

Anchored - Documentation

Anchors is a web application built from scratch using HTML, CSS, JavaScript, Servlet, JDBC connections, and MySQL database. The application provides a user-friendly platform for sailors and boat owners to enter and store information about their vessels and crew, including personal details and vessel specifications. Users can also bind boats to sailors, remove sailors or boats from the database, and search for specific sailors or boats using keywords.

Anchors is a versatile tool that is perfect for anyone who needs to manage their boat or crew data. The application is built using the latest technology and is optimized for performance, making it fast and responsive. Whether you need to manage a small personal vessel or an entire fleet, Anchors can help you streamline your data management needs.

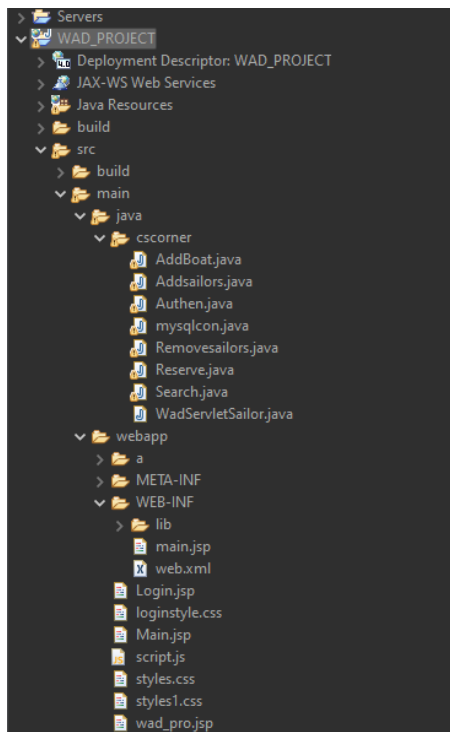
Contributors:

- 122010301016 - Sandeep Dokala
- 122010332047 - Bogendhra Reddy
- 122010330046 - Dhanush Yadav
- 122010317036 - Jayanth Sunkari

Dependencies:

- Eclipse for Web development
- Tomcat server v9
- Database with connector driver
- Browser

Project-Structure:



Project structure is easily manageable using IDEs like Eclipse. Here the WEB-INF folder is important as it contains all our files that are executed on the server. The lib folder in the WEB-INF folder contains all the external libraries like SQL connector in it. And the webapp folder contains our frontend files which the user can view the website and work with it.

Frontend:

Main.jsp: The main file which contains UI for all the functionalities of the web application

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>

<!DOCTYPE html>
<html>
    <head>
        <title>Anchored - Streamlining Sailor and Boat Data Entry</title>
        <link rel="stylesheet" href="styles.css">
    </head>
    <body>
        <div class="header">
            <div class="pro-title">
                <h2>Anchored</h2>
            </div>
            <div class="middle-lane">

            </div>
            <div class="logout-section">
                <button onclick="logout()">Logout</button>
            </div>
        </div>
        <div class="container">
            <div class="content">
                <form class="sailor-section" action="url-addsailors" method="post">
                    <h2 class="add-remove-title">Add/Remove a Sailor: </h2>
                    <div class="sailor-settings">
                        <select onchange="sailorOptCheck()" name="add-remove" id="sailor-add-remove" class="add-remove">
                            <option value="add">Add</option>
                            <option value="remove">Remove</option>
                        </select>
                        <div class="add-section" id="sailor-add">
                            <input type="text" class="add-inp" placeholder="Enter name to add:" name="name" id="sailor-name">
                            <input type="text" class="add-inp" placeholder="Enter Age:" name="age" id="sailor-age">
                            <input type="text" class="add-inp" placeholder="Enter Rating:" name="rating" id="sailor-rating">
                        </div>
                        <div class="remove-section" id="sailor-remove">
                            <input type="text" class="remove-inp" placeholder="Enter id to remove:" name="id" id="sailor-id">
                        </div>
                        <input type="submit" value="Submit" class="submit-btn">
                    </div>
                    <!-- <p id="sailor-query-info" style="color: red; text-align: center; margin: 4px auto;"></p> -->
                </form>
                <hr>
                <form class="boat-section" action="url-addboat" method="post">
                    <h2 class="add-remove-title">Add/Remove a Boat: </h2>
                    <div class="boat-settings">
                        <select onchange="boatOptCheck()" name="add-remove" id="boat-add-remove" class="add-remove">
                            <option value="add">Add</option>
                            <option value="remove">Remove</option>
                        </select>
                        <div class="add-section" id="boat-add">
                            <input type="text" class="add-inp" placeholder="Enter name to add:" name="name" id="boat-name">
                            <input type="text" class="add-inp" placeholder="Enter color:" name="color" id="boat-color">
                        </div>
                        <div class="remove-section" id="boat-remove">
                            <input type="text" class="remove-inp" placeholder="Enter id to remove:" name="id" id="boat-id">
                        </div>
                        <input type="submit" value="Submit" class="submit-btn">
                    </div>
                </form>
                <hr>
                <form class="reserve-section" action="url-reserve" method="post">
                    <h2 class="add-remove-title">Reserve a Boat: </h2>
                    <div class="reserve-settings">
                        <div id="reserve-section">
                            <input type="text" class="add-inp" placeholder="Enter sailor id:" name="sid" id="sailor-id-res">
                            <input type="text" class="add-inp" placeholder="Enter boat id:" name="bid" id="boat-id-res">
                        </div>
                        <!-- <button class="submit-btn" onclick="valReserve()">Submit</button> -->
                        <input type="submit" value="Submit" class="submit-btn">
                    </div>
                </form>
            </div>
        </div>
    </body>
</html>
```

```

        </div>
    </form>
    <hr>
    <form class="view-section" action="url-search" method="post">
        <h2 class="add-remove-title">View details of a Sailor/Boat/Reserves by: </h2>
        <div class="view-settings">
            <input type="text" placeholder="Enter keywords to search the db: " class="add-inp search-bar" name="searchBar">
            <!-- <button class="submit-btn" onclick="valReserve()">Submit</button> -->
            <input type="submit" value="Submit" class="submit-btn">
        </div>
    </form>
</div>
</div>
<script src="script.js"></script>
</body>
</html>

```

Login.jsp:

```

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
    <head>
        <title>Anchored - Streamlining Sailor and Boat Data Entry</title>
        <link rel="stylesheet" href="styles.css">
    </head>
    <body>
        <div class="header">
            <div class="pro-title">
                <h2>Login to Anchored</h2>
            </div>
            <div class="middle-lane">

            </div>
            <div class="logout-section">

            </div>
        </div>
        <div class="container">
            <form class="login-section" action="url-authen">
                <div class="login-content">
                    <h2 class="add-remove-title">Username</h2>
                    <input type="text" placeholder="Enter your username: " class="add-inp login-inp-field" name="usern">
                    <h2 class="add-remove-title">Password</h2>
                    <input type="password" placeholder="Enter your password: " class="add-inp login-inp-field" name="pwd">
                    <!-- <br> -->
                </div>
                <div class="submit-field">
                    <input type="submit" value="Login" class="login-btn-form">
                </div>
            </form>
        </div>
    </body>
</html>

```

styles.css:

```

* {
    margin: 0;
    padding: 0;
}

body {
    background-color: #dddddd;
    display: flex;
    flex-direction: column;
}

.header {
    background-color: #fff;
    display: flex;
    justify-content: space-between;
    box-shadow: 0px 0px 300px -10px rgba(0,0,0,0.75);
}

```

```

    width: 90%;
    border-radius: 20px;
    margin: 0.8em auto;
}

.middle-lane {
    width: 60%;
    background-color: #fff;
    border-radius: 20px;
    height: 64px;
    align-self: center;
    /* box-shadow: 0px 0px 26px 0.5px rgba(0,0,0,0.75); */
}

.pro-title {
    font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;
    font-size: 20px;
    padding: 1em;
    width: fit-content;
}

.logout-section {
    width: 8%;
    display: flex;
    justify-content: center;
    align-self: center;
}

.logout-section button {
    padding: 1em;
    border-radius: 14px;
    border: none;
    font-size: 16px;
}

.logout-section button:hover {
    cursor: pointer;
}

.container {
    min-height: 88vh;
    height: fit-content;
    width: 96%;
    background-color: #fff;
    margin: auto;
    border-radius: 20px;
    display: flex;
    justify-content: center;
}

.content {
    width: 70%;
    margin: auto;
    align-self: center;
    border: 1px solid #dddddd;
    border-radius: 10px;
    height: fit-content;
}

.sailor-section, .boat-section, .reserve-section, .view-section {
    margin: 44px 0;
}

.sailor-settings, .boat-settings, .reserve-settings, .view-settings {
    display: flex;
    width: 80%;
    margin: auto;
    justify-content: center;
}

.remove-section {
    display: none;
}

.add-remove-title {
    margin: 20px 40px;
    font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;
}

.add-remove {

```

```

        align-self: center;
        height: 30px;
    }

    .add-inp, .remove-inp {
        font-size: 18px;
        padding: 4px;
        margin: 4px 8px;
    }

    .submit-btn {
        padding: 8px 12px;
        align-self: center;
    }

    .search-bar {
        width: 60%;
    }

    input[type="text"] {
        outline: none;
    }

    .login-section {
        border: black solid 1px ;
        height: 50vh;
        width: 40%;
        align-self: center;
        border-radius: 30px;
        display: flex;
        flex-direction: column;
        justify-content: center;
    }

    .login-inp-field {
        width: 60%;
        /* margin-left: 10%; */
        margin: auto;
    }

    .login-content {
        width: 70%;
        align-self: center;
        display: flex;
        flex-direction: column;
        /* border: black solid 1px; */
        padding-bottom: 20px;
    }

    .login-btn-form {
        width: 6em;
        height: 40px;
        margin-left: auto;
        align-self: flex-end;
    }

    .submit-field {
        width: 71.8%;
        display: flex;
    }

    table, th, td {
        border: 1px solid black;
    }

    th{
        background-color: #fff;
    }

```

script.js:

```

function sailorOptCheck() {
    var selectedOpt = document.getElementById('sailor-add-remove').value;
    if (selectedOpt == 'add') {
        document.getElementById('sailor-remove').style.display = 'none';
        document.getElementById('sailor-add').style.display = 'block';
    }
}

```

```

    } else if (selectedOpt == 'remove') {
        document.getElementById('sailor-remove').style.display = 'block';
        document.getElementById('sailor-add').style.display = 'none';
    }
}

function boatOptCheck() {
    var selectedOpt = document.getElementById('boat-add-remove').value;
    if (selectedOpt == 'add') {
        document.getElementById('boat-remove').style.display = 'none';
        document.getElementById('boat-add').style.display = 'block';
    } else if (selectedOpt == 'remove') {
        document.getElementById('boat-remove').style.display = 'block';
        document.getElementById('boat-add').style.display = 'none';
    }
}

function valSailor() {
    var selectedOpt = document.getElementById('sailor-add-remove').value;
    if (selectedOpt == 'add') {
        var sailorName = document.getElementById('sailor-name').value;
        var sailorAge = document.getElementById('sailor-age').value;
        var sailorRating = document.getElementById('sailor-rating').value;

        var query = `INSERT INTO SAILORS VALUES(${String(Math.round(Math.random() * 100))}, \`${sailorName}\`, ${sailorAge}, ${sailorRating})`;
        console.log(query);
    } else if (selectedOpt == 'remove') {
        var sailorId = document.getElementById('sailor-id').value;

        var query = `DELETE FROM SAILORS WHERE SID=${sailorId}`;
        console.log(query);
    }
}

function valBoat() {
    var selectedOpt = document.getElementById('boat-add-remove').value;
    if (selectedOpt == 'add') {
        var boatName = document.getElementById('boat-name').value;
        var boatColor = document.getElementById('boat-color').value;

        var query = `INSERT INTO BOATS VALUES(${String(Math.round(Math.random() * 100))}, \`${boatName}\`, \`${boatColor}\`);`;
        console.log(query);
    } else if (selectedOpt == 'remove') {
        var boatId = document.getElementById('boat-id').value;

        var query = `DELETE FROM BOATS WHERE BID=${boatId}`;
        console.log(query);
    }
}

function valReserve() {
    var sailorId = document.getElementById('sailor-id-res').value;
    var boatId = document.getElementById('boat-id-res').value;

    var query = `INSERT INTO RESERVES VALUES(${sailorId}, ${boatId}, \`${getToday()}\`);`;
    console.log(query);
}

function getToday() {
    const date = new Date();
    const formattedDate = date.toLocaleDateString('en-GB', {day: 'numeric', month: 'short', year: 'numeric'}).replace(/ /g, '/');
    return formattedDate;
}

function logout(){
    return(window.location.href = "Login.jsp");
}

function searchCheck(){
    var selectedOpt = document.getElementById('res-search-check').value;
    if (selectedOpt == 'sailors') {
        document.getElementById('boat-id-view').style.display = 'none';
        document.getElementById('res-id-view').style.display = 'none';
        document.getElementById('sailor-id-view').style.display = 'block';
    } else if (selectedOpt == 'boats') {
        document.getElementById('boat-id-view').style.display = 'block';
        document.getElementById('res-id-view').style.display = 'none';
        document.getElementById('sailor-id-view').style.display = 'none';
    } else if (selectedOpt == 'reserves') {
        document.getElementById('boat-id-view').style.display = 'none';
    }
}

```

```

document.getElementById('res-id-view').style.display = 'block';
document.getElementById('sailor-id-view').style.display = 'none';
}
}

```

These above three files take place in the frontend UI to give the user the best experience using the site at their comfort. The Javascript code basically defines how the website should function when the user clicks on something, or does something, or types something on the website. The CSS code defines how the website should look when the user navigates to the website or another page. The JSP file is basically a file that contains HTML code for the website that defines the structure of the website of what the website contains.

When the user enters something and clicks on the submit button of a form, the HTML forms call the servlet program, which is written in java, that is mapped to the form. These servlet forms are basically called server site scripting. Instead of doing things where user can change, working with the data where user cannot send malicious code to the server, we use server side scripting. The mapping of the forms to a certain java programs, in this case it is servlets, are done in a file named “web.xml” in the WEB-INF folder, which is present in the project structure.

web.xml:

```

<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://xmlns.jcp.org/xml/ns/javaee" xsi:schemaLocation="http://xmlns.

    <servlet>
        <servlet-name>addsailor</servlet-name>
        <servlet-class>cscorner.Addsailors</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>addsailor</servlet-name>
        <url-pattern>/url-addsailors</url-pattern>
    </servlet-mapping>

    <servlet>
        <servlet-name>addboat</servlet-name>
        <servlet-class>cscorner.AddBoat</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>addboat</servlet-name>
        <url-pattern>/url-addboat</url-pattern>
    </servlet-mapping>

    <servlet>
        <servlet-name>reserve</servlet-name>
        <servlet-class>cscorner.Reserve</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>reserve</servlet-name>
        <url-pattern>/url-reserve</url-pattern>
    </servlet-mapping>

    <servlet>
        <servlet-name>authen</servlet-name>
        <servlet-class>cscorner.Authen</servlet-class>
    </servlet>
    <servlet-mapping>

```

```

        <servlet-name>authen</servlet-name>
        <url-pattern>/url-authen</url-pattern>
    </servlet-mapping>

    <servlet>
        <servlet-name>search</servlet-name>
        <servlet-class>cscorner.Search</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>search</servlet-name>
        <url-pattern>/url-search</url-pattern>
    </servlet-mapping>

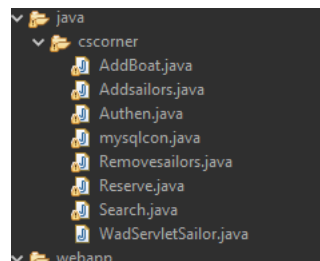
</web-app>

```

Backend:

The data that is sent from the front end is recieved at the backend to their respective files where the data is sent to.

The backend contains the servlet programs that integrates the JDBC connection with the database.



- Addsailors.java - Contains code for servlet and JDBC connections when the user inputs into sailors.

```

package cscorner;

import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public class Addsailors extends HttpServlet {

    public void service(HttpServletRequest req ,HttpServletResponse res)throws IOException{
        String name =req.getParameter("name");
        String age=req.getParameter("age");
        String rating=req.getParameter("rating");
        String id=req.getParameter("id");
        PrintWriter out=res.getWriter();

        out.println("<h1 class= \"'+\"'+\"header\"+'\"'+\">Output of Sailors Table</h1>");
        out.println("<link rel=\"stylesheet\" href=\"styles1.css\">");


        try {
            try {
                Class.forName ("com.mysql.jdbc.Driver");
            } catch (ClassNotFoundException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }

```



```

    }
    String stmt1;
    String stmt2="select * from sailors";
    Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/WAD_DB","user","admin");
    out.println("<p>Connection establishment successful.....</p>");
    Statement st=con.createStatement();
    if (id=="") {
        String stmt ="select max(sid) as maxi from sailors";
        ResultSet rs=st.executeQuery(stmt);
        int sid = 0;
        while (rs.next()) {
            sid =rs.getInt("maxi");
        }
        sid++;

        stmt1="insert into sailors values("+sid+", "+"'+ name +'"+", "+ age +", "+ rating +)";
        out.println("<p>add option selected</p>");
    }else {
        stmt1="delete from sailors where sid="+id;
        out.println("<p>remove option selected</p>");
    }

    st.execute(stmt1);
    ResultSet rs1=st.executeQuery(stmt2);
    out.println("<table style=\"width:100%\"><tr><th>SID</th><th>NAME</th><th>AGE</th><th>RATING</th></tr>");
    while(rs1.next()) {
        out.println("<tr><td>"+rs1.getObject(1)+"</td><td>"+rs1.getObject(2)+"</td><td>"+rs1.getObject(3)+"</td><td>"+rs1.getObject(4)+"</td></tr>");
    }
    out.println("</table>");
    con.close();
    out.println("<p>query executed.....</p>");
} catch (SQLException e) {
    out.println("error.....");
    e.printStackTrace();
}

}

private int length(String name) {
    // TODO Auto-generated method stub
    return 0;
}

}

```

The above [AddSailors.java](#) code takes the details entered by the user, which are SailorName, Age, Rating, and generates a SQL query with values given by the user and sends to the database.

- [AddBoat.java](#) - Contains code for servlet and JDBC connections when the user inputs into boats.

```

package cscorner;
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public class AddBoat extends HttpServlet{

```

```

public void service(HttpServletRequest req ,HttpServletResponse res) throws IOException {
    String name =req.getParameter("name");
    String color =req.getParameter("color");
    String id=req.getParameter("id");

    PrintWriter out=res.getWriter();

    out.println("<h1 class= '"+""+"header"+"'">Output of Boat Table</h1>");
    out.println("<link rel=\"stylesheet\" href=\"stylesheet1.css\">");

    try {
        try {
            Class.forName ("com.mysql.jdbc.Driver");
        } catch (ClassNotFoundException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/WAD_DB","user","admin");
        Statement st=con.createStatement();

        String stmt2="select * from boats;;;
        if (id=="") {
            String stmt ="select max(bid) as maxi from boats";
            ResultSet rs=st.executeQuery(stmt);
            int bid = 0;
            while (rs.next()) {
                bid =rs.getInt("maxi");
            }

            bid++;
            out.println("<p>add option selected.....,</p>");

            String stmt1="insert into boats values("+bid+", '"+ name +"' , '"+color+"' )";

            st.execute(stmt1);
        }else {

            out.println("remove option selected.....");
            String stmt1="delete from boats where bid =" +id;

            st.execute(stmt1);
        }
        ResultSet rs1=st.executeQuery(stmt2);
        out.println("<table style=\"width:100%\">\r\n"
            + "<tr>\r\n"
            + "    <th>BID</th>\r\n"
            + "    <th>BOAT_NAME</th>\r\n"
            + "    <th>BOAT_COLOR</th>\r\n"
            + "</tr>\r\n");
        while(rs1.next()) {
            out.println("<tr>\r\n"
                + "    <td>" +rs1.getObject(1)+"</td>\r\n"
                + "    <td>" +rs1.getObject(2)+"</td>\r\n"
                + "    <td>" +rs1.getObject(3)+"</td>\r\n"
                + "</tr>\r\n");
        }
        out.println("</table>");
        con.close();
        out.println("<p>query executed.....</p>");
    }catch(SQLException e) {
        out.println("<p>error.....,</p>");
        e.printStackTrace();
    }
}

```

The above [AddBoat.java](#) code takes the details entered by the user, which are BoatName, Color, and generates a SQL query with values given by the user and sends to the database.

The above [AddSailors.java](#) code and [AddBoat.java](#) code also reads the max id from the database and generates a new id for the current new values that are going to be entered in the database.

- [Reserve.java](#) - Contains code for sevlet and JDBC connections. This program takes two inputs SID and BID and binds them together in the database.

```
package cscorner;
import java.time.format.DateTimeFormatter;
import java.time.LocalDate;

import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public class Reserve extends HttpServlet{

    public void service(HttpServletRequest req ,HttpServletResponse res) throws IOException {
        int sid=Integer.parseInt(req.getParameter("sid"));
        int bid=Integer.parseInt(req.getParameter("bid"));

        PrintWriter out=res.getWriter();

        out.println("<h1 class= \"<\">header\">\">Output of Reserves Table</h1>");
        out.println("<link rel=\"stylesheet\" href=\"styles1.css\">");

        try {
            try {
                Class.forName ("com.mysql.jdbc.Driver");
            } catch (ClassNotFoundException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }

            System.out.println("bid"+bid +"sid"+sid);
            Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/WAD_DB","user","admin");
            Statement st=con.createStatement();

            String stmt2="select * from reserves;";

            String stmt1="insert into reserves values("+sid+", "+bid+", "+"'+java.time.LocalDate.now()+"'"+")";

            st.execute(stmt1);

            ResultSet rs1=st.executeQuery(stmt2);
            out.println("<table style=\"width:100%\">\r\n"
                + "<tr>\r\n"
                + "    <th>SID</th>\r\n"
                + "    <th>BID</th>\r\n"
                + "    <th>DAY</th>\r\n"
                + "</tr>\r\n");
            while(rs1.next()) {
                out.println("<tr>\r\n"
                    + "    <td>"+rs1.getObject(1)+"</td>\r\n"
                    + "    <td>"+rs1.getObject(2)+"</td>\r\n"
                    + "    <td>"+rs1.getObject(3)+"</td>\r\n"
                    + "</tr>\r\n");
            }
            out.println("</table>");
            con.close();
            out.println("<p>query executed.....,</p>");
        }catch(SQLException e) {
            out.println("<p>error.....,</p>");
        }
    }
}
```

```

        e.printStackTrace();
    }
}

}

```

The same goes for the removing data from the database.

- Removesailors.java

```

package cscorner;

import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public class Removesailors extends HttpServlet{

    public void service(HttpServletRequest req ,HttpServletResponse res) throws IOException {
        int id=Integer.parseInt(req.getParameter("id"));

        PrintWriter out=res.getWriter();

        out.println("id is "+ id );

        try {
            try {
                Class.forName ("com.mysql.jdbc.Driver");
            } catch (ClassNotFoundException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }

            System.out.println("sid of sailor"+id);
            Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/WAD_DB","user","admin");
            Statement st=con.createStatement();
            String stmt ="select sname from sailors where sid="+id;
            ResultSet rs=st.executeQuery(stmt);
            while (rs.next()) {
                out.println(rs.getInt("sname"));
            }
            String stmt1="delete from sailors where sid="+id;

            st.execute(stmt1);

            con.close();
            out.println("query executed.....");
        }catch(SQLException e) {
            out.println("error.....");
            e.printStackTrace();
        }
    }
}

```

- [Search.java](#) - This is the code that allows user to view or search the database for a certain Sailor or Boat, by inputting their names, age, date, anything that is available into the search field separating by commas (",").

```

package cscorner;

import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;

import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public class Search extends HttpServlet {

    public void service(HttpServletRequest req ,HttpServletResponse res)throws IOException{
        String noquery =req.getParameter("searchBar");
        PrintWriter out=res.getWriter();

        out.println("<h1 class= \"\"'+\"header\"+\"'+\">Output of Search</h1>");
        out.println("<link rel=\"stylesheet\" href=\"stylesheet1.css\">");

        try {
            try {
                Class.forName ("com.mysql.jdbc.Driver");
            } catch (ClassNotFoundException e) {
                e.printStackTrace();
            }
        }
        // String stmt1;
        Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/WAD_DB","user","admin");
        out.println("<p>Connection establishment successful.....</p>");
        Statement st=con.createStatement();

        String stmt="SELECT SAILORS.SID AS 'SAILOR ID', \r\n"
            + "SAILORS.SNAME AS 'SAILOR NAME', \r\n"
            + "SAILORS.SAGE AS 'SAILOR AGE', \r\n"
            + "SAILORS.RATING AS 'SAILOR RATING', \r\n"
            + "RESERVES.BID AS 'RESERVED BOAT', \r\n"
            + "BOATS.BNAME AS 'BOAT NAME', \r\n"
            + "BOATS.COLOR AS 'BOAT COLOR', \r\n"
            + "RESERVES.DAY\r\n"
            + "FROM RESERVES\r\n"
            + "INNER JOIN BOATS ON RESERVES.BID=BOATS.BID\r\n"
            + "INNER JOIN SAILORS ON RESERVES.SID=SAILORS.SID ";
        String[] str = noquery.split(", ",6 );
        String arr[]= {"SAILOR ID","SAILOR NAME","SAILOR AGE","SAILOR RATING","RESERVED BOAT","BOAT NAME","BOAT COLOR","DAY"};
        ArrayList<String> sid=new ArrayList<String>();
        ArrayList<String> sname=new ArrayList<String>();
        ArrayList<String> sage=new ArrayList<String>();
        ArrayList<String> rating=new ArrayList<String>();
        ArrayList<String> bid=new ArrayList<String>();
        ArrayList<String> bname=new ArrayList<String>();
        ArrayList<String> color=new ArrayList<String>();
        ArrayList<String> day=new ArrayList<String>();
        ArrayList<String> ar=new ArrayList<String>();
        for (String i:str) {
            ar.add(i);
        }
        ResultSet rs = st.executeQuery( stmt+"");
        String s="WHERE ";
        rs.next();
        while(rs.next())
        {
            sid.add(rs.getString(arr[0]));
            sname.add(rs.getString(arr[1]));
            sage.add(rs.getString(arr[2]));
            rating.add(rs.getString(arr[3]));
            bid.add(rs.getString(arr[4]));
            bname.add(rs.getString(arr[5]));
            color.add(rs.getString(arr[6]));
            day.add(rs.getString(arr[7]));
        }
        for (String i:ar) {
            if (sid.contains(i)) {
                s=s+"sailors.sid= "+i;
            }
            else if(sname.contains(i)){

```

```

        s=s+"sailors.sname= \"" + i + "\"";
    }
    else if(sage.contains(i)){
        s=s+"sailors.sage= "+i;
    }
    else if(rating.contains(i)){
        s=s+"rating= "+i;
    }
    else if(bid.contains(i)){
        s=s+"reserves.bid= "+i;
    }
    else if(bname.contains(i)){
        s=s+"boats.bname= \"" + i + "\"";
    }
    else if(color.contains(i)){
        s=s+"boats.color= \"" + i + "\"";
    }
    else if(day.contains(i)){
        s=s+"reserves.day= \"" + i + "\"";
    }
    else {
        out.println("invalid attribute value"+i);
    }
    s+=" and ";
}
int l=s.length();
String Final=stmt+s.substring(0,l-5)+" ";
ResultSet rs1=st.executeQuery(Final);
out.println("<table style='width:100%'>\r\n"
+ "<tr>\r\n"
+ " <th>"+arr[0]+"</th>\r\n"
+ " <th>"+arr[1]+"</th>\r\n"
+ " <th>"+arr[2]+"</th>\r\n"
+ " <th>"+arr[3]+"</th>\r\n"
+ " <th>"+arr[4]+"</th>\r\n"
+ " <th>"+arr[5]+"</th>\r\n"
+ " <th>"+arr[6]+"</th>\r\n"
+ " <th>"+arr[7]+"</th>\r\n"
+ "</tr>\r\n");
while(rs1.next()) {
    out.println("<tr>\r\n"
+ " <td>"+rs1.getObject(1)+"</td>\r\n"
+ " <td>"+rs1.getObject(2)+"</td>\r\n"
+ " <td>"+rs1.getObject(3)+"</td>\r\n"
+ " <td>"+rs1.getObject(4)+"</td>\r\n"
+ " <td>"+rs1.getObject(5)+"</td>\r\n"
+ " <td>"+rs1.getObject(6)+"</td>\r\n"
+ " <td>"+rs1.getObject(7)+"</td>\r\n"
+ " <td>"+rs1.getObject(8)+"</td>\r\n"
+ "</tr>\r\n");
}
out.println("</table>");
con.close();
out.println("<p>query executed.....</p>");
}catch(SQLException e) {
    out.println("<p>error.....,</p>");
    e.printStackTrace();
}

}

}

//<table style="width:100%">
//<tr>
// <th>Company</th>
// <th>Contact</th>
// <th>Country</th>
//</tr>
//<tr>
// <td>Alfreds Futterkiste</td>
// <td>Maria Anders</td>
// <td>Germany</td>
//</tr>
//<tr>
// <td>Centro comercial Moctezuma</td>
// <td>Francisco Chang</td>
// <td>Mexico</td>

```

```
//</tr>
//</table>
```

```
//rs1.getObject(1)+" "+rs1.getObject(2)+" "+rs1.getObject(3)+" "+rs1.getObject(4)+" "+rs1.getObject(5)+" "+rs1.getObject(6)+" "+rs1.getObje
```

The user also need to login to the account in order to executes all these functions

- [Authen.java](#) - Login authentication for the right user to give rights.

```
package cscorner;

import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletResponse;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class Authen extends HttpServlet{
    int flag=0;
    public void service(HttpServletRequest req ,HttpServletResponse res)throws IOException{
        res.setContentType("text/html");
        String user=req.getParameter("usern");
        String pwd=req.getParameter("pwd");
        PrintWriter out=res.getWriter();
        if (user.equals("admin") && pwd.equals("admin")) {
            this.flag=1;
            res.sendRedirect(req.getContextPath() + "/Main.jsp");
        }else {
            out.println("invalid credentials");
        }
    }
}
```

Outputs:

Login.jsp:

Login to Anchored

Username

Password

Logging in with default user and password - admin & admin.

Main.jsp:

AnchoredLogout

Add/Remove a Sailor:

Add ▾

Enter name to add:

Enter Age:

Enter Rating:

Submit

Add/Remove a Boat:

Add ▾

Enter name to add:

Enter color:

Submit

Reserve a Boat:

Enter sailor id:

Enter boat id:

Submit

View details of a Sailor/Boat/Reserves by:

Enter keywords to search the db:

Submit

The javascript functions does the job in changing the textboxes view based on the selected input Add/Remove.

Add/Remove a Sailor:

Remove ▾

Enter id to remove:

Submit

Add/Remove a Boat:

Remove ▾

Enter id to remove:

Submit

Reserve a Boat:

Enter sailor id:

Enter boat id:

Submit

View details of a Sailor/Boat/Reserves by:

Enter keywords to search the db:

Submit

Adding 'Michael, 27, 9.6' into the sailors.

Add/Remove a Sailor:

Add

We can see that the entered data is inserted into the database at the end:

Output of Sailors Table

Connection establishment successful.....
add option selected.....

SID	NAME	AGE	RATING
1	Sandeep	20	10.0
2	Bogghu	18	8.0
3	DC	20	1.0E8
4	Sad	12	1.0
5	Jayanth	11	0.0
6	boghesh	20	12.0
7	Michael	27	9.6

query executed.....

As we know that sid of Michael is 7, we can try remove that sailor.

Add/Remove a Sailor:

Remove

Connection establishment successful.....
remove option selected.....

SID	NAME	AGE	RATING
1	Sandeep	20	10.0
2	Bogghu	18	8.0
3	DC	20	1.0E8
4	Sad	12	1.0
5	Jayanth	11	0.0
6	boghesh	20	12.0

query executed.....

We can also do the same for boats.

Add/Remove a Boat:

Add

add option selected.....

BID	BOAT_NAME	BOAT_COLOR
1	India	blue
2	Sad	blue
3	Angry	red
4	ABDC	pink
5	Naruto	Orange

query executed.....

remove option selected.....

BID	BOAT_NAME	
1	India	blue
2	Sad	blue
3	Angry	red
4	ABDC	pink

query executed.....

We can also link a boat with a sailor, for example ("Michael" has a boat named 'Sunny'):

Add/Remove a Sailor:

10	100ghesh
7	Michael

query executed

Add/Remove a Boat:

1	ABDC
5	Sunny

query executed.....

Reserve a Boat:

SID	BID	
1	1	2023-04-02
1	3	2023-04-02
2	2	2023-04-02
3	4	2023-04-02
7	5	2023-04-06

query executed.....,

We can search for a sailor with keywords...

View details of a Sailor/Boat/Reserves by:

Connection establishment successful.....

SAILOR ID	SAILOR NAME	SAILOR AGE	SAILOR RATING	RESERVED BOAT	BOAT NAME	BOAT COLOR	DAY
7	Michael	20	10.0	5	Sunny	Orange	2023-04-06
7	Michael	20	10.0	6	Going Merry	Blue	2023-04-06

query executed.....

And finally we can logout by clicking on the logout button. All the data is stored in the database, and can be accessed by the same user later.

Conclusion:

In conclusion, the Anchors web application is a powerful tool for sailors and boat owners to manage their vessel and crew data. Built from scratch using HTML, CSS, JavaScript, Servlet, JDBC connections, and MySQL database, Anchors provides a user-friendly platform for users to enter and store information about their boats and sailors. The application allows users to bind boats to sailors, remove sailors or boats from the database, and search for specific sailors or boats using keywords.

Anchors is designed with user-friendliness in mind, making it easy for even those with limited technical knowledge to use. The application is optimized for performance, ensuring that users can access and manage their data quickly and easily. Additionally, the application is built using best practices to ensure that user data is secure and protected at all times.