

Summer Semester 2021

Datenbanken und WebTechniken

# Project Term Paper

SUBMITTED BY

Lalit Narayan Wazir

Matriculation # 676627

Study Program: MasterAutomotive Software engineering

July 05, 2021

PROJECT NAME

## School Grade Management System

This project corresponds to a web application with an aim to help a school to migrate its grading system from paper to digital. So that the administration, faculty and students can have better overview of the test results, available courses and the performance of all pupils.

School Grading System



TECHNISCHE UNIVERSITÄT  
CHEMNITZ

Page 1 of 14

## Table of Contents

<b>1 Technology Overview .....</b>	<b>3</b>
1.1 H2 Database.....	3
1.2 Java.....	3
1.3 Spring Boot .....	3
1.4 Spring JPA .....	3
1.5 Spring Security .....	3
1.5 React.....	3
1.6 Bootstrap.....	3
1.7 CSS.....	4
1.8 Material-UI.....	4
1.10 Eclipse and Visual Studio IDE .....	4
<b>2 Project: School Online Grading System .....</b>	<b>4</b>
2.1 Requirements .....	4
2.1.1 Requirements Breakdown .....	4
2.2 Requirement Implementation .....	7
2.2.1 Database .....	7
2.2.2 Functionality Explanation .....	7
2.3 Deployment .....	12
<b>3 Conclusion.....</b>	<b>15</b>
 Figure 1 : Project Structure	4
Figure 2:Database Diagram	7
Figure 3: Admin View: User Management	8
Figure 4:Admin View: Class Management	8
Figure 5:Subject Management & Pupil Assignment	9
Figure 6:Teacher Assigned Subjects Management	9
Figure 7:Teacher View: Assigned Subject And Test Management	10
Figure 8:Pupil View: List of all Assigned Subject with Average Grade	10
Figure 9: Pupil View – Viewing Test result of Selected Subject	11
Figure 10:Login Page	11
Figure 11:JWT Authentication in action -check authorization Header	12
 Table 1:Requirement Breakdown	6
Table 2:Required Packages for installation	13

## References

## Appendix

# 1 Technology Overview

## 1.1 H2 Database

H2 Database is an in-memory relational SQL database program. It comes integrated with spring boot project. It is open source software and also provides browser based console application which can be accessed through the following URL: <http://localhost:9000/h2-console/login.jsp?sessionId=7f8a06ea6be56992d2298754af34fc02> Please Note: In order to access this URL ,you need to start the backend code on your machine. The Main Motivation for choosing this Database was because it is an In Memory Database that is very easy to setup and deploy.

## 1.2 Java

Java 8 is used for developing the backend Rest API using Spring boot. You can get java 8 from the following link: <https://bit.ly/3hETmNk>. The main motivation behind choosing java for developing the backend Rest API is because it is the primary language for development using spring framework.

## 1.3 Spring Boot

Spring Boot 2.5.0 is used for developing the Backend Rest API because it provides a robust framework to implement security features like Authentication using JWT, Dependency Injection and Auto Configuration. Moreover, It makes the development of any Rest or Soap based API very easy. You can bootstrap a spring project template using spring Initializer( <https://start.spring.io> ) and for further information on spring boot please refer: <https://spring.io/projects/spring-boot>.

## 1.4 Spring JPA

Spring JPA is part of Spring Data which Is part of spring framework. It is a library/framework that adds an extra layer of abstraction on the top of our JPA provider (like Hibernate).It reduces boilerplate code and helps in mapping Java Object to SQL Table in Relational Databases. Furthermore, Also provides built in support for running complex queries in the database using @Query Annotation and JPA Repositories.

## 1.5 Spring Security

Spring Security is part of Spring framework. It is authentication and Access Control framework. In this project it is used to develop Authentication using JWT and to implement access control for users belonging to the one of the following roles: Teacher, Pupil and Admin. For more information on spring security, please refer: <https://spring.io/projects/spring-security>

## 1.5 React

React is a declarative, efficient and component based JavaScript Library for building User Interface. It p The main motivation for using it in this project was to learn about this popular framework and build user interface for this application using it.

For more Details on this Framework , Please refer: <https://reactjs.org>

## 1.6 Bootstrap

Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development [24]. This component is used to simplify the user interface development and to facilitate an appealing look to the application. The version 4.0 is used in the application

## 1.7 CSS

CSS is the language used to style Html documents and described how they should be displayed on the screen. It stands for Cascading Style sheets. In this project CSS3 is being used.

## 1.8 Material-UI

Material-UI Is a library used to develop Responsive React UI Components. So that the react UI components can adapt to changing screen sizes dynamically. It also provides robust framework for different built in components like Button, grids, Alert, Dialog, Snack bars etc. It Is available at <https://material-ui.com/getting-started/usage/>

## 1.10 Eclipse and Visual Studio IDE

In this project Eclipse IDE is used for developing java based spring boot backend Rest API and Visual Studio is used for developing the frontend react based application.

Both are Open Source Software.

Eclipse is available at : <https://www.eclipse.org/downloads/>

Visual Studio is available at : <https://code.visualstudio.com/download>

# 2 Project: School Online Grading System

## 2.1 Requirements

The top-level requirements for the project include a backend which is capable of providing, storing and processing data received or sent to the frontend. While processing, a database is used to store the aggregated data which can be later accessed by the end user using an interactive frontend. Frontend consumes the data from data store using a REST based Web Service. The diagram below describes the scenario even better.



Figure 1 : Project Structure

### 2.1.1 Requirements Breakdown

The table below illustrates the requirements in more detail.

APPLICATION STACK	TODOs	REMARK
Front End	UI to allow Users to login to the application with their username and password.	Implemented JWT Token Based authentication
	Users should belongs either one of the following roles: Admin, Teacher and Pupil and after login should continue with their corresponding role	
	UI for Admin users	<ol style="list-style-type: none"> <li>1. List all available users</li> <li>2. Add new user <ul style="list-style-type: none"> <li>• User is assigned a username, password, first name and last name on creation.</li> <li>• Each user is identified by unique user id generated by backend.</li> </ul> </li> <li>3. Delete user</li> <li>4. Edit user attributes</li> </ol>
	UI for classes / Course management for admin users	<ol style="list-style-type: none"> <li>1. List all available classes/courses</li> <li>2. Add new course</li> <li>3. Assign and de-assign pupils</li> <li>4. Remove class</li> </ol>
	UI for management for subject in Admin View for each course.	<ol style="list-style-type: none"> <li>1. List all available subjects</li> <li>2. Add new subject</li> <li>3. Edit subject attributes</li> <li>4. Remove a subject archive subject.</li> </ol>
	UI For teacher View	UI for listing all assigned subject of a teacher
	Teacher View: UI for Listing all pupil with their average grade on selecting an assigned subject by teacher in teacher view	<ol style="list-style-type: none"> <li>1. Ui For adding new test</li> <li>2. UI for editing test attributes like name.</li> </ol>
	UI for listing all pupils with their grade in a test in teacher view	<ol style="list-style-type: none"> <li>1. Ui for add or change of grades of one pupil in the test.</li> <li>2. Ui for removing test</li> </ol>

	UI for Pupil View	<ol style="list-style-type: none"> <li>1. Ui for listing all assigned subjects of pupil with their average grade</li> <li>2. UI for listing all test with their grades corresponding to a selected assigned subject.</li> </ol>
<b>Back End REST Interface</b>	Storing password in hashed format using bcrypt password hashing java library	
	<p>Implemented Rest API Endpoint for adding, editing and removing users. Rest</p> <p>Implemented endpoint to export pdf of list of users for admin view</p>	
	Implemented JWT authentication layer for accessing the Rest API.	
	Implemented Rest Endpoints for course, subject, and test management	<p>Endpoints for:</p> <ol style="list-style-type: none"> <li>1. Fetching, Adding, editing deleting course.</li> <li>2. Fetching, Adding deleting and editing subjects.</li> <li>3. Fetching, adding, deleting and editing tests.</li> </ol>
	Expose all functionalities as REST API	A set of endpoints
<b>Database</b>	Conceptualize the Schema	

Table 1:Requirement Breakdown

## 2.2 Requirement Implementation

### 2.2.1 Database

This application highly depends on the student data. However, the diagram below shows the minimal conceptual schema supporting the current implementation.

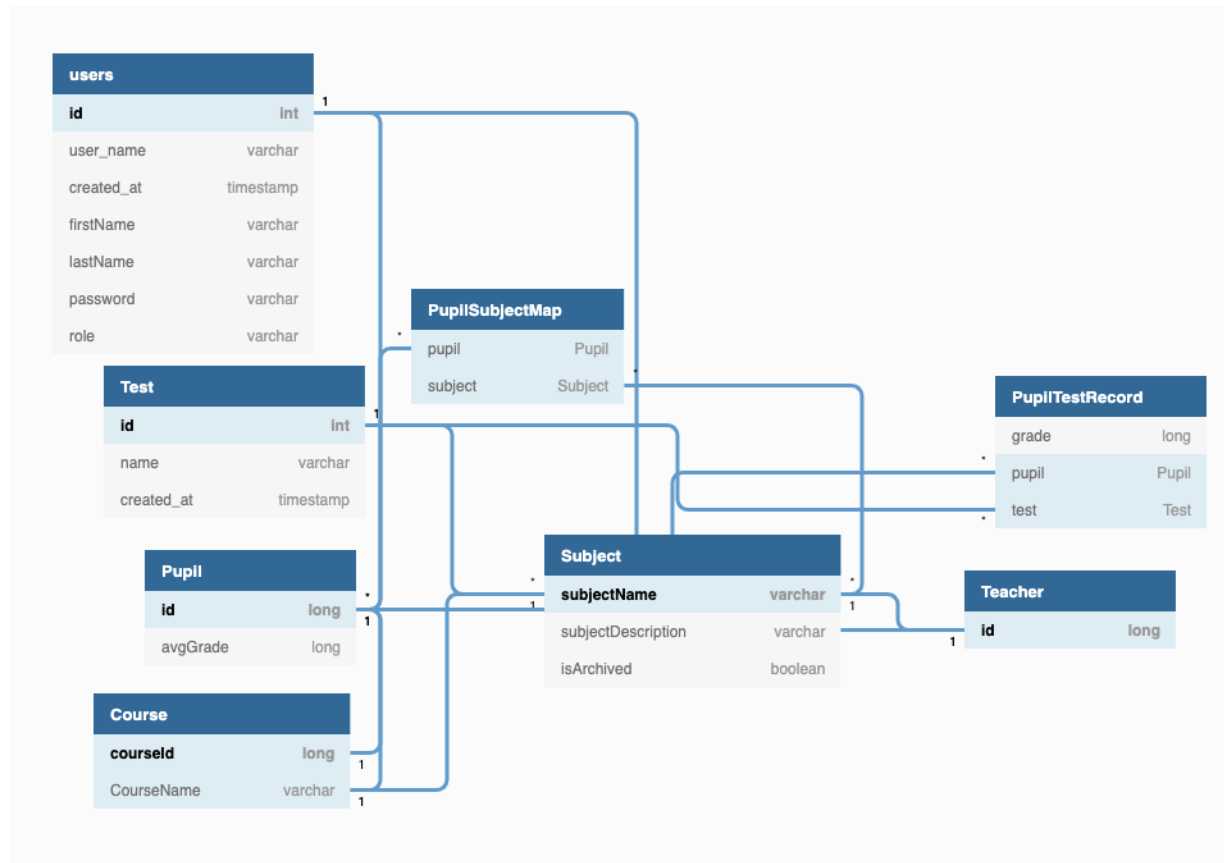


Figure 2: Database Diagram

### 2.2.2 Functionality Explanation

This section contains the detail of some basic logics which are applied to implement the core part of the application.

#### Task 1: Storing data in H2 Data Base and password in hashed format

All Data is stored in Database and is accessed only via the rest API. The frontend can't directly access it.

All Password are stored in hash format using BCrypt password Hashing algorithm

#### Task 2: Admin View: User Management

- List all users
- Add new user
- Edit user
- Remove user

Users:					
Id	User	Role	Created At	Delete	Update
6	agoyal	PUPIL	2021-07-04T22:00:00.000+00:00	Delete	Update
7	sushukla	PUPIL	2021-07-04T22:00:00.000+00:00	Delete	Update
8	teacher	TEACHER	2021-07-04T22:00:00.000+00:00	Delete	Update
9	hwazir	ADMIN	2021-07-05T11:56:13.689+00:00	Delete	Update
10	username2	PUPIL	2021-07-05T11:56:13.689+00:00	Delete	Update
11	username3	PUPIL	2021-07-05T11:56:13.689+00:00	Delete	Update
12	username4	PUPIL	2021-07-05T11:56:13.689+00:00	Delete	Update
13	username5	PUPIL	2021-07-05T11:56:13.689+00:00	Delete	Update

New User Export

Figure 3: Admin View: User Management

## Task 2.2: Admin View: Course/ Class Management

- List all available classes
- Add new class:
- Edit class name
- Remove a class

Course Id	Course Name	Delete	Update	View Details
1	Master in Automotive Software Engineering	Delete	Update	View Details
2	Master in Web Engineering	Delete	Update	View Details
3	Master in Computer Science	Delete	Update	View Details
4	Bachelors in Information Technology	Delete	Update	View Details
5	Masters in Data Science	Delete	Update	View Details

New Course Export

Figure 4: Admin View: Class Management

## Task 2.3: Admin View: Subject Management

- List available subjects on selecting a course.
- Add Subject
- Edit Subject



- d) Remove subject
- e) Assign and de-assign pupil to class
- f) Archive subject

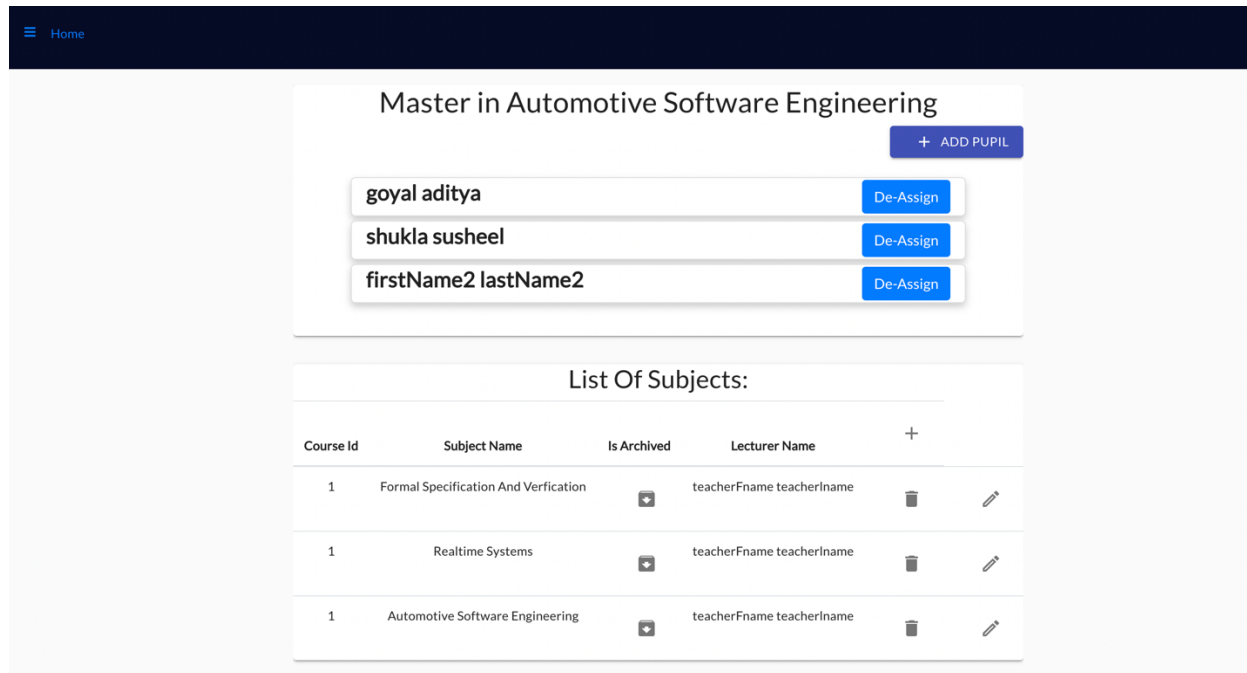


Figure 5:Subject Management & Pupil Assignment

### Task 3: Teacher View:

This view provides the following features:

- a) List of assigned subjects

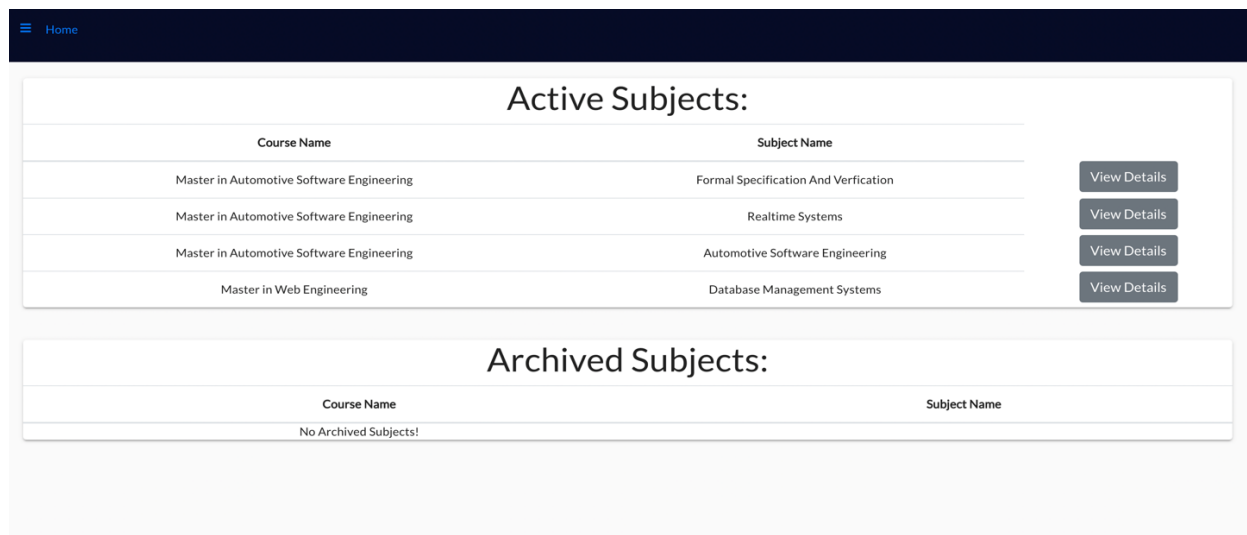


Figure 6:Teacher Assigned Subjects Management

- b) Selection of one of the assigned subjects and afterwards management of tests:
- School Grading System

- List all pupils with their average grade
- Add new test
- Edit test
- List all pupil along with their grade in the test
- Remove test

Home

Formal Specification And Verification

Tutor: teacherFname teacherIName  
Description: No Description  
Archived: No

Pupil Id	Pupil Name	Avg Grade
1	goyal aditya	11
2	shukla susheel	0
3	firstName2 lastName2	0

+ ADD TEST

Test Id	Test Name	createdAt			
1	Test 1	05-07-2021 16:26:04			

Figure 7:Teacher View: Assigned Subject And Test Management

## Task 4: Pupil View

Pupil View has the following features:

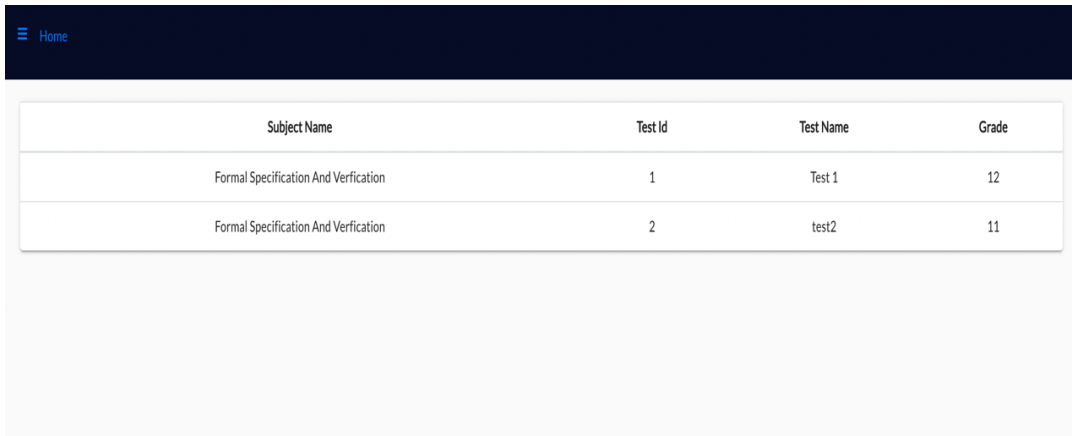
- List All assigned Subjects along with their average test grade

Home

Subject Name	Average Grade	
Automotive Software Engineering	0	
Data Structure and Algorithms	0	
Design of Software for Embedded Systems	0	
Formal Specification And Verification	11.375	
Realtime Systems	10	

Figure 8:Pupil View: List of all Assigned Subject with Average Grade

- Select one assigned subject and list all corresponding tests along with their grades



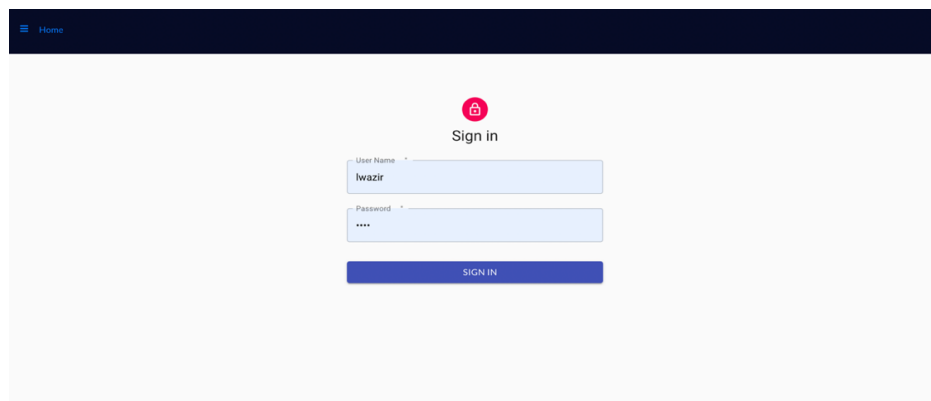
The screenshot shows a web application interface with a dark blue header containing a menu icon and the word "Home". Below the header is a table with four columns: "Subject Name", "Test Id", "Test Name", and "Grade". The table contains two rows of data, both for the subject "Formal Specification And Verification".

Subject Name	Test Id	Test Name	Grade
Formal Specification And Verification	1	Test 1	12
Formal Specification And Verification	2	test2	11

Figure 9: Pupil View – Viewing Test result of Selected Subject

## Task 5 : Users start with Logging into the application

In here , in addition to using username and password authentication. I am also using JWT Authentication by requesting the server for JWT token and then sending it in s authentication header in every subsequent request from the frontend to backend as shown below:



The screenshot shows a login page with a dark blue header containing a menu icon and the word "Home". In the center, there is a "Sign in" section with a red lock icon. Below the icon are two input fields: "User Name" with the value "lwazir" and "Password" with masked characters "\*\*\*\*". A blue "SIGN IN" button is positioned below the password field.

Figure 10:Login Page

```

Request Headers    View source
Accept: application/json, text/plain, */*
Accept-Encoding: gzip, deflate, br
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8,de;q=0.7
authorization: Bearer eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJlc2VybmFtZTU1LjYyY2xlc2I6W3siYXV0aG9yaXR5Ijo1UjVQSUwifV0sImV4cCI6MTYyNjEwMDYyMCwiaWF0IjoxNjI1NDk1ODIwfQ.L0vX5FDCEtRT003C6oDnrLtNHihZ0QcgfyRwcUWPczWRfCdJMIoBEQYkmtG0SA5m5UyRBY-Rjm32FGH0E5NlGA
Connection: keep-alive
Host: localhost:9000
Origin: http://localhost:4500
Referer: http://localhost:4500/
sec-ch-ua: " Not;A Brand";v="99", "Google Chrome";v="91", "Chromium";v="91"
sec-ch-ua-mobile: ?0
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-site
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.114 Safari/537.36

```

Figure 11:JWT Authentication in action -check authorization Header

## 2.3 Deployment

### 2.3.1 Package Overview

The application is distributed as a zip archive (schoolmanagementSystem.zip) which travels along with this document. After extracting the zip file two directory titled “frontend” and “backend” can be noticed. “frontend” refers to the frontend application and “backend” serves as the backend Spring boot REST API. A database is not needed to be created since it is handled through backend itself. A glimpse of the package is illustrated below for reference.

#### SchoolManagementSystem

```

|__ frontend          [ frontend ]
|  |__ README.md
|  |__ src
|  |__ public
|  |__ package.json
|
|__ backend           [ backend ]
|  |__ schoolManagementsystem
|     |__ pom.xml
|     |__ src
|         |__ main.java.com.lwazir.project.schoolManagementsystem
|             |__ controller
|             |__ errors
|             |__ jwt
|             |__ model
|             |__ repository
|             |__ utility
|             |__ SchoolManagementsystemApplication.java. [entry point]

```

```

|__ README.md      -   [Contains installation instructions for both backend and frontend]
|__ Installation Instruction. [ Same as above ]

```

## 2.3.2 Installation and Prerequisites

Installation is as simple as extracting the zip archive to somewhere in the disk.

APPS	REQUIRED PACKAGES
<b>Front end</b>	<p>All required Dependency are mentioned in the package.json file present in the frontend folder.</p> <p>The only prerequisites is to install NPM and Node.</p> <p>Once it is installed, please follow the instructions as mentioned in the ReadMe Document attached with this file</p>
<b>backend</b>	<p>All Required Dependencies are mentioned in the pom.xml file</p> <p>They will automatically get downloaded as soon as the backend code is imported into the eclipse IDE as existing maven project and the SchoolManagementsystemApplication.java file is run as java application. Whereupon the backend will be deployed at localhost:9000</p> <p>However, the installation of java 8 and Eclipse IDE is prerequisite for running backend.</p>

*Table 2:Required Packages for installation*

## 2.3.3 Configure

Both Applications have some default configuration and they are as follows. Both application willtake care of these nothing required from user.

### Configuration Parameters: Backend

The following configuration can found in application.properties file located in src/main/resources folder.

SERVER\_HOST=localhost  
SERVER\_PORT=9000  
H2\_Database Console url=http://localhost:9000/h2-console/login.jsp  
H2 username =sa  
H2 password=<blank>  
API Documentation = http://localhost:9000/swagger-ui.html

### Configuration Parameters: Frontend

The following configuration are configured in package.json file in frontend folder

SERVER\_HOST=localhost  
SERVER\_PORT=4500  
Application URL=http://localhost:4500/

### **2.3.4 How to Run**

Please refer the ReadMe Instruction file or Installation\_instruction.doc for running both frontend and backend code. These files can be found in root folder of the unzipper folder.

Otherwise, you can also refer the following video for running both frontend and backend application here is the link to the video:

[https://www.youtube.com/watch?v=\\_HTMtAdfZyo](https://www.youtube.com/watch?v=_HTMtAdfZyo)

### 3 Conclusion

The overall development experience of School Management System was absolutely satisfying. Especially the opportunity to explore the React and Spring boot world to a further extent is worth stating. Working with H2 Database and Spring JPA was another fun part. As it helped me to learn about the relational databases and ORM Frameworks in the context of a practical application. Furthermore, developing the JWT- based Authentication feature using Spring Security was very fun along with developing feature for exporting list Of user into a pdf was also very interesting. However, The well-defined set of requirements made it even helpful to deal with some misperceptions during the development period. But, as a concluding statement it can be confidently declared that, the this School Management System is strong prototype that can be used in actual production with some more improvements with the UI.As I couldn't do it due to shortage of time.

## References

- [1] Richter D. (2021, July 05). "DBW Project" [PDF File]. Retrieved from <https://www.tuchemnitz.de/informatik/DVS/lehre/DBW/Projekt/DBW%20Project.pdf>
- [2] Richter D. (2021 July 05). "Questions and Answers concerning project task in Datenbanken und Web-Techniken 2021" [Text File]. Retrieved from <https://www.tuchemnitz.de/informatik/DVS/lehre/DBW/Projekt/DBW%20Project%20QA.txt>
- [3] Spring Boot Tutorial. By in28Minutes (2021, July 5). Retrieved from <https://www.springboottutorial.com>
- [4] Stack Overflow (2021, July 05) Retrieved from <https://stackoverflow.com>
- [5] React Tutorial. (2021, July 05). Retrieved from <https://reactjs.org/tutorial/tutorial.html>
- [6] Material-Ui Components (2021, July 05). Retrieved from <https://material-ui.com>
- [7] CSS3 Tutorial. (2021, July 05). Retrieved from: <https://www.w3schools.com/css/>
- [8] Introduction on Spring JPA (2021, July 05).Retrieved from <https://www.baeldung.com/the-persistence-layer-with-spring-data-jpa> .
- [9] Creating PDF with Java Itext library(2021, July 05). Retrieved from <https://www.vogella.com/tutorials/JavaPDF/article.html>
- [10] Bootstrap Tutorial (2021 July 05). Retrieved from <https://www.w3schools.com/bootstrap4/>
- [11] Documenting Spring boot Rest Api using Open Api. (2021, July 05). Retrieved from <https://www.baeldung.com/spring-rest-openapi-documentation>
- [12] Creating Database Diagram (2021 July 05). Retrieved from <https://dbdiagram.io/home>
- [13] BootStrapping Spring project (2021 July 05). Retrieved from <https://start.spring.io>



# Appendix

## API Documentation

Please refer the following link to access the API Documentation: <https://bit.ly/3xulokc>

Otherwise, This API documentation is also available on the following url:

<http://localhost:9000/swagger-ui/index.html?configUrl=/v3/api-docs/swagger-config>. However ,

Please note that this URL can only be accessed when one is running the Spring boot backend application on his machine.

-- End of Document --