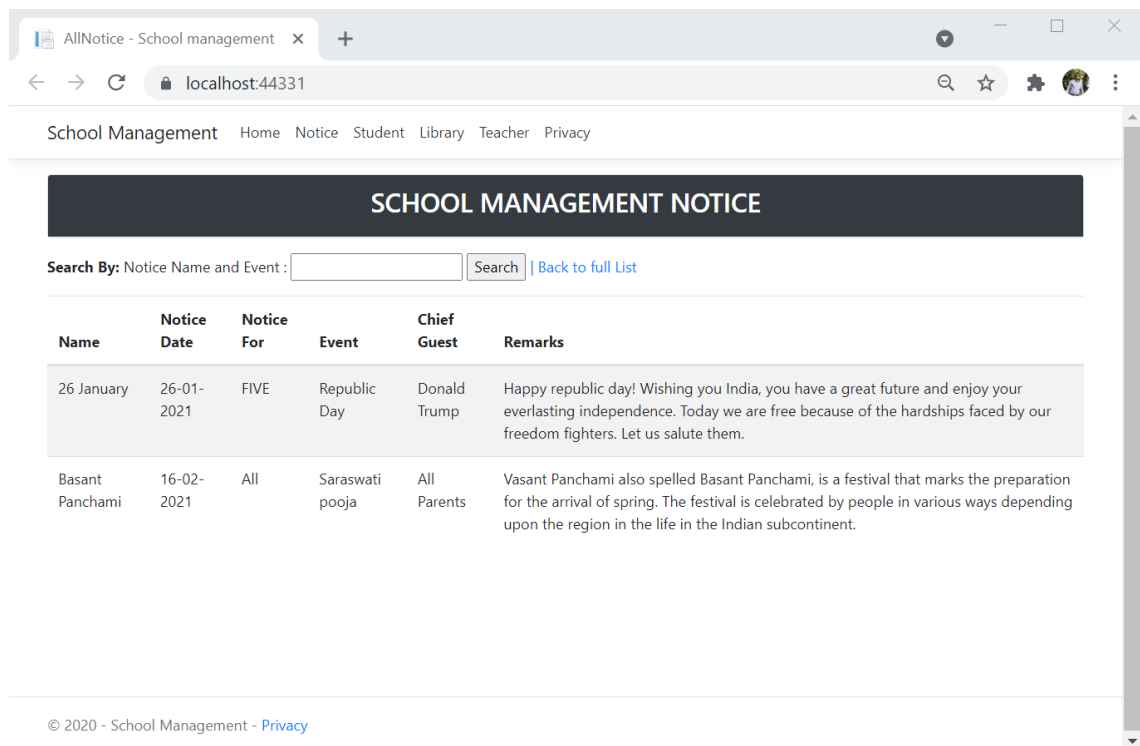
3.1 SchoolController

URL Exposed		Purpose
/School/Library		See all book in library and borrow
Http Method	GET	
Parameter 1	-	
Return	<IEnumerable<Book> >	
/School/BorrowBook?BookId=2		Borrow a Book from Library
Http Method	GET	
Parameter 1	BookId	
Return	<Book>	
/School/AllStudent		Find a student by name or show all student
Http Method	GET	
Parameter 1	-	
Return	<IEnumerable<Student>>	
/School/AllTeacher		Show list of teachers
Http Method	GET	
Parameter 1	-	
Return	<IEnumerable<Teacher>>	
/School /Thanks?BorrowId=1		Book Borrow Information
Http Method	GET	
Parameter 1	BorrowId	
Return	<Book>	

4. WIREFRAME

1. Below are expected wireframes for creating the application GUI, you can also add new as expected.
2. Navigate on page using Menu Bar.

Home/Index Page (Show all school notice)



Student List Page

The screenshot shows a web browser window with the address bar displaying 'localhost:44331/School/AllStudent'. The page title is 'All Student - School management'. The navigation bar includes 'School Management', 'Home', 'Notice', 'Student', 'Library', 'Teacher', and 'Privacy'. The main content area features a yellow header with the text 'STUDENT INFORMATION'. Below this, there is a search bar with the label 'Search By: Student By Name:' and a 'Search' button. A link 'Back to full List' is also present. The main table lists student information with columns: Name, Date of Birth, Class List, Section, Phone, and Father Name. The table contains two rows of data. The footer shows '© 2020 - School Management - Privacy'.

Name	Date of Birth	Class List	Section	Phone	Father Name
Uma Kumar	01-03-1990	TEN	A	9631438113	Gopal PD Singh
Sonoo Kumar	20-10-1994	NINE	B	9631432345	Gopal PD Singh

Library Page

The screenshot shows a web browser window with the address bar displaying 'localhost:44331/School/Library'. The page title is 'All Book - School management'. The navigation bar includes 'School Management', 'Home', 'Notice', 'Student', 'Library', 'Teacher', and 'Privacy'. The main content area features a green header with the text 'SCHOOL LIBRARY'. Below this, there is a table listing book information with columns: BookName, Publication, Writer, Stock, and Action. The table contains three rows of data. The footer shows '© 2020 - School Management - Privacy'.

BookName	Publication	Writer	Stock	Action
Work with Angular	Microsoft-Press	Yasvant K	6	Borrow Book
Microsoft Azure Development	Microsoft-Press	Marshall Copeland, David Golobde	5	Borrow Book
Deploying And Developing .Net core	Microsoft-Press	Tim Cook	10	Borrow Book

Borrow Book Page

Book Borrow - School management x +

localhost:44331/School/Borrowbook?BookId=2&class=btn%20btn-primary

School Management Home Notice Student Library Teacher Privacy

Borrow Book from Library

From Date To Date

25-09-2021 30-09-2021

Submit Back To Library

© 2020 - School Management - [Privacy](#)

Borrow Book info Page

Borrow Book Information - School x +

localhost:44331/School/Thanks?BorrowId=1

School Management Home Notice Student Library Teacher Privacy

Book Borrow information

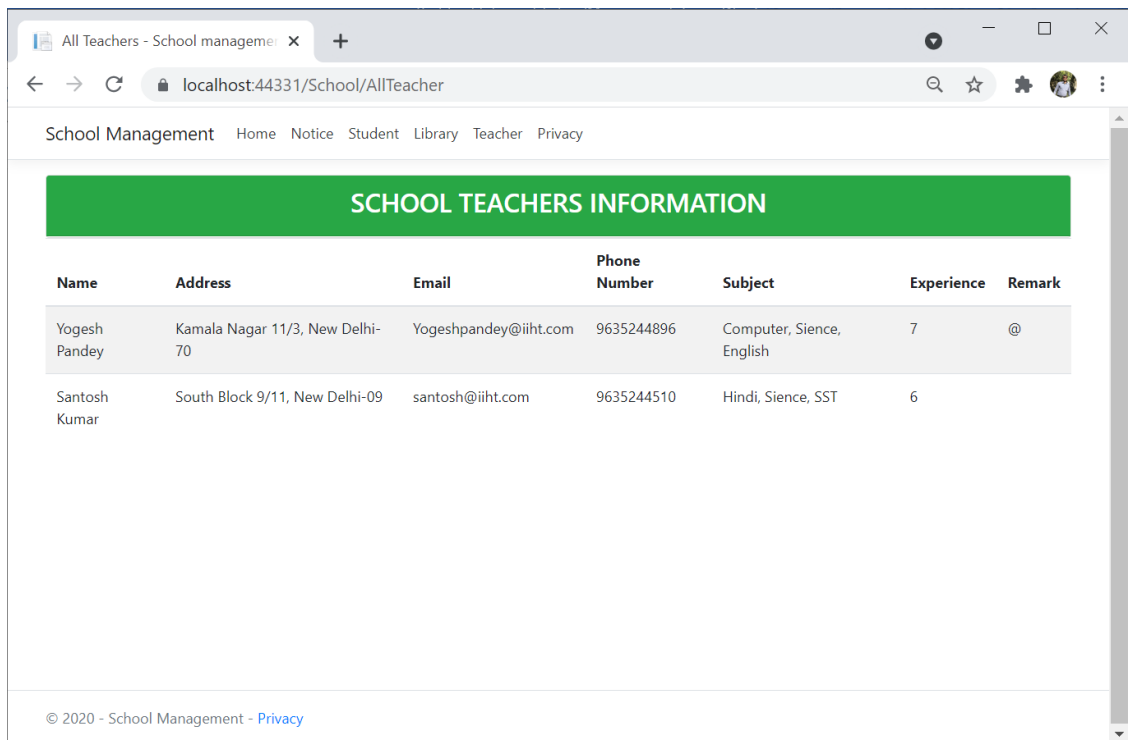
BorrowId	1	BookId	2
From Date	25-09-2021 00:00:00	To Date	30-09-2021 00:00:00

[Go To Home](#) [Back to Book List](#)

Note Your Borrow Infomation and use while collecting Book from Library

© 2020 - School Management - [Privacy](#)

AllTeacher Page



School Management Home Notice Student Library Teacher Privacy

SCHOOL TEACHERS INFORMATION

Name	Address	Email	Phone Number	Subject	Experience	Remark
Yogesh Pandey	Kamala Nagar 11/3, New Delhi-70	Yogeshpandey@iiht.com	9635244896	Computer, Science, English	7	@
Santosh Kumar	South Block 9/11, New Delhi-09	santosh@iiht.com	9635244510	Hindi, Science, SST	6	

© 2020 - School Management - [Privacy](#)

Note: All page navigation achieve using menu bar

5. BUSINESS VALIDATIONS

5.1 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.

5.2 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.
- **Database used is Embedded InMemory with pre-written entities data and value.**

6. CONSIDERATIONS

- Your code will also be evaluated for code quality, naming conventions, readability etc.
 - Make sure you do not modify existing class and method names and their signatures, else it would severely affect the final result.
 - Make sure you do not add any new class or methods, else it would severely affect the final result.
 - Make sure you do not modify any test cases, else it would severely affect final result
-

7. 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. To build your project use command:
(SchoolManagement / **dotnet build**)
5. To launch your application, Run the following command to run the application:
(SchoolManagement / **dotnet run**)
6. This editor Auto Saves the code.
7. 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.
8. 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.
9. To run the test cases in CMD, Run the following command to test the application:
(SchoolManagement / **dotnet test --logger "console;verbosity=detailed"**)
(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)
10. 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.

11. 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.

12. 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.
