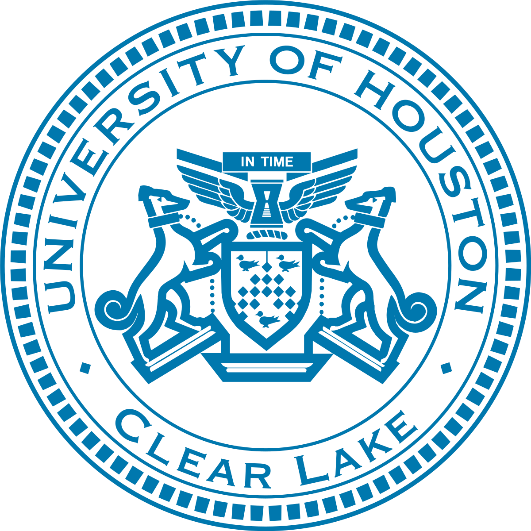
Primary University Logos | Downloads | Marketing and Communications |  University of Houston-Clear Lake

**CSCI 5333: Database Management Systems (DBMS)**

**(Fall 2023)**



**Project**

**Integrated Delivery System**

**Programmer Manual**

***Group Name:*** TechShip Innovators

***Group Members:*** Balaji Bojadla (2286913),

Pranavi Kolagani (2146799)

# **Introduction**

In the pursuit of creating a robust and efficient database system for the Integrated Delivery System project, **PostgreSQL** emerged as the cornerstone of our implementation. PostgreSQL, renowned for its open-source nature, extensibility, and robust feature set, provided the ideal platform to structure and manage the intricate web of data integral to our project. Leveraging the power of PostgreSQL, we aimed to build a database that seamlessly integrates user registration, package management, and delivery tracking functionalities. This report delves into the details of our database implementation, elucidating the design choices, schema structures, and the utilization of PostgreSQL's features to realize a scalable and secure solution for the complex requirements of an integrated delivery system.

Database Server: **PostgreSQL**

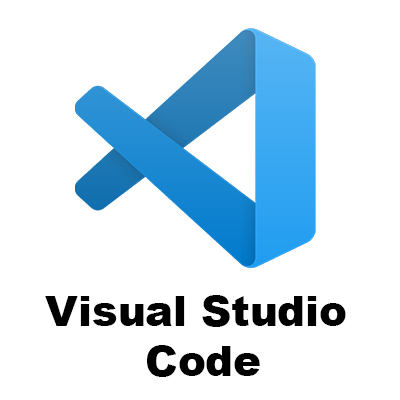
Frontend Technology: **Vue.js**

Backend Technology: **Springboot**

Server: (Embedded) **Tomcat Server**

Database Name: **ids**

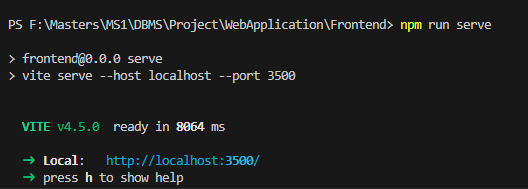
## **Frontend**

We have used Vue.js to develop the frontend. We have used the Visual Studio Code IDE to develop the user interface.

To start the frontend vue.js server we need to open the terminal(cmd) in the project Frontend directory and enter the following command: **npm run serve**



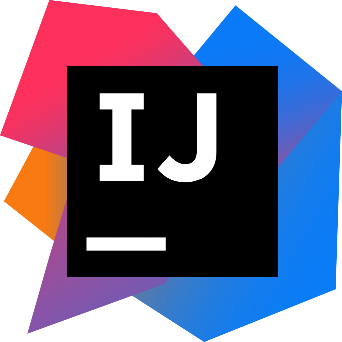
Now the server will be started, and you will see the port number as below referring on which port the server started.



If you see the above information in your terminal, then the serve has successfully started. You can navigate to the [**http://localhost:3500/**](http://localhost:3500/)to see the user interface of the website.

## **Backend**

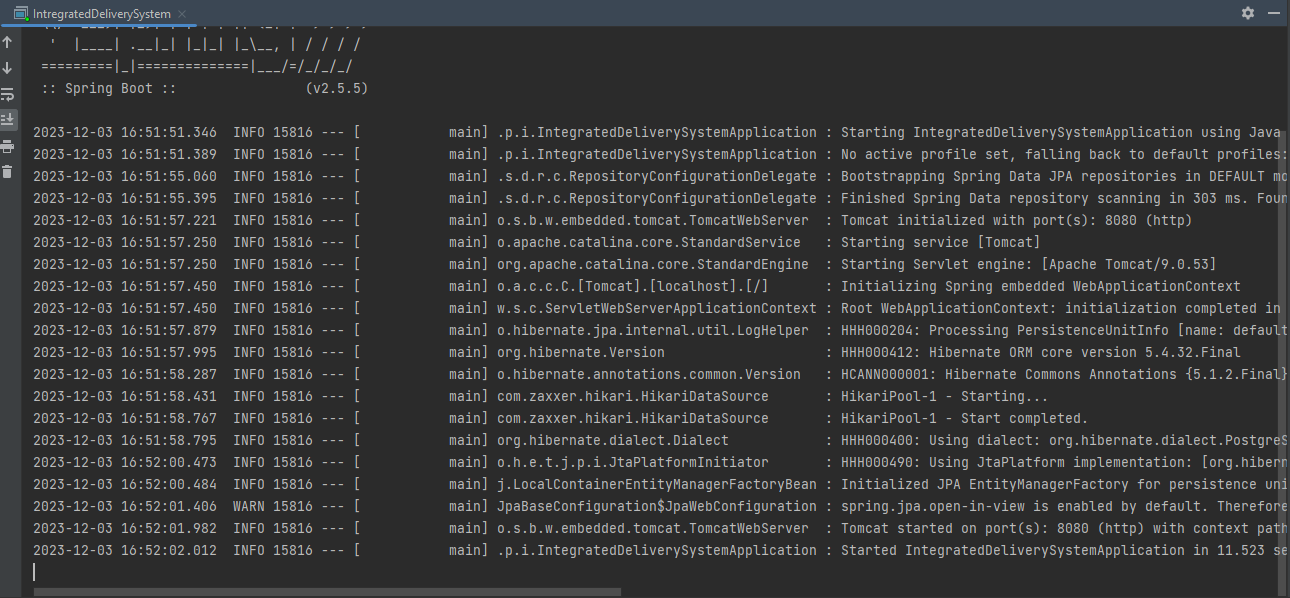
To develop the backend of the web application we have used **Springboot** technology. For this we have used the IDE IntelliJ Community version.



To Start the backend server you just need to open the project backend folder in the IntelliJ IDE and click run button of top right corner.



Now you will see the Terminal opens automatically, do some configuration work for you and starts the backend server.



If you see the terminal screenshot you can observe that the server has successfully started on port number 8080

A screen shot of a computer

Description automatically generated

## **Database**

We have use **PostgreSQL** to create the database.



## **Connecting PostgreSQL Database to Backend**

1. **Add dependency for PostgreSQL - Spring Data JPA**

Declare the following dependency in your project’s pom.xml file:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | <**dependency**>      <**groupId**>org.postgresql</**groupId**>      <**artifactId**>postgresql</**artifactId**>      <**scope**>runtime</**scope**>  </**dependency**>  <**dependency**>      <**groupId**>org.springframework.boot</**groupId**>      <**artifactId**>spring-boot-starter-data-jpa</**artifactId**>  </**dependency**> |

1. **Configure Data Source Properties**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | spring.datasource.url=jdbc:postgresql://localhost:5432/ids  spring.datasource.username=postgres  spring.datasource.password=Password@123  spring.jpa.hibernate.ddl-auto=update  spring.jpa.show-sql=true  spring.jpa.properties.hibernate.format\_sql=true  spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.PostgreSQL81Dialect |

Next, specify database connection information in the Spring Boot application configuration file (application.properties) as follows:

The Database server is running on the port number: 5432

**A screenshot of a computer

Description automatically generated**