CEN-4910C

Request For Employee Development Funds

Project Team

Natasha Phyle

Nicolo Perrelli

Adam Hunt

Craig Mackey

Mariele Felipe Pinheiro Rosa

Fernando Jiminez

Alex Visco

Table of Contents

Purpose...........................................................................................................................................3

Development Environment............................................................................................................3

IDE.......................................................................................................................................3

IDE Soruces.........................................................................................................................3

Persistence........................................................................................................................................3

Hibernate..........................................................................................................................................4

What is Hibernate................................................................................................................4

Why use Hibernate...............................................................................................................4

Maven..............................................................................................................................................4

TomCat............................................................................................................................................4

Database...........................................................................................................................................4

Populate Database................................................................................................................5

Dynamic Website.............................................................................................................................8

How to Configure IntelliJ....................................................................................................8

Pom.xml...............................................................................................................................10

Create Persistence.xml.......................................................................................................11

Configure Persistance.xml.................................................................................................11

How to Configure TomCat................................................................................................12

Connect to Database..........................................................................................................13

Index.jsp.............................................................................................................................14

What is a Servlet................................................................................................................14

How to Configure Servlet..................................................................................................14

Entity..................................................................................................................................15

Purpose

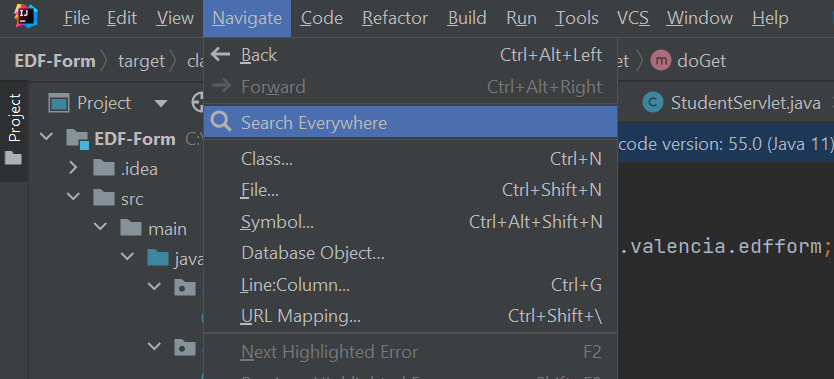
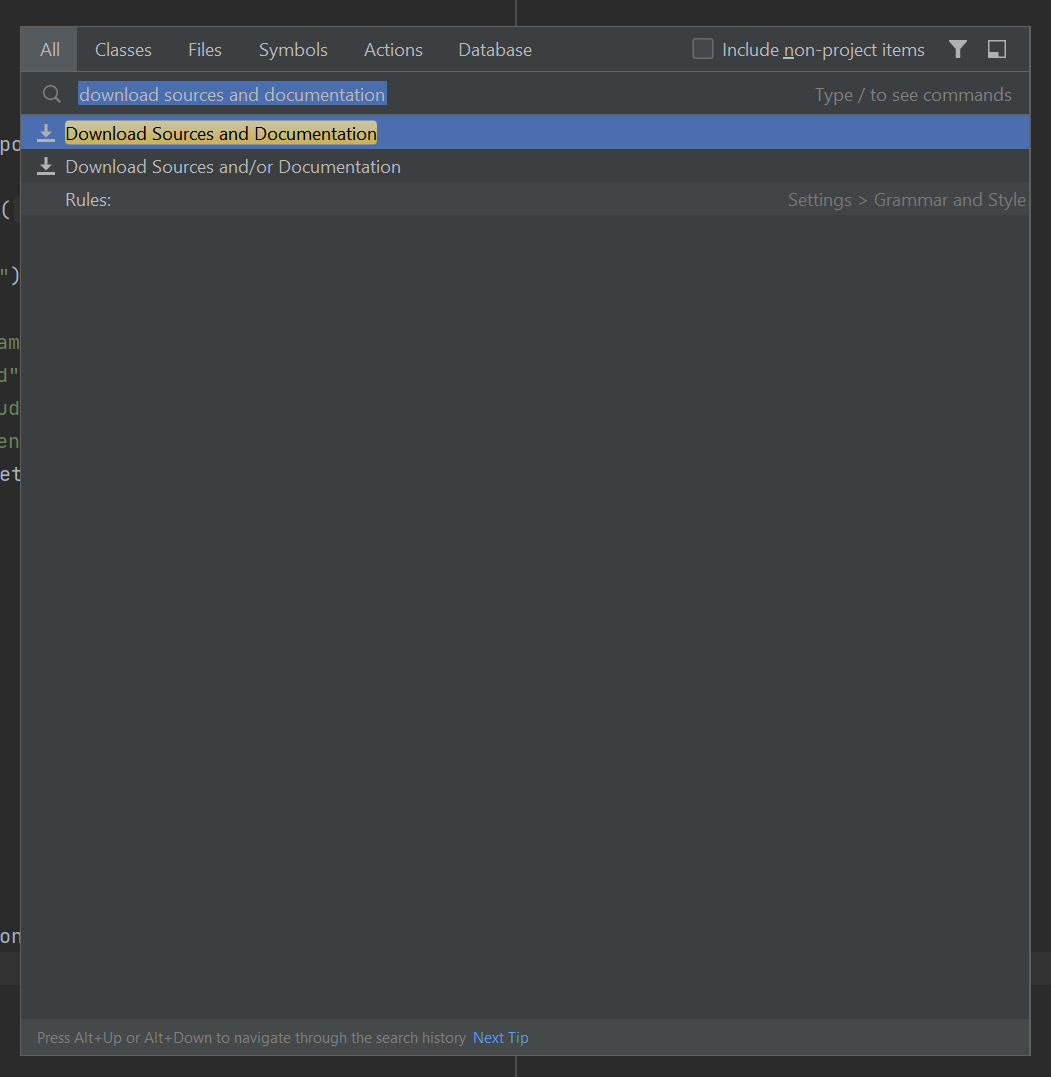
This project will decrease the amount of time and hassle involved with submitting a request for Employee Development Funds (EDF) while increasing the amount of security involved. Currently at Valencia if a student wanted to submit a EDF form they would need to find then print the form, fill it out with a pen or pencil, submit it, then wait for multiple administrators to scan and approve the form. This project allows students to quickly, easily access, fill out the form online while knowing that all information entered will be kept private, and reach the correct administrator.

Development Environment

IDE

Our team decided on using Intelligent Ultimate because of its built in enterprise tools, which helps when tracking issues, testing framework, and using server automation.

IDE Sources

To install all the necessary sources, go to the menu bar at the top of IntelliJ, click on Navigate, and then click on Search Everywhere.

A window in the center of your screen will open. At the top of this new window click in the text field and enter “Download sources and Documentation”. This will download any sources that are needed for the project to run.

Persistence

Our team decided to use persistence to ensure that the data entered by users is not lost when the webpage is closed or when the server restarts. Persistence also ensures that the system can handle increased load efficiently. This would also track the progress of the form and who will be sent the next link to sign/fill.

Hibernate

Hibernate is an Object-Relational Mapping (ORM) framework that simplifies and streamlines the interaction between Java applications and relational databases.

This is a framework that takes care of generating the necessary SQL statements to perform CRUD (Create, Read, Update, Delete) operations, which saves development time and reduces the chance of errors. Hibernate also provides a database abstraction layer, meaning the Java code remains independent from the specific database vendor.

Maven

It is a powerful build automation and dependency management tool. Maven simplifies the management of project dependencies.

You can specify the required libraries such as Hibernate, database connectors, or web frameworks in the project's pom.xml file; which will cause Maven to automatically download and include them in your project.

Tomcat

Apache Tomcat is an open-source web server and servlet container which serves as a Java-based application server that allows the deployment and execution of Java Servlets, JavaServer Pages (JSP), and other Java-based web components. Tomcat's primary purpose is to host and run Java web applications, including dynamic webpages

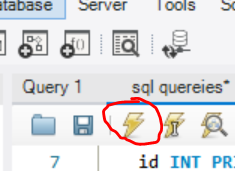
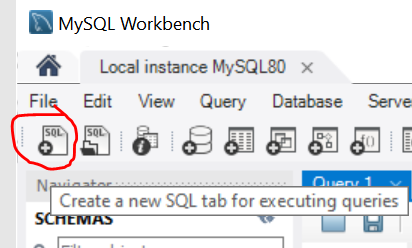
Tomcat allows the developer to deploy a web application easily. Once the dynamic webpage is developed in IntelliJ, it can package into a WAR (Web ARchive) file and deployed to Tomcat. Tomcat will handle the runtime execution of the application.

Database

Our team decided to use MySQL because it is an open-source relational database management system, which means it is freely available for use and it offers various security features, including user authentication, access control, and data encryption, helping you secure sensitive information in your database.

When installing MySQL make sure to write down and remember the username and password that was used as they will be used when setting up the persistence.xml file.

Populate Database

Once MySQL is installed and running, create the database and populate it with tables. To do this open MySQL and connect to your localhost using your username and password that you created during installation. Once logged in, click New Query Tab or CTRL + T

Copy and paste the following code into the new window that opens, save the query, and click the yellow lightning bolt to create the database and then populate the database with tables.

CREATE DATABASE edfDB;

USE edfDB;

CREATE TABLE student (

id INT PRIMARY KEY AUTO\_INCREMENT,

student\_name VARCHAR(255),

student\_vid VARCHAR(255),

student\_department VARCHAR(255),

student\_position VARCHAR(255),

student\_start\_date DATE,

student\_purpose VARCHAR(300),

student\_college\_benefit VARCHAR(300),

student\_sign VARCHAR(255),

student\_sign\_date DATE,

student\_email VARCHAR(255),

student\_form\_type ENUM('EMPLOYER', 'COLLEGE'),

student\_status ENUM('NOT-COMPLETED' , 'COMPLETED'),

student\_completed\_time DATETIME);

CREATE TABLE supervisor (

id INT PRIMARY KEY AUTO\_INCREMENT,

supervisor\_name VARCHAR(255),

supervisor\_explain VARCHAR(500),

supervisor\_sign VARCHAR(255),

supervisor\_sign\_date DATE,

supervisor\_email VARCHAR(255),

supervisor\_status ENUM('NOT-COMPLETED' , 'COMPLETED'),

supervisor\_completed\_time DATETIME,

student\_id INT,

FOREIGN KEY (student\_id) REFERENCES student(id));

CREATE TABLE odhr (

id INT PRIMARY KEY AUTO\_INCREMENT,

odhr\_funds TINYINT(1),

odhr\_eligibility TINYINT(1),

odhr\_comment VARCHAR(300),

odhr\_sign VARCHAR(255),

odhr\_name VARCHAR(255),

odhr\_sign\_date DATE,

odhr\_email VARCHAR(255),

odhr\_status ENUM('NOT-COMPLETED' , 'COMPLETED'),

odhr\_completed\_time DATETIME,

student\_id INT,

FOREIGN KEY (student\_id) REFERENCES student(id));

CREATE TABLE employer (

id INT PRIMARY KEY AUTO\_INCREMENT,

employer\_training\_title VARCHAR(255),

employer\_host\_org VARCHAR(255),

employer\_location VARCHAR(255),

employer\_locationstart\_date DATE,

employer\_locationend\_date DATE,

employer\_registration\_cost DECIMAL(10,2),

employer\_registration\_pay TINYINT(1),

employer\_email VARCHAR(255),

employer\_status ENUM('NOT-COMPLETED' , 'COMPLETED'),

employer\_completed\_time DATETIME,

student\_id INT,

FOREIGN KEY (student\_id) REFERENCES student(id));

CREATE TABLE college (

id INT PRIMARY KEY AUTO\_INCREMENT,

college\_course\_name VARCHAR(255),

college\_course\_number VARCHAR(255),

college\_credit\_hours DECIMAL(4,1),

college\_degree\_title VARCHAR(255),

college\_name VARCHAR(255),

college\_coursestart\_date DATE,

college\_courseend\_date DATE,

college\_partof\_degree TINYINT(1),

college\_degree\_type TINYINT(1),

college\_tuition\_type TINYINT(1),

college\_email VARCHAR(255),

college\_status ENUM('NOT-COMPLETED' , 'COMPLETED'),

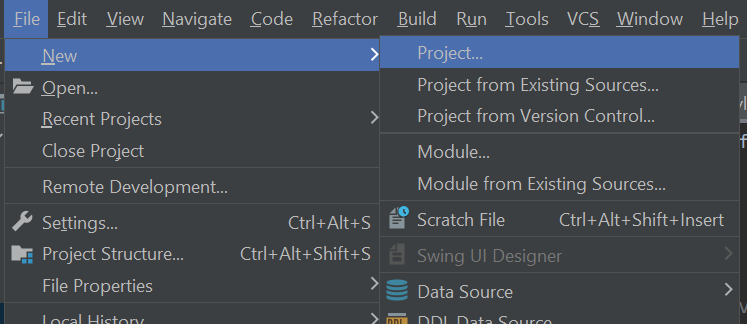
college\_completed\_time DATETIME,

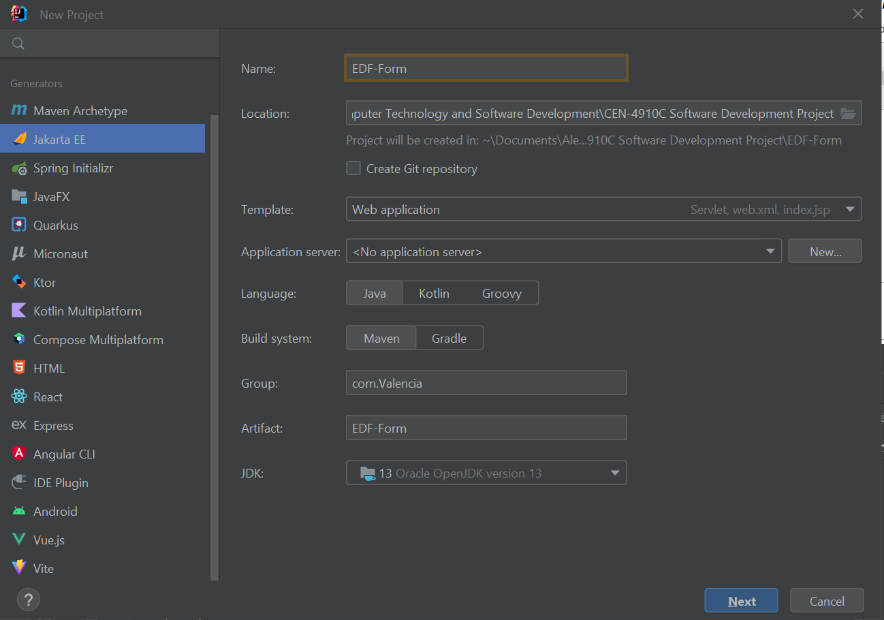
student\_id INT,

FOREIGN KEY (student\_id) REFERENCES student(id));

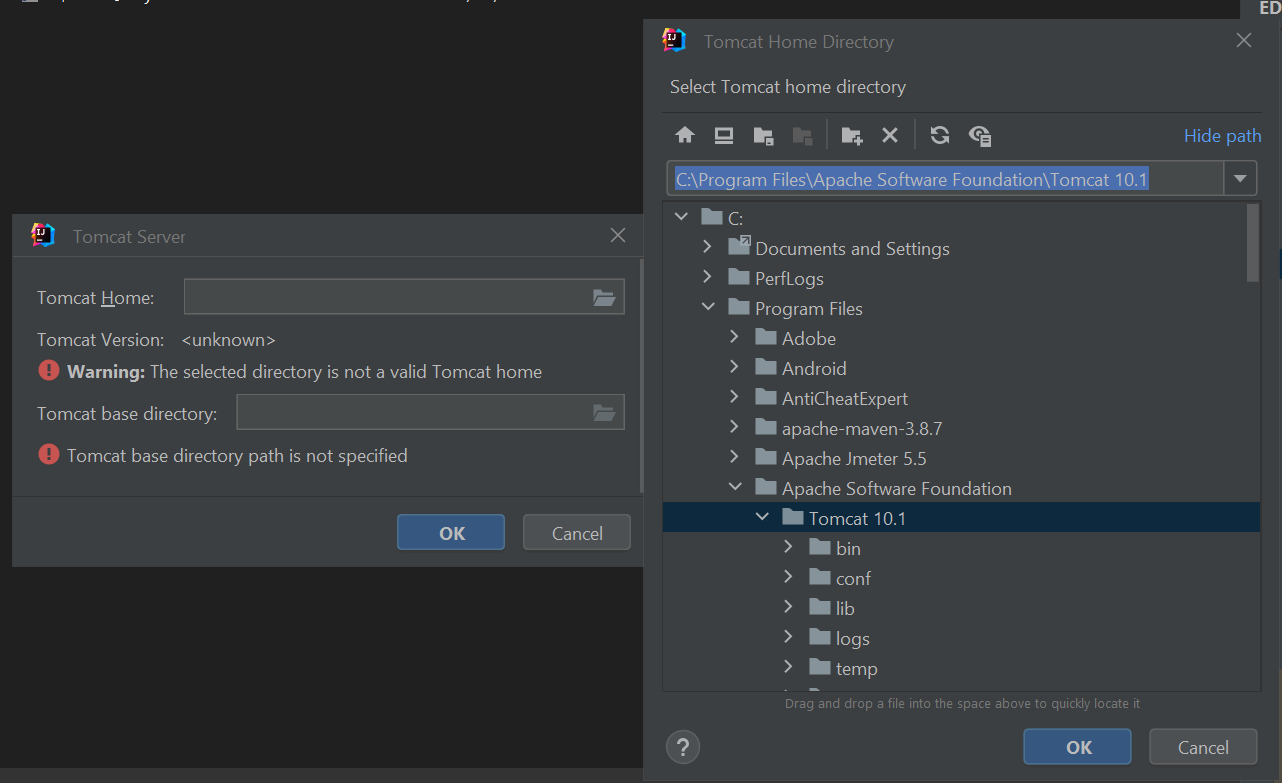
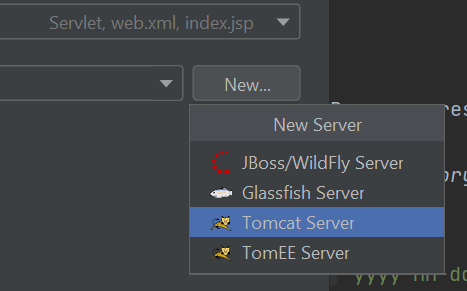
Dynamic Website

How to Configure IntelliJ?

To create the dynamic website first create a new project at the top left part of the menu bar. Click file, new, project... This will open a new window.

Select Java EE or Jakarta EE from the left side of the window. Enter a name for the project and select a location for it to be stored in, then change the template to Web Application. Also change the Application Server to tomcat by clicking New…

Once the new window opens click the folder on the right side of Tomcat Home and find the location where tomcat is installed. Click okay to close both windows.

Now that tomcat is set up and visible as the Application server, make sure Language is set to Java and Build System is set to Maven. Change the group name to com.Valencia and select a JDK that works for you, our team used version 13. Once done select next to continue to the next window and click okay to finish setting up the project.

Pom.xml

On the left side of the IntelliJ window, open the project folders until you find the pom.xml file. Make sure to use the same group Id and artifactId that was used during the project configuration. Next, add these dependencies below properties.

<groupId>com.valencia</groupId>  
<artifactId>EDF-Form</artifactId>  
<version>1.0-SNAPSHOT</version>  
<name>EDF-Form</name>  
<packaging>war</packaging>

<dependencies>  
 <dependency>  
 <groupId>jakarta.servlet</groupId>  
 <artifactId>jakarta.servlet-api</artifactId>  
 <version>5.0.0</version>  
 <scope>provided</scope>  
 </dependency>  
 <dependency>  
 <groupId>org.hibernate.orm</groupId>  
 <artifactId>hibernate-core</artifactId>  
 <version>6.0.2.Final</version>  
 </dependency>  
 <dependency>  
 <groupId>org.glassfish.jaxb</groupId>  
 <artifactId>jaxb-runtime</artifactId>  
 <version>3.0.2</version>  
 </dependency>  
 <dependency>  
 <groupId>org.junit.jupiter</groupId>  
 <artifactId>junit-jupiter-api</artifactId>  
 <version>${junit.version}</version>  
 <scope>test</scope>  
 </dependency>  
 <dependency>  
 <groupId>org.junit.jupiter</groupId>  
 <artifactId>junit-jupiter-engine</artifactId>  
 <version>${junit.version}</version>  
 <scope>test</scope>  
 </dependency>  
 <dependency>  
 <groupId>com.mysql</groupId>  
 <artifactId>mysql-connector-j</artifactId>  
 <version>8.0.32</version>  
 </dependency>  
 <dependency>  
 <groupId>jakarta.persistence</groupId>  
 <artifactId>jakarta.persistence-api</artifactId>  
 <version>3.1.0</version>  
 </dependency>  
</dependencies>

Also, do not forget to include the plugins within the build.

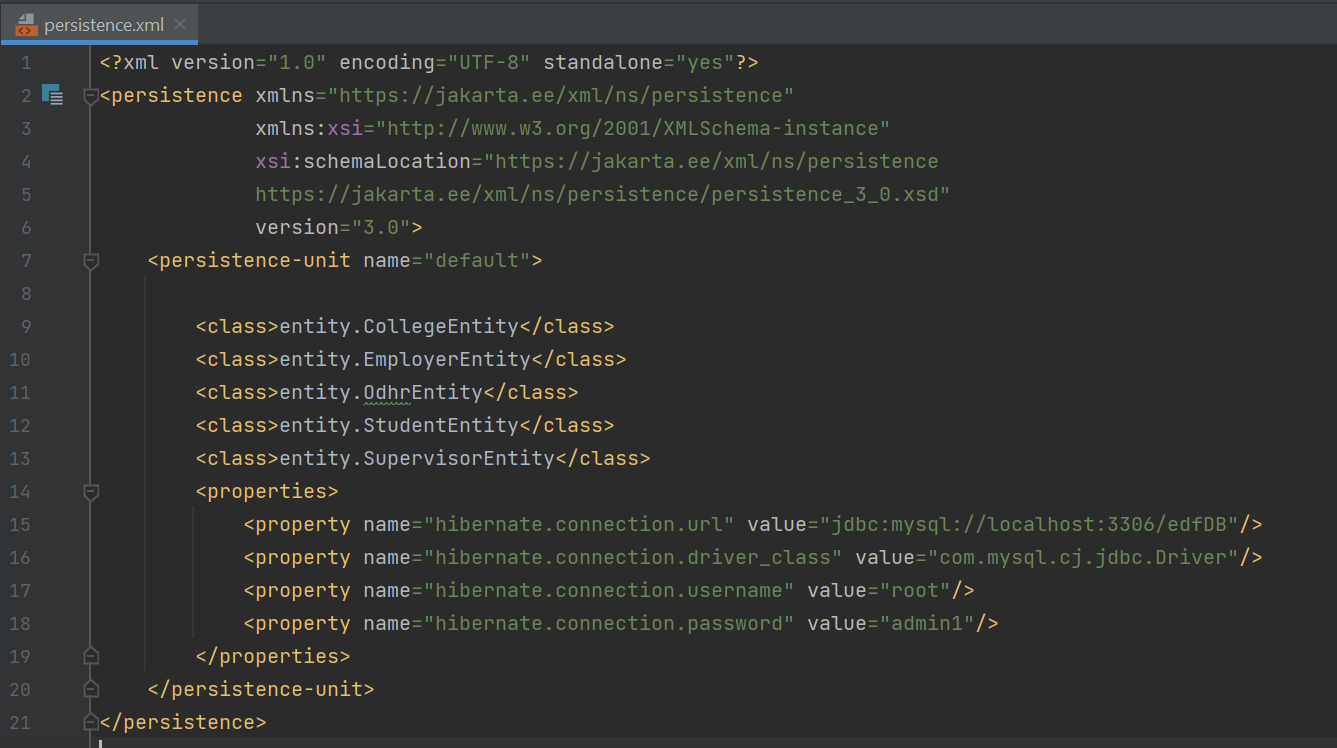
<plugins>  
 <plugin>  
 <groupId>org.apache.maven.plugins</groupId>  
 <artifactId>maven-war-plugin</artifactId>  
 <version>3.3.2</version>  
 </plugin>  
</plugins>

Create Persistence.xml

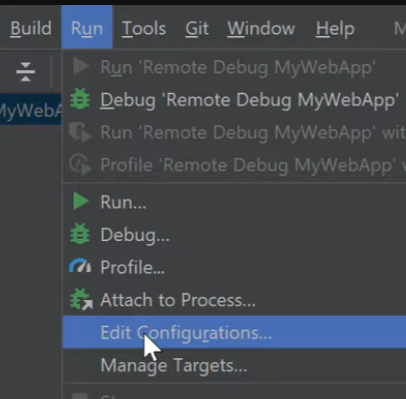
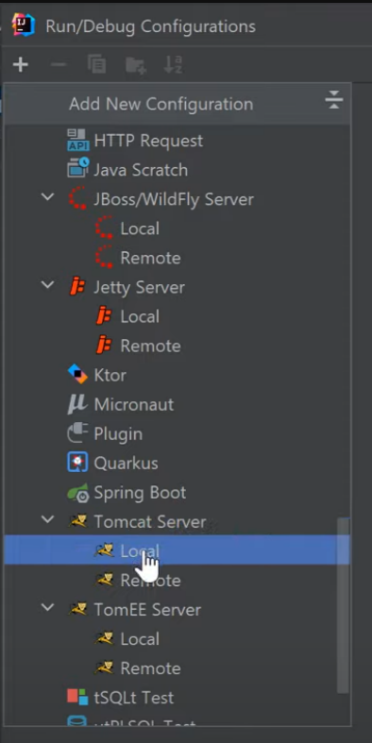
Inside the IntelliJ project, create a new directory named META-INF under the src/main/resources folder. Create a new XML file named persistence.xml inside the META-INF directory.

Configure persistence.xml

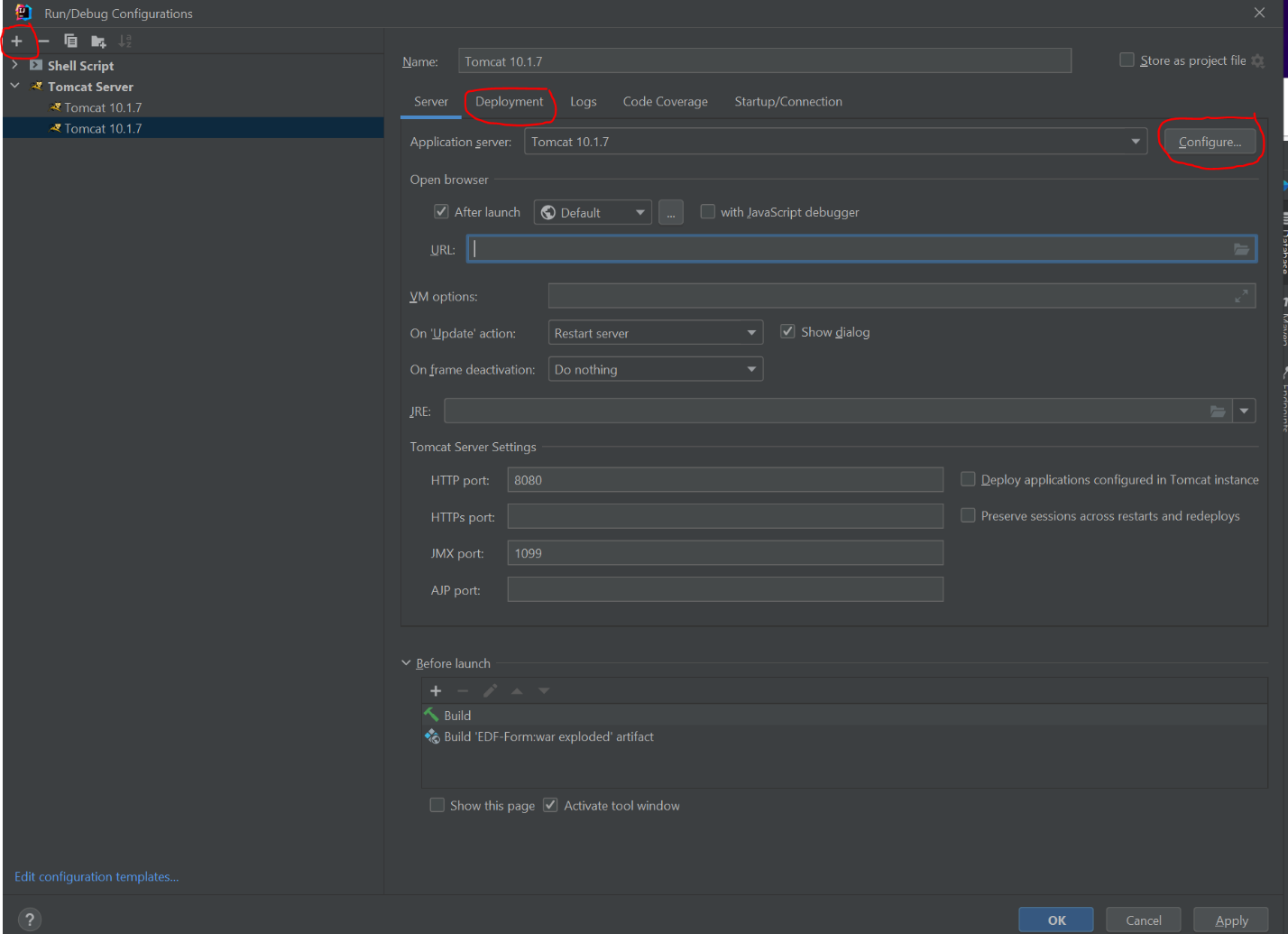
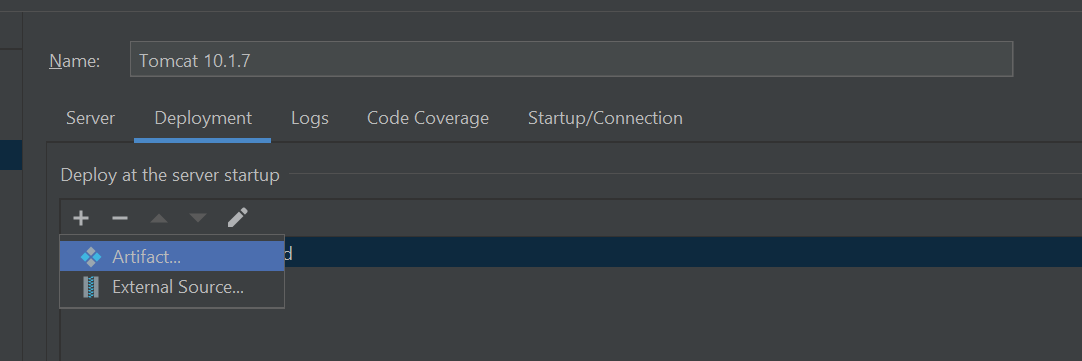
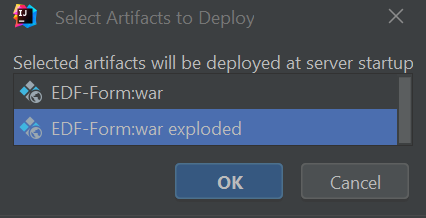
This file contains a list of JPA entity classes that represent the database tables, and the properties for the database connection. Make sure to define the persistence-unit name as default. The classes represent database tables. The property value for username and password are the same as the username and password that were set up with MySQL.



How to Configure Tomcat

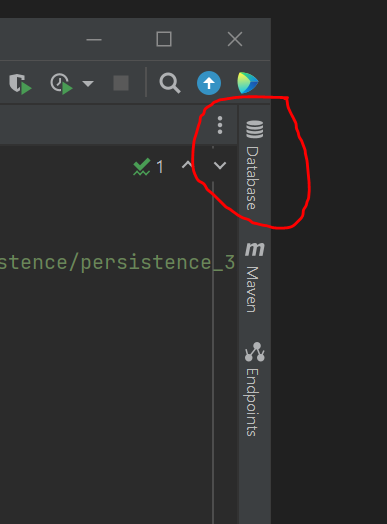
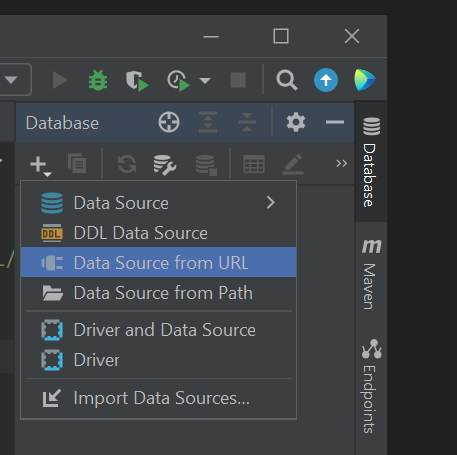
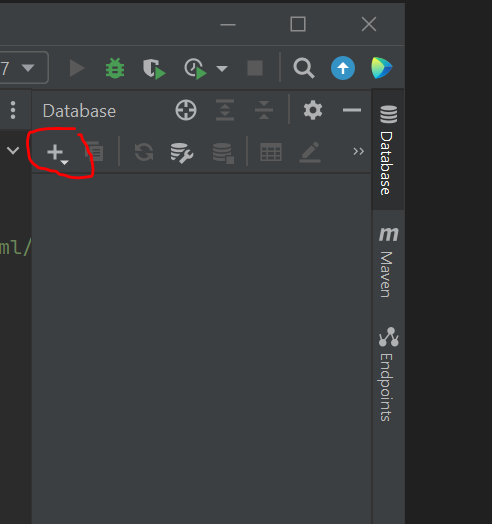
At the top middle of the menu bar click Run then Edit Configurations… To create a new tomcat configuration in the new window click the plus sign, scroll down to find, and select Tomcat Server Local.

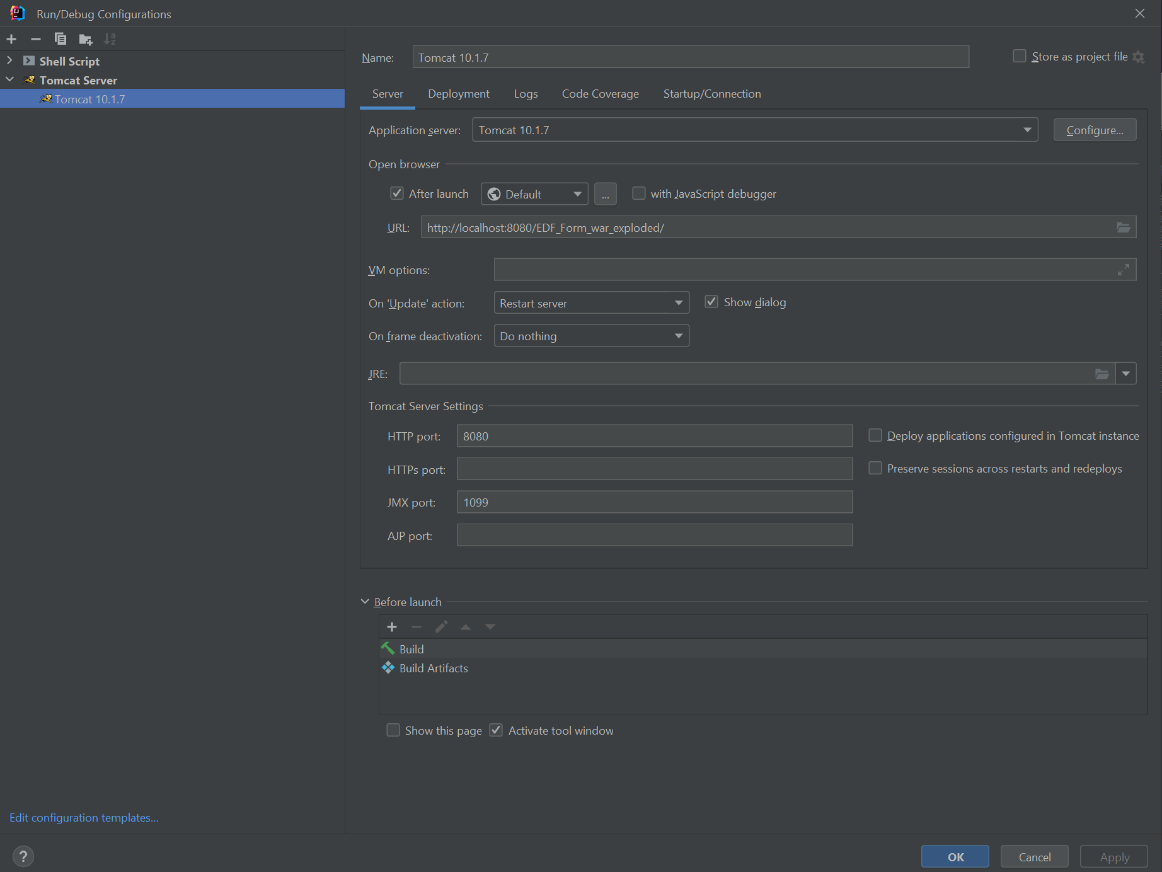
Give the configuration a name at the top of the window and then set the Application Server by selecting Configure. Just like before, click the folder on the right side of Tomcat Home and find the location where tomcat is installed then click okay.

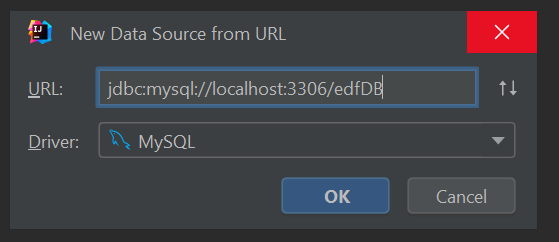
Now go to the Deployment Tab to add the application artifact that needs to get deployed to the server. Click the plus sign and click Artifact, in the new window select projectName:war exploded then click okay.

Click on Server to go back to the previous window to make sure the URL is using the database url with /projectName\_war\_exploded/ following it. Also make sure the HTTP port is set to 8080 and JMX port is 1099. Click Okay at the bottom to finish the configuration. Press Shift + F10 to start the server.

Connect to Database

Open IntelliJ , look at the top right side of the window under the X to close the program, for the tab labeled Database, and click it open a new window.



Now click the plus sign in that new window, then find and flick Data Source From URL. Enter the same database URL that was set up in the persistence.xml file.

Index.jsp

Inside the IntelliJ project, navigate to the src/main/webapp directory (create it if it does not exist). Create a new file named index.jsp inside the webapp directory. This file is the front end for our project which is displayed when the project runs.

What is a Servlet?

A Java servlet that handles the dynamic webpage's HTTP POST requests which allows users to enter data into a website. Navigate to the src/main/java/com.valencia.edfform (create it if it does not exist).

How to Configure a Servlet?

Create a new Servlet file ex. StudentServlet.java. The file needs to extend HttpServlet, making it a servlet that handles HTTP requests. Ensure that you import the necessary classes and packages for the Servlet, JPA (Hibernate), and date formatting.

Create a doPost method that tells the servlet to handle HTTP POST requests. Create an EntityManagerFactory (emf) and EntityManager (em) using JPA's Persistence class to interact with the database. Parse the date input received from the webpage into a java.sql.Date using SimpleDateFormat.

Create a new StudentEntity object and set its properties based on the form input received from the webpage. Begin a JPA transaction, save the StudentEntity object to the database, and commit the transaction. Then Close the EntityManager and EntityManagerFactory and redirect the user to a "success.html" page after successfully saving the data.

Entity

The entity classes are created using JPA annotations, which allow them to be mapped to the corresponding database tables. They represent the structure of the tables and provide a way for Java code to interact with the data stored in the database. The use of Hibernate and Tomcat allows for persistence of the data and enables communication with administrators for approval.

[](https://www.youtube.com/embed/NrfXyzO2rlU?feature=oembed)