Project Report

Epics

Employee management system

Team: Andromeda
Tanusree Debnath
ID: 013897364
Email: tanusree.debnath@sjsu.edu
Kevin Chang
Ngan Luu
San Jose State University

Summer 2021

CS157A

Goals and Description of the application

The Epics is an online web app which enables companies to save their employee information in a very organized manner. Employees can save their data. The database administrator and employees can view, save and delete their data from the database according to their permission level. Each employee has a different level of permission in the database. One employee cannot view every detail of another employee. Employees cannot delete data of another employee. One employee cannot change the data of another employee unless the employee is authorized to do so. Epics is passionate about its mission of transforming employee management services into online services through innovation.

Application Requirements and Architecture

Presentation Tier

In Epics, the user interface will be a web application that handles the interaction with the user and displays information. I will build this web application with the programming language Java.

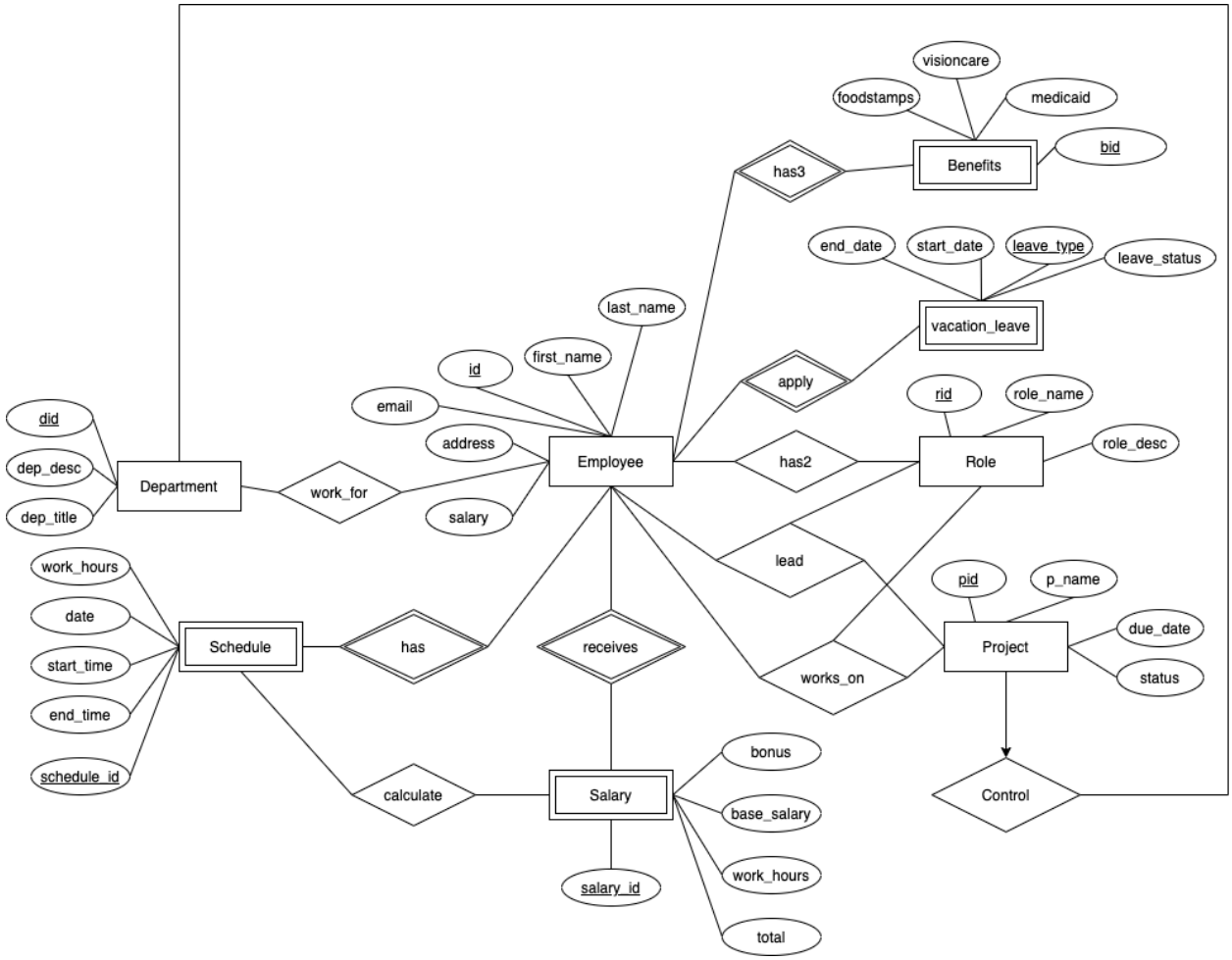
Logic Tier

Scope will use Apache Tomcat9 as a backend server for handling user requests and communicating with the database.

Data Tier

Our project will use ORACLE as our RDBMS(Relational Database Manage System) to manage and provide access to the data.

ER data model with the requirement of minimum 10 relations for the application



A database design, including tables and relationships among tables

```
CREATE TABLE EMPLOYEE
(
        ID NUMBER not null,
        FIRST_NAME VARCHAR2(50) not null,
        LAST_NAME VARCHAR2(50) not null,
        EMAIL VARCHAR(50) not null,
        ADDRESS VARCHAR(50) not null,
        DEPARTMENT VARCHAR2(50),
        SALARY NUMBER
)

CREATE TABLE HAS_SCHEDULE
(
        SCHEDULE_ID NUMBER NOT NULL,
```

```
    WORK_HOURS NUMBER NOT NULL,

)

insert into EMPLOYEE values (1, 'Rina' , 'sales', 1254);
insert into EMPLOYEE values (2, 'Tina' , 'teach', 1256);
insert into EMPLOYEE values (3, 'Sina' , 'sales', 1255);
```

Major design decisions

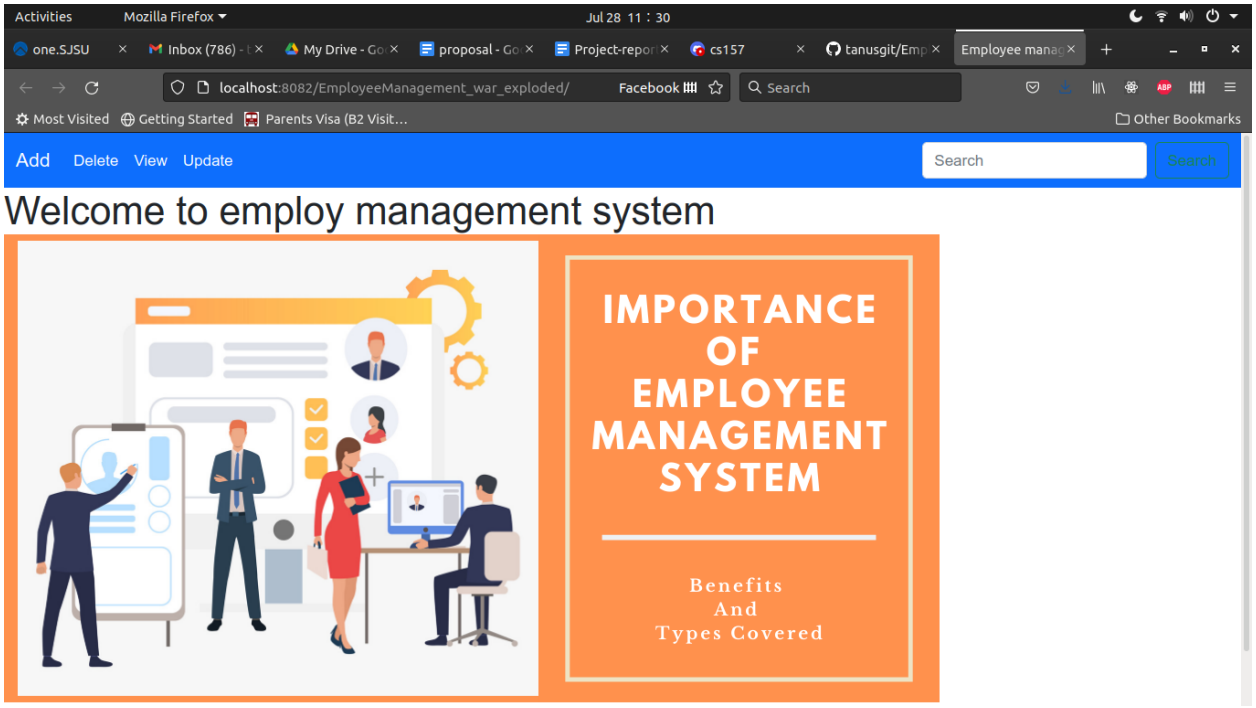
As we did not get so much time to implement the webapp we had to leave few features unimplemented. We decided to implement the 3 main features of the webapp which are add functionality, delete and view functionality. We chose tomcat as our server and we implemented the project using Java programming language as we all are familiar with Java.

Implementation details

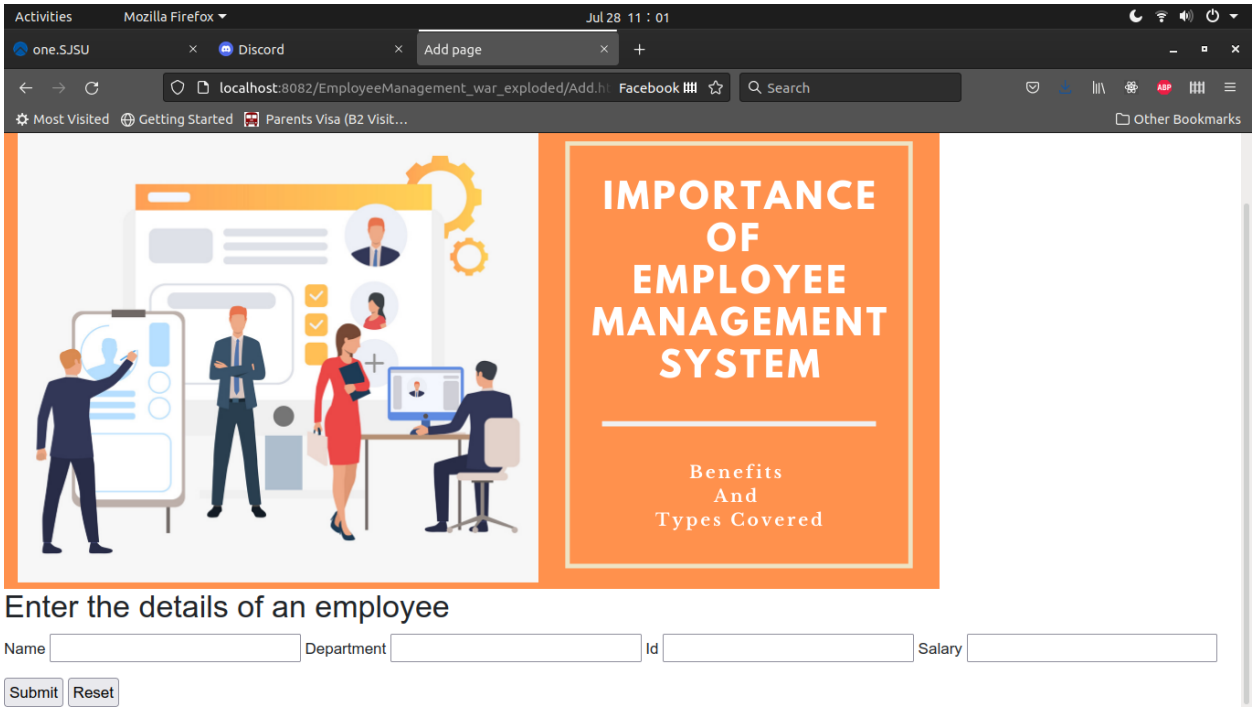
We have implemented the webapp using Java programming language.
We have used servlet to add the employee to the database.
We have used Oracle as our database.
We have used an index.jsp file to show the home page.
We have used an Add.html file to render the add page on the screen.
We have used GUI features to add, show and delete employees.

Demonstration of example system run

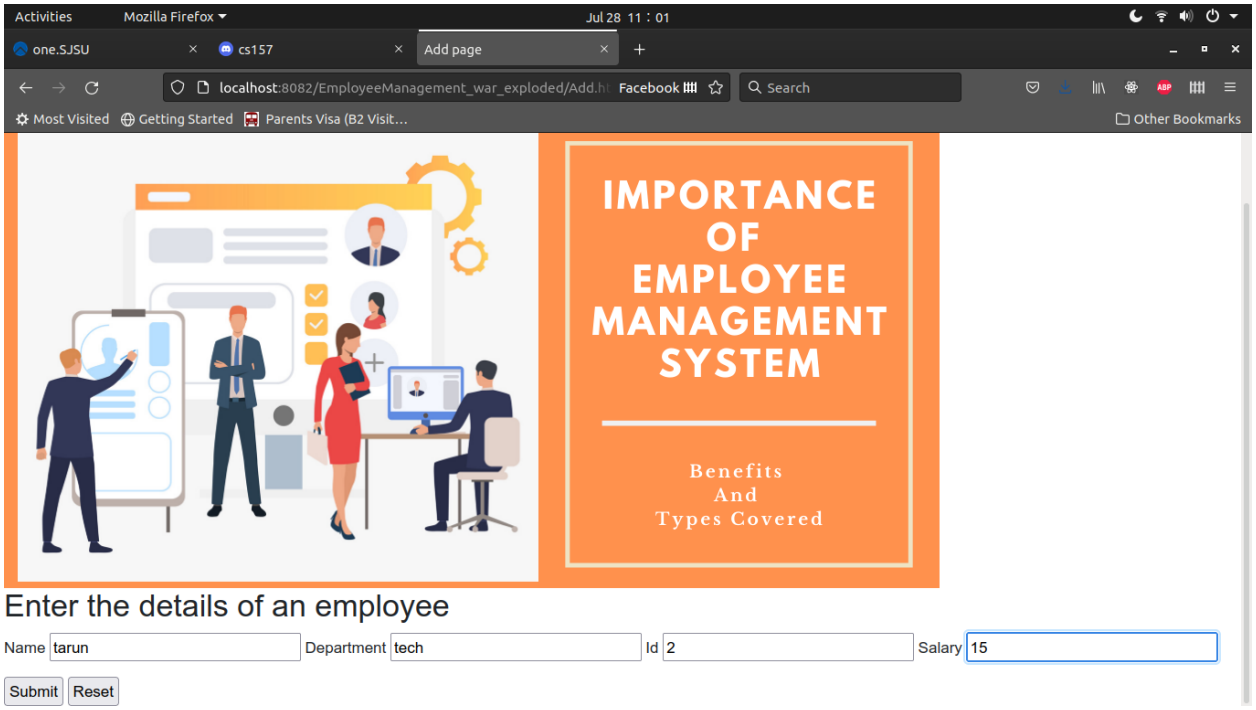
First clone to repo to your local machine.
Click on edit configuration and select tomcat 9 as your server.
You need to add the ojdbc14 jar file to the project path.
Then click on the run button.
Make sure you are running tomcat.
Then the homepage of the webapp would be displayed on the screen.



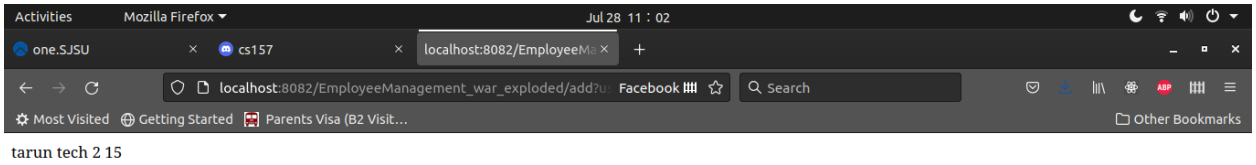
If you click on the add button you would see this page.



Now fill the form.



After clicking on the submit button you would get this view.



Conclusions

We have learned a lot about web applications and how to connect our applications to the database by doing this project. We did not have much time to implement everything in this short amount of time as we are taking this

curse in the summer session. So, we left a few features out from our application. Our web application is running fine and it is able to add, delete and view employees to the database.

Lessons learned

We have learned that it is very important to add the jar file inorder to connect to the database. We were struggling in the beginning to connect to the database as we did not know how to add the jar files. We have learned about servlets and many useful functions in the java servlet.

Possible improvements

We could use a better graphical user interface for our project. We could show many functionalities in our project. We could work in person for better communication.