# Introduction

Learner’s Academy is a school that has an online management system. As part of this project, we aim to develop a backend administrative portal that will enable the school to efficiently manage its classes, subjects, students, and teachers. With this system in place, the school will be able to keep track of all its resources in a more organized and streamlined manner.

The proposed backend admin portal will be designed and developed by a full stack developer who will utilize a GitHub repository to manage the project artifacts. The system will have a single administrator login who will have access to various features that will allow them to manage the school’s resources.

The administrator will be able to set up a master list of all subjects for all classes, set up a master list of all teachers, set up a master list of all classes, assign classes for subjects from the master list, assign teachers to a class for a subject, and get a master list of students.

The proposed system will also have an option to view a class report that will show all the information about the class, such as the list of students, subjects, and teachers. With this feature, the school will be able to easily generate reports on the performance of different classes, subjects, and teachers.

Overall, the goal of this project is to deliver a high-quality product that will meet the needs of Learner’s Academy as early as possible. By creating an efficient and organized system, we aim to provide the school with the tools it needs to manage its resources effectively and enhance the learning experience of its students.

# Developer Details

Amit Kumar Shah is a Full Stack Java Bootcamp student who has recently completed a comprehensive course in Full Stack Development. With a strong foundation in Java programming and front-end technologies such as HTML, CSS, and JavaScript, Amit has the skills and knowledge required to develop robust and scalable web applications.

Amit is passionate about programming and always strives to stay up-to-date with the latest technologies and best practices in the field. He is highly motivated and possesses excellent problem-solving skills, which allows him to effectively identify and resolve issues in a timely manner.

Throughout his training, Amit has gained hands-on experience working on various projects, including building responsive websites, developing dynamic web applications, and implementing complex algorithms. He has also worked on several group projects, where he has demonstrated strong team collaboration and communication skills.

Amit's technical skills include proficiency in Java, Spring Boot, Hibernate, React, Angular, and Node.js. He is also well-versed in database design and management, as well as version control using Git.

If you would like to learn more about Amit's skills and experience, or discuss potential project opportunities, please don't hesitate to reach out to him via email at [amitshah99m@gmail.com](mailto:amitshah99m@gmail.com).

# Sprint Planning

Sprint 1:

* Set up the database schema for the project, including tables for subjects, teachers, classes, and students.
* Implement a backend servlets for adding and retrieving subjects, teachers, and classes from the database.
* Implement a frontend interface for the administrator to view and manage subjects, teachers, and classes.

Sprint 2:

* Implement a backend servlets for assigning subjects to classes and assigning teachers to classes for specific subjects.
* Implement a frontend interface for the administrator to assign subjects to classes and teachers to classes for specific subjects.
* Implement a backend servlets for retrieving a master list of students and assigning students to specific classes.
* Implement a frontend interface for the administrator to view and manage students and their class assignments.

# Concepts used in the project

Java Servlets are a Java-based technology that allows developers to create web applications that run on a web server. Servlets are server-side programs that can handle client requests and generate dynamic content. They can be used to create applications such as e-commerce websites, online banking systems, and social networking platforms.

JSP (JavaServer Pages) is another Java-based technology that allows developers to create dynamic web pages. JSP pages are server-side programs that can generate HTML, XML, or other types of documents. They can be used to create templates for web pages that are dynamically generated based on user input or other data sources.

MySQL is an open-source relational database management system (RDBMS) that is commonly used in web development. It provides a reliable, scalable, and secure way to store and manage large amounts of data. MySQL is often used in conjunction with Java Servlets and JSP pages to store and retrieve data from a database.

When used together, Java Servlets, JSP pages, and MySQL can be used to create powerful and dynamic web applications. For example, a web application that allows users to search for and purchase products online could be created using these technologies. The Java Servlets could handle user requests, the JSP pages could generate dynamic HTML pages based on the user's input, and MySQL could store and retrieve data about the products and transactions.

In Java web development, the doGet and doPost methods are two of the most important methods that are used to handle HTTP GET and POST requests. These methods are implemented in the HttpServlet class, which is a Java class that provides a framework for handling HTTP requests and responses.

The doGet method is used to handle HTTP GET requests, which are requests that retrieve data from the server. This method is typically used to display information to the user or to retrieve data from a database. The doPost method, on the other hand, is used to handle HTTP POST requests, which are requests that send data to the server. This method is typically used to submit forms or to upload files to the server.

When a client sends an HTTP request to the server, the request is processed by an instance of the HttpServlet class. The servlet uses the HttpServletRequest and HttpServletResponse classes to get information about the request and to send a response back to the client. The HttpServletRequest class provides methods for getting information about the request, such as the request method (GET or POST), the request URL, and the request parameters. The HttpServletResponse class provides methods for sending a response back to the client, such as setting the response status code, setting response headers, and sending response content.

The HttpServlet class provides a framework for implementing these methods, and it also provides other methods that can be used to handle other types of HTTP requests, such as HTTP PUT and DELETE requests. The HttpServlet class can also be extended to create custom servlets that provide additional functionality beyond what is provided by the built-in methods.

In Java, attributes are used to store data that can be accessed by different parts of an application. One way to set attributes is to use the setAttribute method, which is available in several classes such as HttpServletRequest, HttpSession, and ServletContext. This method takes a key-value pair as input, where the key is a string that identifies the attribute, and the value is the data to be stored. Other methods, such as getAttribute and removeAttribute, can be used to retrieve or remove attributes as needed. Attributes are often used to store data that needs to be shared between different components of a web application.

In Java, the SQL Connection, ResultSet, and PreparedStatement are three classes that are commonly used when working with databases. These classes are part of the JDBC (Java Database Connectivity) API, which provides a standard way for Java applications to interact with databases.

The SQL Connection class represents a connection to a database. When a connection is established, it allows the Java application to send SQL statements to the database and to receive results back. The Connection class provides methods for creating Statement and PreparedStatement objects, which can be used to execute SQL queries and updates.

The ResultSet class is used to retrieve data from a database after a SQL query has been executed. When a query is executed using a Statement or PreparedStatement object, a ResultSet object is returned that contains the results of the query. The ResultSet class provides methods for navigating through the results, retrieving data from individual rows, and getting metadata about the results.

The PreparedStatement class is used to execute parameterized SQL statements. A parameterized statement is a SQL statement that contains placeholders for data values, which are filled in at runtime. The PreparedStatement class provides methods for setting the values of these parameters, which helps to prevent SQL injection attacks and allows for more efficient execution of multiple similar queries.

In order to use these classes, the Java application must first establish a connection to the database using a driver that is specific to the type of database being used. Once the connection is established, the application can use the Connection class to create Statement or PreparedStatement objects, and can use these objects to execute SQL queries and updates.