

WT Assignment/Practicals2

Name: - Kaustubh Shrikant Kabra

Class: - Third Year Engineering(Computer)

Div.: - A **ERP No.:** - 38

Website: Frontend of a landing page for a camera application.

HTML code:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Capture</title>
    <link
href="https://fonts.googleapis.com/css?family=Montserrat|Ubuntu" rel="stylesheet">
    <!-- Google Fonts -->
    <link rel="preconnect"
href="https://fonts.googleapis.com">
    <link rel="preconnect"
href="https://fonts.gstatic.com"
crossorigin>
    <link
href="https://fonts.googleapis.com/css2?family=Montserrat:wght@900&display=swap"
rel="stylesheet">
    <!-- CSS Stylesheets -->
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css">
    <link rel="stylesheet"
href="css/styles.css" />

    <!-- Font Awesome Styles -->
    <script defer
src="https://kit.fontawesome.com/5251786828.js" crossorigin="anonymous"></script>

    <!-- Bootstrap Scripts -->
    <script
src="https://code.jquery.com/jquery-3.2.1.slim.min.js"
integrity="sha384-
KJ3o2DKtIkVYIK3UENzmM7KCKRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"
crossorigin="anonymous"
></script>
    <script
```

```
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"
"
integrity="sha384-
ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"
crossorigin="anonymous"
></script>
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"
integrity="sha384-
JZR6Spejh4U02d8j0t6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmY1"
crossorigin="anonymous"
></script>
</head>

<body>
  <section class="colored-section"
id="title">
    <div class="container-fluid">
      <!-- Nav Bar -->
      <nav class="navbar navbar-expand-
lg navbar-dark">
        <a class="navbar-brand"
href="">capture</a>

        <button
class="navbar-toggler"
type="button"
data-toggle="collapse"
data-
target="#navbarSupportedContent"
aria-
controls="navbarSupportedContent"
aria-expanded="false"
aria-label="Toggle navigation"
>
          <span class="navbar-toggler-
icon"></span>
        </button>
        <div class="collapse navbar-
collapse" id="navbarSupportedContent">
          <ul class="navbar-nav ml-
auto">
            <li class="nav-item">
              <a class="nav-link"
href="#footer">Contact</a>
            </li>
            <li class="nav-item">
              <a class="nav-link"
href="#pricing">Pricing</a>
            </li>
            <li class="nav-item">
              <a class="nav-link"
href="#cta">Download</a>
            </li>
```

```

        </ul>
    </div>
</nav>

<!-- Title -->

<div class="row">
    <div class="col-lg-6">
        <h1 class="big-heading">Capture what can't be
captured.</h1>
        <h4>With our photo capturing
app.</h4>
        <button type="button"
class="btn btn-outline-dark btn-lg
download-button"><i class="fab fa-
apple"></i> Download</button>
        <button type="button"
class="btn btn-outline-light btn-lg
download-button"><i class="fab fa-google-
play"></i> Download</button>
    </div>

    <div class="col-lg-6">
        
    </div>
</div>
</div>
</section>

<!-- Features -->

<section class="white-section"
id="features">

    <div class="row">

        <div class="feature-box col-lg-4">
            <i class="icon fas fa-check-
circle fa-4x"></i>
            <h3>Easy to use.</h3>
            <p>Easy to operate and click
dynamic views</p>
        </div>

        <div class="feature-box col-lg-4">
            <i class="icon fas fa-bullseye
fa-4x"></i>
            <h3>Elite Clientele</h3>
            <p>Upload your photography and
earn likes!</p>
        </div>

        <div class="feature-box col-lg-4">
            <i class="icon fas fa-heart fa-
4x"></i>

```

```

        <h3>Guaranteed to work.</h3>
        <p>Bugs free and effective image
capturing.</p>
    </div>

</div>

</section>

<!-- Testimonials -->

<section class="colored-section"
id="testimonials">

    <div id="testimonial-carousel"
class="carousel slide" data-ride="false">
        <div class="carousel-inner" >

            <div class="carousel-item active">
                <h2>Great app.The quality of
capturing scenic beauty with this app is
remarkable.</h2>
                
                <em>Pebbles, New York</em>
            </div>
            <div class="carousel-item" >
                <h2 class="testimonial-
text">Posting my work gained me tons of
likes and followers!</h2>
                
                <em>Beverly, Illinois</em>
            </div>

        </div>

        <a class="carousel-control-prev"
href="#testimonial-carousel" role="button"
data-slide="prev">
            <span class="carousel-control-prev-
icon"></span>
        </a>

        <a class="carousel-control-next"
href="#testimonial-carousel" role="button"
data-slide="next">
            <span class="carousel-control-next-
icon"></span>
        </a>
    </div>

```

```

</section>

<!-- Press -->

<section class="colored-section"
id="press">

    
    
    
    
</section>

<!-- Pricing -->

<section class="white-section"
id="pricing">
    <h2>Our plans include a range of
options for you to choose.</h2>
    <p>Simple and affordable price plans
for you.</p>

    <div class="row">
        <div class="pricing-column col-lg-4
col-md-6">
            <div class="card">
                <div class="card-header">
                    <h3>Free Version</h3>
                </div>
                <div class="card-body">
                    <h2>Free</h2>
                    <p>Unlimited captures</p>
                    <p>Limited cloud storage</p>
                    <p>No posting services</p>
                    <button class="btn btn-lg btn-
block btn-outline-dark" type="button">Sign
Up</button>
                </div>
            </div>
        </div>

        <div class="pricing-column col-lg-4
col-md-6">
            <div class="card">
                <div class="card-header">
                    <h3>Monthly Plan</h3>
                </div>
                <div class="card-body">
                    <h2>$49 / mo</h2>
                    <p>Unlimited captures</p>
                    <p>Unlimited cloud storage</p>

```

```

        <p>Limited Posting
services</p>
        <button class="btn btn-lg
btn-block btn-dark" type="button">Sign
Up</button>
    </div>
</div>

    <div class="pricing-column col-lg-4
col-md-12">
        <div class="card">
            <div class="card-header">
                <h3>Annual Plan</h3>
            </div>
            <div class="card-body">
                <h2>$299 / yr</h2>
                <p>Unlimited captures</p>
                <p>Unlimited cloud storage</p>
                <p>Unlimited Posting
services</p>
                <button class="btn btn-lg btn-
block btn-dark" type="button">Sign
Up</button>
            </div>
        </div>
    </div>

</div>

</section>

<!-- Call to Action -->
<div class="container"></div>
<div class="bottom row">
    <div class="col">
        <section class="colored-
section" id="cta">
            <h3 class="text-bottom">Get your app
right away and 'Capture' the world!</h3>
            <button type="button" class="btn
btn-outline-dark btn-lg download-
button"><i class="fab fa-apple"></i>
Download</button>
            <button type="button"
class="btn btn-outline-light btn-lg
download-button"><i class="fab fa-google-
play"></i> Download</button>
        </section>
    </div>
</div>

<!-- Footer -->

```

```

    <footer class="white-section"
id="footer">
    <ul class="social-network social-
circle">
        <li><a href="#"
class="icoFacebook" title="Facebook"><i
class="fa fa-facebook"></i></a></li>
        <li><a href="#" class="icoTwitter"
title="Twitter"><i class="fa fa-
twitter"></i></a></li>
        <li><a href="#" class="icoGoogle"
title="Google +"><i class="fa fa-google-
plus"></i></a></li>
        <li><a href="#"
class="icoInstagram" title="Instagram"><i
class="fa fa-instagram" aria-
hidden="true"></i></a></li>

    </ul>
    <p>© Copyright 2018 TinDog</p>
</footer>
</body>
</html>

```

CSS code

```

#title{
    font-family: 'Montserrat', sans-serif;
}
/* containers*/
.colored-section{
    background-color: #ff4c68;
    color:#fff;
}
.container-fluid{
    padding: 7% 15%;
}
#title .container-fluid{
    padding: 3% 15% 7%;
}
.big-heading{
font-family: 'Montserrat', sans-serif;
font-size: 3rem;
}
h1,h2,h3,h4,h5,h6{
    font-family: 'Montserrat', sans-serif;
    font-style: bold;
}
h2{
    font-family: 'Montserrat';

```

```

    font-style: bold;
    font-size: 3rem;
    line-height:1.5 ;
}
p{
    color: #afafaf;
}
body{
    font-family: 'Montserrat';
    text-align: center;
}
/*Navigation Bar*/
.navbar{
    padding-bottom: 4.5rem;
}
.navbar-brand{
    font-family: "Ubuntu";
    font-size: 2.5rem;
    font-weight: bold;
}
.nav-item{
    padding: 0 18px;
}
.nav-link{
    font-size: 1.2rem;
    font-family: 'Montserrat-Light', sans-
serif;
    color:#fff;
}
.download-button{
margin: 5% 3% 5% 0%
}
/* Title Image */
.title-image{
    width: 60%;
    transform: rotate(25deg);
    position: absolute;
    right: 30%;
}
/* Features */
#features{
    padding: 7% 15%;
    background-color: white;
    position: relative;

```

```

}

.feature-box{

    padding: 5%;

}

.icon{
    color: #ef8172;
    margin-bottom: 1rem;
}

.icon:hover{
    color: #ff4c68;
}

/* Testimonials Section */
#testimonials{
    padding: 7% 15%;

    background-color: #ef8172;
    color: #fff;
}

.testimonial-image{
    width: 10%;
    border-radius: 100%;
    margin: 20px;
}

#press{
    background-color: #ef8172;

    padding-bottom: 3%;
}

.press-logo{
    width: 15%;
    margin: 20px 20px 50px;
}

.carousel-item{
    padding: 7% 15%;
}

/* pricing section */
#pricing{
    padding: 100px;
}

.pricing-column{
    padding: 3% 2%;
}

@media (max-width: 1028px) {

```

```

    #title{
        text-align: center;
    }
    .title-image{
        position: static;
        transform: rotate(0);
    }
}

#cta{

    font-family: 'Montserrat', sans-serif;
    font-size: 3rem;
    padding: 7% 15%;
}

.text-bottom{
    color: #fff;
    font-size: 3.5rem;
    line-height: 1.5;
}

.bottom{
    background-color: #ff4c68;

    padding: 3% 15% 7%;
}

#footer{

padding: 7% 15%;
}

/*footer social icons */
ul.social-network {
    list-style: none;
    display: inline;
    margin-left:0 !important;
    padding: 0;
}
ul.social-network li {
    display: inline;
    margin: 0 5px;
}

/*footer social icons */
.social-network a.icoRss:hover {
    background-color: #F56505;
}
.social-network a.icoFacebook:hover {
    background-color:#3B5998;
}
.social-network a.icoTwitter:hover {
    background-color:#33ccff;
}
.social-network a.icoGoogle:hover {

```

```

background-color:#BD3518;
}
.social-network a.icoVimeo:hover {
background-color:#0590B8;
}
.social-network a.icoLinkedin:hover {
background-color:#007bb7;
}
.social-network a.icoInstagram:hover {
background-color: orangered;
}
.social-network a.icoRss:hover i, .social-
network a.icoFacebook:hover i, .social-
network a.icoTwitter:hover i,
.social-network a.icoGoogle:hover i,
.social-network a.icoVimeo:hover i,
.social-network a.icoLinkedin:hover i,
.social-network a.icoInstagram:hover i {
color:#fff;
}
a.socialIcon:hover, .socialHoverClass {
color:#44BCDD;
}

.social-circle li a {
display:inline-block;
position:relative;
margin:0 auto 0 auto;
-moz-border-radius:50%;
-webkit-border-radius:50%;
border-radius:50%;
text-align:center;
width: 50px;

```

```

height: 50px;
font-size:20px;
}
.social-circle li i {
margin:0;
line-height:50px;
}

.social-circle li a:hover i,
.triggeredHover {
-moz-transform: rotate(360deg);
-webkit-transform: rotate(360deg);
-ms--transform: rotate(360deg);
transform: rotate(360deg);
-webkit-transition: all 0.2s;
-moz-transition: all 0.2s;
-o-transition: all 0.2s;
-ms-transition: all 0.2s;
transition: all 0.2s;
}
.social-circle i {
color: black;
-webkit-transition: all 0.8s;
-moz-transition: all 0.8s;
-o-transition: all 0.8s;
-ms-transition: all 0.8s;
transition: all 0.8s;
}

```


Output Screenshots:


capture Contact Pricing Download

**Capture what can't
be captured.**


With our photo capturing app.

[Download](#)
[Download](#)






Easy to use.
Easy to operate and click



Elite Clientele
Upload your photography and



**Guaranteed to
work.**

< **Great app. The quality of capturing scenic beauty with this app is remarkable.** >



Pebbles, New York

TC TechCrunch

TNW
THE NEXT WEB

BUSINESS INSIDER

Mashable

Our plans include a range of options for you to choose.

Simple and affordable price plans for you.

Free Version

Free

Unlimited captures

Limited cloud storage

No posting services

[Sign Up](#)

Monthly Plan

\$49 / mo

Unlimited captures

Unlimited cloud storage

Limited Posting services

[Sign Up](#)

Annual Plan

\$299 / yr

Unlimited captures

Unlimited cloud storage

Unlimited Posting services

[Sign Up](#)

**Get your app right away
and 'Capture' the world!**

 **Download**

 **Download**

© Copyright 2018 TinDog

WT Assignment/Practicals -03

Name: - Kaustubh Shrikant Kabra

Class: - Third Year Engineering(Computer)

Div.: - A **ERP No.:** - 38

employee.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="employee.xsl"?>
<!DOCTYPE company SYSTEM "employee.dtd">
<company>
  <employee>
    <username>Akash123</username>
    <password>akash123</password>
    <firstname>Akash</firstname>
    <lastname>Mete</lastname>
    <gender>Male</gender>
    <email>akashmete129@gmail.com</email>
    <position>Software Developer</position>
    <salary>9.5lpa</salary>
    <contact>+91 7051794388</contact>
  </employee>

  <employee>
    <username>AlexWest</username>
    <password>alex123</password>
    <firstname>Alex</firstname>
    <lastname>West</lastname>
    <gender>Female</gender>
    <email>alex123@gmail.com</email>
    <position>Software Developer</position>
    <salary>8.5lpa</salary>
    <contact>+91 5555943848</contact>
  </employee>

  <employee>
    <username>BarryAllen</username>
    <password>barry123</password>
    <firstname>Barry</firstname>
    <lastname>Allen</lastname>
    <gender>Male</gender>
    <email>barryallen@gmail.com</email>
    <position>Frontend Developer</position>
    <salary>7.5lpa</salary>
    <contact>+91 6666794388</contact>
  </employee>
</company>
```

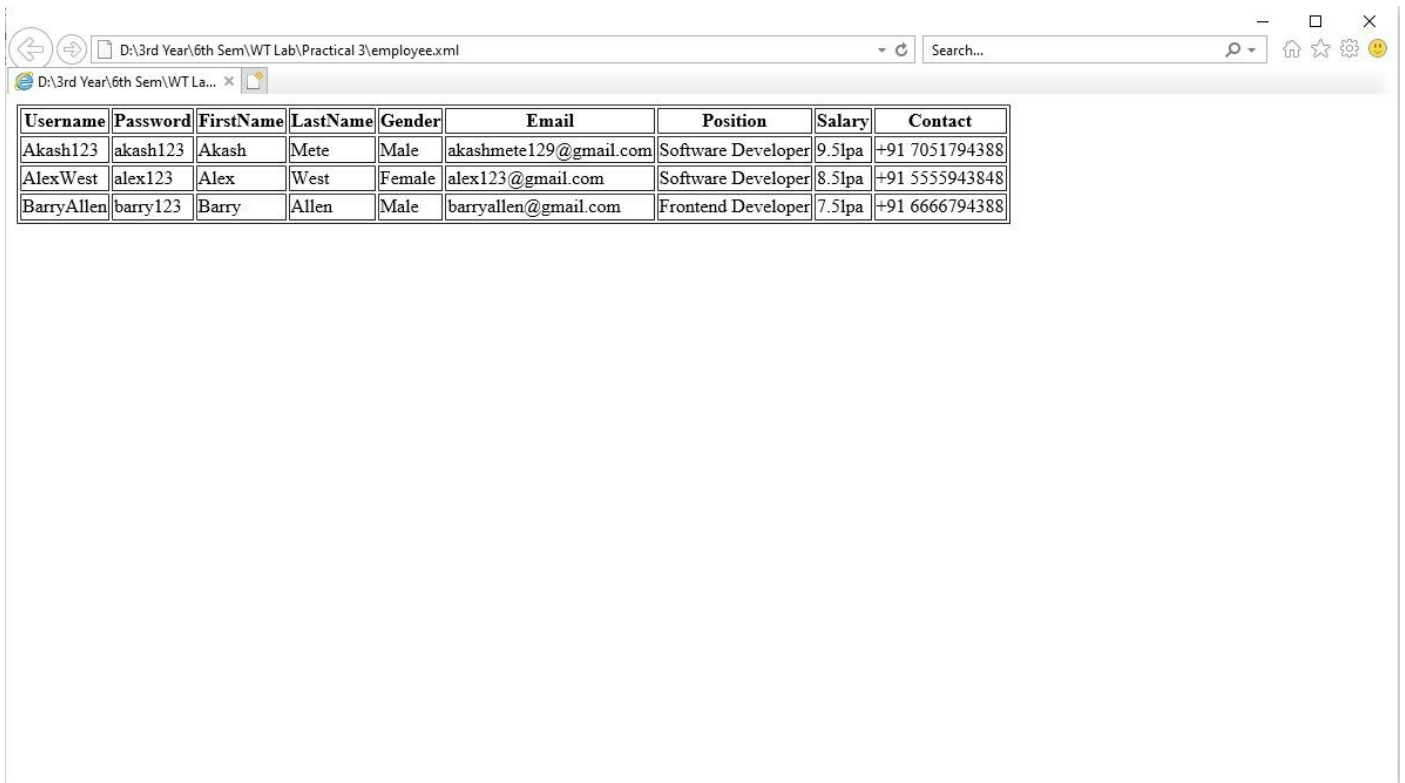
employee.dtd

```
<!ELEMENT company (employee+)>
<!ELEMENT employee
(username,password,firstname,lastname,gender,email,position,salary,contact)>
<!ELEMENT username (#PCDATA)>
<!ELEMENT password (#PCDATA)>
<!ELEMENT firstname (#PCDATA)>
<!ELEMENT lastname (#PCDATA)>
<!ELEMENT gender (#PCDATA)>
<!ELEMENT email (#PCDATA)>
<!ELEMENT position (#PCDATA)>
<!ELEMENT salary (#PCDATA)>
<!ELEMENT contact (#PCDATA)>
```

employee.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="employee.xsl"?>
<!DOCTYPE company SYSTEM "employee.dtd">
<company><employee>
    <username>Akash123</username>
    <password>akash123</password>
    <firstname>Akash</firstname>
    <lastname>Mete</lastname>
    <gender>Male</gender>
    <email>akashmete129@gmail.com</email>
    <position>Software Developer</position>
    <salary>9.5lpa</salary>
    <contact>+91 7051794388</contact>
</employee>
<employee>
    <username>AlexWest</username>
    <password>alex123</password>
    <firstname>Alex</firstname>
    <lastname>West</lastname>
    <gender>Female</gender>
    <email>alex123@gmail.com</email>
    <position>Software Developer</position>
    <salary>8.5lpa</salary>
    <contact>+91 5555943848</contact>
</employee>
<employee>
    <username>BarryAllen</username>
    <password>barry123</password>
    <firstname>Barry</firstname>
    <lastname>Allen</lastname>
    <gender>Male</gender>
    <email>barryallen@gmail.com</email>
    <position>Frontend Developer</position>
    <salary>7.5lpa</salary>
    <contact>+91 6666794388</contact>
</employee></company>
```

Output:



The screenshot shows a web browser window with the address bar displaying the file path: D:\3rd Year\6th Sem\WT Lab\Practical 3\employee.xml. The browser's address bar also includes a search field and navigation icons. The main content area displays the XML data as a table with the following structure:

Username	Password	FirstName	LastName	Gender	Email	Position	Salary	Contact
Akash123	akash123	Akash	Mete	Male	akashmete129@gmail.com	Software Developer	9.5lpa	+91 7051794388
AlexWest	alex123	Alex	West	Female	alex123@gmail.com	Software Developer	8.5lpa	+91 5555943848
BarryAllen	barry123	Barry	Allen	Male	barryallen@gmail.com	Frontend Developer	7.5lpa	+91 6666794388

WT Assignment/Practicals-04

Name: - Kaustubh Shrikant Kabra

Class: - Third Year Engineering(Computer)

Div.: - A **ERP No.:** - 38

ReactJs Application

Index.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <link rel="icon" href="%PUBLIC_URL%/favicon.ico" />
    <link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link href="https://fonts.googleapis.com/css2?family=Inconsolata&display=swap"
rel="stylesheet">
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <meta name="theme-color" content="#000000" />
    <meta
      name="description"
      content="Web site created using create-react-app"
    />
    <link rel="apple-touch-icon" href="%PUBLIC_URL%/logo192.png" />

    <link rel="manifest" href="%PUBLIC_URL%/manifest.json" />
    <title>Calculator App</title>
  </head>
  <body>
    <noscript>You need to enable JavaScript to run this app.</noscript>
    <div id="root"></div>

  </body>
</html>
```

Index.js

```
import React from 'react';
import ReactDOM from 'react-dom';
import './index.css';
import App from './App';

ReactDOM.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>,
  document.getElementById('root')
);
```

OperationButton.js

```
import { ACTIONS } from "../App"

export default function OperationButton({ dispatch, operation }) {
  return(
    <button
      onClick={() => dispatch({ type: ACTIONS.CHOOSE_OPERATION, payload: {operation}})}
    >
      {operation}
    </button>
  )}
}
```

DigitButton.js

```
import { ACTIONS } from "../App"

export default function DigitButton({ dispatch, digit }) {
  return(
    <button
      onClick={() => dispatch({ type: ACTIONS.ADD_DIGIT, payload: {digit}})}
    >
      {digit}
    </button>
  )}
}
```

App.js

```
import "../App.css";
import React, { useReducer } from "react";
import DigitButton from "../DigitButton";
import OperationButton from
"../OperationButton";

export const ACTIONS = {
  ADD_DIGIT: "add-digit",
  CHOOSE_OPERATION: "choose-operation",
  CLEAR: "clear",
  DELETE_DIGIT: "delete-digit",
  EVALUATE: "evaluate",
};

function reducer(state, { type, payload }) {
  switch (type) {
    case ACTIONS.ADD_DIGIT:
      if (state.overwrite) {
        return {
          ...state,
          currentOperand:payload.digit,
          overwrite: false,

```

```
        }
      }

      if (payload.digit === "0" &&
state.currentOperand === "0") {
        return state;
      }
      if (payload.digit === "." &&
state.currentOperand.includes(".")) {
        return state;
      }

      return {
        ...state,
        currentOperand:
`${state.currentOperand} ||
""}${payload.digit}`,
      };

    case ACTIONS.CHOOSE_OPERATION:
      if (state.currentOperand == null &&
state.previousOperand == null) {
        return state;
      }

```

```

    if (state.currentOperand == null) {
      return {
        ...state,
        operation: payload.operation,
      }
    }

    if (state.previousOperand == null) {
      return {
        ...state,
        operation: payload.operation,
        previousOperand:
state.currentOperand,
        currentOperand: null,
      };
    }

    return {
      ...state,
      previousOperand: evaluate(state),
      operation: payload.operation,
      currentOperand: null,
    };

    case ACTIONS.CLEAR:
      return {};

    case ACTIONS.DELETE_DIGIT:
      if (state.overwrite) {
        return {
          ...state,
          overwrite: false,
          currentOperand: null
        }
      }
      if (state.currentOperand == null)
return state
      if (state.currentOperand.length ===
1){
        return { ...state, currentOperand:
null}
      }

      return {
        ...state,
        currentOperand:
state.currentOperand.slice(0, -1)
      }
    case ACTIONS.EVALUATE:
      if (
        state.operation == null ||
        state.currentOperand == null ||
        state.previousOperand == null
      ) {
        return state
      }

      return {

```

```

        ...state,
        overwrite: true,
        previousOperand: null,
        operation: null,
        currentOperand: evaluate(state),
      }

      default:
        return {}
    }
  }

function evaluate({ currentOperand,
previousOperand, operation }) {
  const prev =
parseFloat(previousOperand);
  const current =
parseFloat(currentOperand);
  if (isNaN(prev) || isNaN(current))
return "";
  let computation = "";

  switch (operation) {
    case "+":
      computation = prev + current;
      break;
    case "-":
      computation = prev - current;
      break;
    case "*":
      computation = prev * current;
      break;
    case "÷":
      computation = prev / current;
      break;

    default:
      return {}
  }

  return computation.toString()
}

const INTEGER_FORMATTER = new
Intl.NumberFormat("en-us", {
  maximumFractionDigits: 0,
})

function formatOperand(operand) {
  if(operand == null) return
  const [integer,decimal] =
operand.split('.')
  if(decimal == null) return
INTEGER_FORMATTER.format(integer)
  return
`${INTEGER_FORMATTER.format(integer)}.${de
cimal}`
}

```

```

function App() {
  const [{ currentOperand,
previousOperand, operation }, dispatch] =
useReducer(
  reducer,
  {}
);

  return (
    <div className="calculator-grid">
      <h1>Calculator App</h1>
      <div className="output">
        <div className="previous-operand">
          {formatOperand(previousOperand)}
        </div>
        <div className="current-
operand">{formatOperand(currentOperand)}</
div>
      </div>
      <button
        className="span-two"
        onClick={() => {
          dispatch({ type: ACTIONS.CLEAR
});
        }}
      >
        AC
      </button>
      <button onClick={() =>
dispatch({type:
ACTIONS.DELETE_DIGIT})}>DEL</button>
      <OperationButton operation="÷"
dispatch={dispatch} />
      <DigitButton digit="1"
dispatch={dispatch} />
      <DigitButton digit="2"
dispatch={dispatch} />
      <DigitButton digit="3"
dispatch={dispatch} />
      <OperationButton operation="*"
dispatch={dispatch} />
      <DigitButton digit="4"
dispatch={dispatch} />
      <DigitButton digit="5"
dispatch={dispatch} />
      <DigitButton digit="6"
dispatch={dispatch} />
      <OperationButton operation="+"
dispatch={dispatch} />
      <DigitButton digit="7"
dispatch={dispatch} />
      <DigitButton digit="8"
dispatch={dispatch} />
      <DigitButton digit="9"
dispatch={dispatch} />
      <OperationButton operation="-"
dispatch={dispatch} />
    </div>
  );
}

```

```

      <DigitButton digit="."
dispatch={dispatch} />
      <DigitButton digit="0"
dispatch={dispatch} />
      <button className="span-two"
onClick={() => dispatch({type:
ACTIONS.EVALUATE})}></button>
    </div>
  );
}

export default App;

```

App.css

```

*, *::before, *::after {
  box-sizing: border-box;
}

body {
  margin: 0;
  background-image: linear-gradient(
95.2deg, rgba(173,252,234,1) 26.8%,
rgba(192,229,246,1) 64% );
  max-width: 500px;
  margin: auto;
  font-family: 'Inconsolata', monospace;
}

h1 {
  width: 400px;
  height: 50px;
  font-family: 'Inconsolata', monospace;
  margin: 0%;
  text-align: center;
}

.calculator-grid {
  display: grid;
  margin-top: 2rem;
  grid-template-columns: repeat(4,6rem);
  grid-template-rows: minmax(7rem,auto)
repeat(5,6rem);
  justify-content: center;
}

.calculator-grid > button {
  cursor: pointer;
  font-size: 2rem;
  border: 1px solid white;
  outline: none;
  background-color: rgba(255,255,255,.9);
}

```



```

.calculator-grid >button:hover,
.calculator-grid >button:focus{
  background-color: rgba(255,255,255,.9);
}

.span-two{
  grid-column: span 2;
}

.output{
  grid-column: 1 / -1;
  background-color: rgba(0,0,0,.75);
  display:flex ;
  flex-direction: column;
  align-items: flex-end;
  justify-content: space-around;
  padding: .75rem;
  word-wrap: break-word;
  word-break: break-all;
}

.output .previous-operand {
  color: rgba(255,255,255,.75);
  font-size: 1.5rem;
}

.output .current-operand {
  color: white;
  font-size: 2.5rem;
}

```

Calculator App

AC	DEL		÷
1	2	3	*
4	5	6	+
7	8	9	-
.	0	=	

Calculator App

100 + 10			
AC	DEL		÷
1	2	3	*
4	5	6	+
7	8	9	-
.	0	=	

Calculator App

110			
AC	DEL		÷
1	2	3	*
4	5	6	+
7	8	9	-
.	0	=	

Calculator App

100 ÷
10

AC		DEL	÷
1	2	3	*
4	5	6	+
7	8	9	-
.	0	=	

Calculator App

0.8615384615384616 *
6

AC		DEL	÷
1	2	3	*
4	5	6	+
7	8	9	-
.	0	=	

Calculator App

10

AC		DEL	÷
1	2	3	*
4	5	6	+
7	8	9	-
.	0	=	

Calculator App

5.1692307692307695

AC		DEL	÷
1	2	3	*
4	5	6	+
7	8	9	-
.	0	=	

Calculator App

			0 ÷ 0
AC	DEL	÷	
1	2	3	*
4	5	6	+
7	8	9	-
.	0	=	

Calculator App

			NaN
AC	DEL	÷	
1	2	3	*
4	5	6	+
7	8	9	-
.	0	=	

localhost:3000 says
Invalid Input! The result is NaN

Calculator

0 ÷
0

AC		DEL	÷
1	2	3	*
4	5	6	+
7	8	9	-
.	0	=	

localhost:3000 says
Invalid Input! Please re-enter the values

Calculator

∞

AC		DEL	÷
1	2	3	*
4	5	6	+
7	8	9	-
.	0	=	

WT Assignment/Practicals-05

Name: - Kaustubh Shrikant Kabra

Class: - Third Year Engineering(Computer)

Div.: - A **ERP No.:** - 38

Code:

Register.java

```
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
/* * Servlet implementation class Register */
@WebServlet("/Register")
public class Register extends HttpServlet {
    private static final long serialVersionUID = 1L;
    /* * @see HttpServlet#HttpServlet() */
    public Register() {
        super();
        // TODO Auto-generated constructor stub
    }
    /** @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response) */
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        // TODO Auto-generated method stub
        response.getWriter().append("Served at: ").append(request.getContextPath());
    }
    /*** @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response) */
```

```
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws  
ServletException, IOException {
```

```
    response.setContentType("text/html");
```

```
    PrintWriter out=response.getWriter();
```

```
    String fname = request.getParameter("fname");
```

```
    String lname = request.getParameter("lname");
```

```
    String username = request.getParameter("username");
```

```
    String password = request.getParameter("password");
```

```
    String address = request.getParameter("address");
```

```
    String contact = request.getParameter("contact");
```

```
    try {
```

```
        Class.forName("com.mysql.jdbc.Driver");
```

```
        Connection con =
```

```
        DriverManager.getConnection("jdbc:mysql://localhost:3306/employee","root","Akash@24");
```

```
        PreparedStatement ps = con.prepareStatement("insert into employee  
values(?,?,?, ?, ?, ?)");
```

```
        ps.setString(1, fname);
```

```
        ps.setString(2, lname);
```

```
        ps.setString(3, username);
```

```
        ps.setString(4, password);
```

```
        ps.setString(5, address);
```

```
        ps.setString(6, contact);
```

```
        int i = ps.executeUpdate();
```

```
        if(i > 0) {
```

```
            out.print("You're data has been stored successfully");
```

```
        }
```

```
    }catch(Exception e) {
```

```
        System.out.println(e);
```

```
    }
```

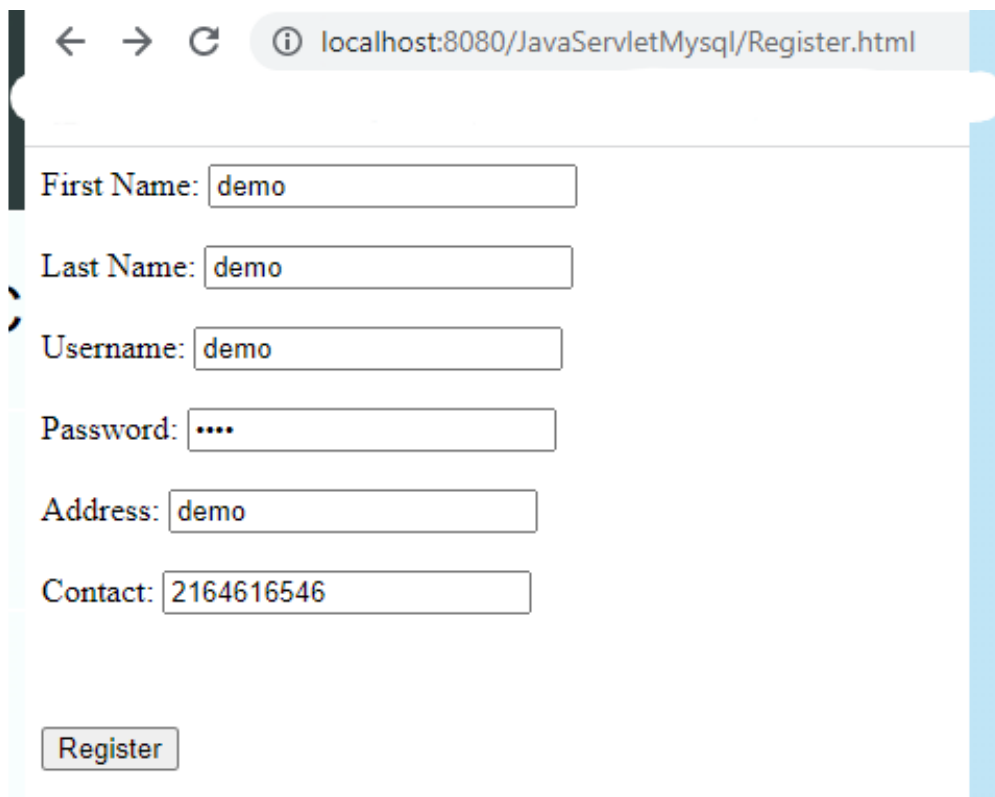
```
}
```

```
}
```

Register.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Registration Form</title>
</head>
<body>
<form action="Register" method="post">
First Name: <input type="text" name="fname" /><br><br>
Last Name: <input type="text" name="lname" /><br><br>
Username: <input type="text" name="username" /><br><br>
Password: <input type="password" name="password" /><br><br>
Address: <input type="text" name="address" /><br><br>
Contact: <input type="text" name="contact" /><br><br>
<br><br>
<input type="submit" value="Register" />
</form>
</body>
</html>
```

Output:



← → ↻ ⓘ localhost:8080/JavaServletMysql/Register.html

First Name:

Last Name:

Username:

Password:

Address:

Contact:



Query 1 x

Limit to 1000 rows

SQLAdditions

Jump to

```
1 use employee;  
2 select * from employee;
```

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Result Grid Filter Rows: Export: Wrap Cell Content:

	first_name	last_name	username	password	address	contact
▶	Akash	Mete	akash	1234	Hadapsar,pune	7057315655
	Onasvee	Banarse	orion	1234	Pune	7057315655
	kaustubh	kabra	KK29	kk29	Jalgaon	7568455777
	demo	demo	demo	1234	demo	2164616546

Result Grid

Form Editor

Field Types

Query Stats

Execution Plan

employee 5 x

Read Only

Context Help

Snippets

You're data has been stored successfully

WT Assignment/Practicals-06

Name: - Kaustubh Shrikant Kabra

Class: - Third Year Engineering(Computer)

Div.: - A **ERP No.:** - 38

Code

Member.java

```
public class Member
{
    private String uname,password,email,phone;

    public Member() {
        super();
    }

    public Member(String uname, String password, String email, String phone) {
        super();
        this.uname = uname;
        this.password = password;
        this.email = email;
        this.phone = phone;
    }

    public String getUname() {
        return uname;
    }

    public void setUname(String uname) {
        this.uname = uname;
    }

    public String getPassword() {
        return password;
    }

    public void setPassword(String password) {
        this.password = password;
    }

    public String getEmail() {
        return email;
    }

    public void setEmail(String email) {
        this.email = email;
    }

    public String getPhone() {
        return phone;
    }

    public void setPhone(String phone) {
        this.phone = phone;
    }
}
```

Register.java

```
import java.io.IOException;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

/** Servlet implementation class Register */
@WebServlet("/Register")

public class Register extends HttpServlet {

    private static final long serialVersionUID = 1L;

    /** @see HttpServlet#HttpServlet() */
    public Register() {
        super();
        // TODO Auto-generated constructor stub
    }

    /** @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response) */
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {

        // TODO Auto-generated method stub

        response.getWriter().append("Served at: ").append(request.getContextPath());
    }

    /** @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response) */
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {

        String uname=request.getParameter("uname");

        String password=request.getParameter("password");

        String email=request.getParameter("email");

        String phone=request.getParameter("phone");

        Member member=new Member(uname, password, email, phone);

        RegisterDao rdao=new RegisterDao();

        String result=rdao.insert(member);

        response.getWriter().println(result);

    }

}
```

RegisterDao.java

```
import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.SQLException;

public class RegisterDao {

    private String dburl = "jdbc:mysql://localhost:3306/demo";

    private String dbuname = "root";

    private String dbpassword = "Akash@24";

    private String dbdriver = "com.mysql.jdbc.Driver";


    public void loadDriver(String dbdriver)

    {

        try {

            Class.forName(dbdriver);

        } catch (ClassNotFoundException e) {

            // TODO Auto-generated catch block

            e.printStackTrace();

        }

    }

    public String insert(Member member) {

        loadDriver(dbdriver);

        try {

            Class.forName("com.mysql.jdbc.Driver");

        } catch (ClassNotFoundException e1) {

            // TODO Auto-generated catch block

            e1.printStackTrace();

        }

        Connection con;

        String result="Data Entered Successfully";

        try {
```

```

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

        con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/demo","root","Akash@24");

        PreparedStatement ps = con.prepareStatement("insert into member
values(?,?,?,?);");

        ps.setString(1, member.getUsername());
        ps.setString(2, member.getPassword());
        ps.setString(3, member.getEmail());
        ps.setNString(4, member.getPhone());
        ps.executeUpdate();

    }catch(SQLException e) {
        result="Data Not Entered Successfully";
        e.printStackTrace();
    }
    return result;

}

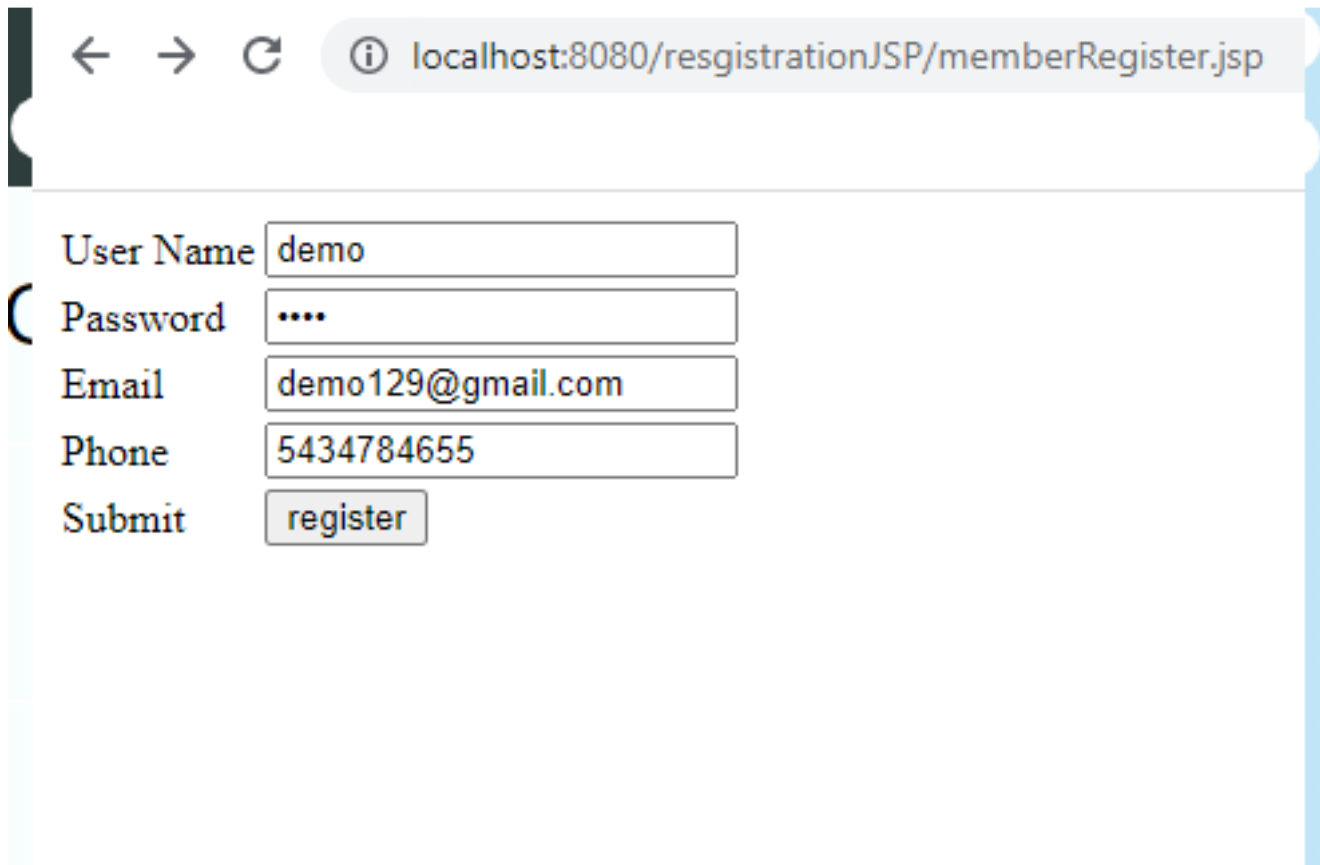
}

```

memberRegister.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
    <form action="Register" method="post">
        <table>
            <tr>
                <td>User Name</td>
                <td><input type="text" name="uname"></td>
            </tr>
            <tr>
                <td>Password</td>
                <td><input type="password" name="password"></td>
            </tr>
            <tr>
                <td>Email</td>
                <td><input type="text" name="email"></td>
            </tr>
            <tr>
                <td>Phone</td>
                <td><input type="text" name="phone"></td>
            </tr>
            <tr>
                <td>Submit</td>
                <td><input type="submit" value="register"></td>
            </tr>
        </table>
    </form>
</body>
</html>
```

Output:



← → ↻ ⓘ localhost:8080/resgistrationJSP/memberRegister.jsp

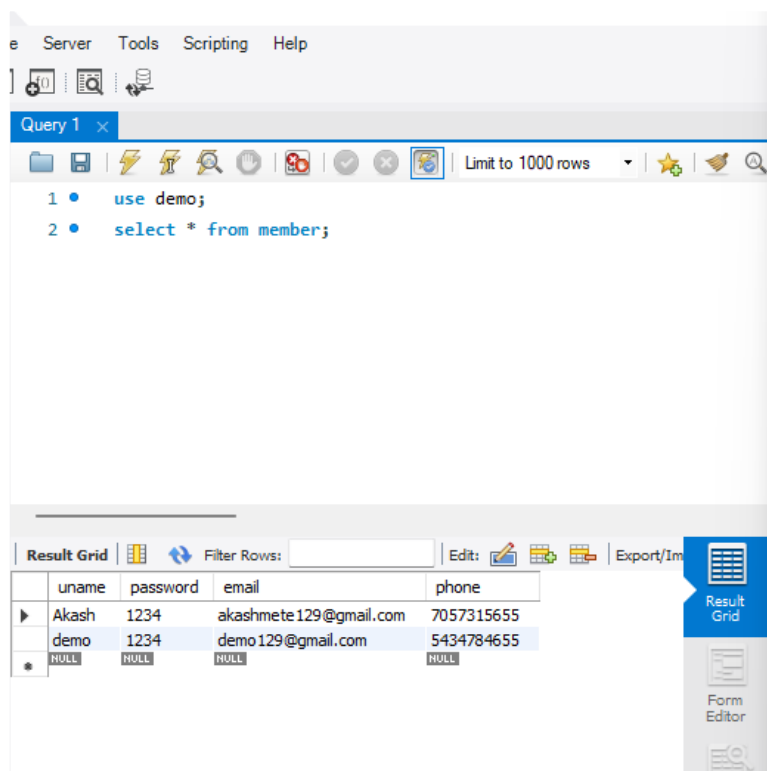
User Name

Password

Email

Phone

Submit



Server Tools Scripting Help

Query 1

```
1 • use demo;  
2 • select * from member;
```

Limit to 1000 rows

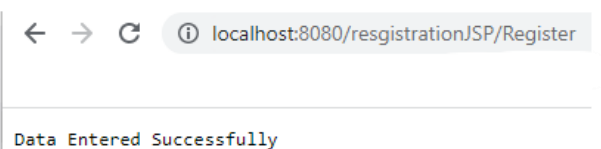
	uname	password	email	phone
▶	Akash	1234	akashmete129@gmail.com	7057315655
	demo	1234	demo129@gmail.com	5434784655
*	NULL	NULL	NULL	NULL

Result Grid

Filter Rows: Edit: Export/Import

Result Grid

Form Editor



← → ↻ ⓘ localhost:8080/resgistrationJSP/Register

Data Entered Successfully

WT Assignment/Practicals - 07

Name: - Kaustubh Shrikant Kabra

Class: - Third Year Engineering(Computer)

Div.: - A ERP No.: - 38

Code:

Index.php

```
EXPLORER  ...  index.php X
  BANKING-SYSTEM
  > public
  alluser.php
  connfile.php M
  createnewuser.php
  getnewuser.php
  index.php
  README.md
  selecteduser.php
  singleuser.php
  transferhistory.php
  transfermoney.php

1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta http-equiv="X-UA-Compatible" content="IE=edge">
7   <meta name="viewport" content="width=device-width, initial-scale=1.0">
8   <title>Banking System</title>
9
10  <!-- Bootstrap & CSS Links -->
11  <link rel="stylesheet" href="/public/css/styles.css">
12  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-EV5TQ3/azprG1Ann3QDgpJLIm9Nao0Yz1ztcQ1wFsp3yD65VohhpuuCOmLASjC" crossorigin="a"
13  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLtUuSfAP+JcXn/tWtIaxVXM" crossorigin="a"
14  <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-KJ3o2DKtIkvYIK3UEENwM7KCKR/rE9/Qp66AZG3wFDMVNA/GpGFF93hXpG5KKN" crossorigin="anonymous"></script>
15  <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.0/umd/popper.min.js" integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibx39j7fakFpskvXuvf80b4Q" crossorigin="anon"
16  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js" integrity="sha384-JZ2R6Sejh4U02d8j0t6vLEHfe/3QGiRR5QX5fFwP11MqUvDayjUarS+76PVCmY1" crossorigin="anonymous"
17  </head>
18
19 <body>
20   <section class="main">
21     <div class="nav-bar container-fluid" style="padding-left: 0px;padding-right: 0px;">
22       <!-- Header -->
23       <!-- Nav Bar -->
24       <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
25         <a class="navbar-brand" href="index.php">Banking System</a>
26
27         <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="
28           <span class="navbar-toggler-icon"></span>
29         </button>
30         <div class="collapse navbar-collapse" id="navbarSupportedContent">
31           <ul class="navbar-nav ml-auto">
32             <li class="nav-item">
33               <a class="nav-link" href="index.php">Home</a>
34             </li>
35             <li class="nav-item">
36               <a class="nav-link" href="alluser.php">User</a>
37             </li>
38             <li class="nav-item">
39               <a class="nav-link" href="transfermoney.php">Transfer Money</a>
40             </li>
41           </ul>
42         </div>
43       </nav>
44     </div>
45
46     <div class="container">
47       <div class="banner-image">
48         <div class="inner-banner-image" style="padding-top: 0px;">
49           
50         </div>
51       </div>
52     </div>
53   </section>
54 </body>
55 </html>
```

connfile.php – connecting to local mysql database

```
<?php
$mysql_hostname = "localhost";
$mysql_user = "root";
$mysql_password = "Akash@24";
$mysql_database = "bankingdata";
$port = 3307;
$conn
= mysqli_connect($mysql_hostname,$mysql_user,$mysql_password,$mysql_database,$port);

if(!$conn){

    die("Could not connect to the database due to the following error -->
".mysqli_connect_error());
}

?>
```

allusers.php – getting all account holders details

```
<table class="center content-table ">
  <tr>
    <th>Id</th>
    <th>Name</th>
    <th>Details</th>
  </tr>
  <?php
    while ($rows = mysqli_fetch_assoc($result)) {
      ?>
      <tr>
        <td><?php echo $rows['id'] ?></td>
        <td><?php echo $rows['acName'] ?></td>
        <td><a href="singleuser.php?id= <?php echo $rows['id']; ?>" class="btn btn-
primary">View Details</a></td>
      </tr>
      <?php
    }
  ?>
</table>
```

View All Users

id	Name	Details
1	Akash	View Details
2	Harsh	View Details
3	Kaustubh	View Details
4	Orion	View Details
5	Siddhi	View Details
6	Vaish	View Details
7	Sanjay	View Details
8	Karan	View Details
9	Arjun	View Details
22	Demo	View Details

Limit to 101

```
1 • select * from allusers;
2
```

Result Grid

	id	acName	acEmail	balance
▶	1	Akash	akashmete129@gmail.com	30000
	2	Harsh	harsh123@gmail.com	91000
	3	Kaustubh	kaustubh123@gmail.com	99000
	4	Orion	orionoriginal@gmail.com	100000
	5	Siddhi	siddhi123@gmail.com	95000
	6	Vaish	vaish123@gmail.com	120000
	7	Sanjay	sanjay232@gmail.com	85000
	8	Karan	karan333@gmail.com	60000
	9	Arjun	arjun243@gmail.com	65000
	23	Ankit Patil	ankitpatil123@gmail.com	1000
*	NULL	NULL	NULL	NULL

createnewuser.php - creating new account for new user

```
<!-- Create new user -->
<?php
    include 'connfile.php';

    $name = $_REQUEST['name'];
    $email = $_REQUEST['email'];
    $balance = $_REQUEST['balance'];

    $sql = "INSERT INTO allusers VALUES ('','$name','$email','$balance')";

    if (mysqli_query($conn, $sql)) {

    } else {
        echo '<script type="text/javascript">';
        echo ' alert("Oops! Failed to register your account. Try again
later.");' // showing an alert box.
        echo '</script>';
    }
?>
```

Create New User Account

New User? Get registered here

Name
Ankit

Email address
ankit123@gmail.com

Balance
1000

☒ Information provided is TRUE

Submit

1 • select * from allusers;
2

Limit to 10

Result Grid | Filter Rows: | Edit:

	id	acName	acEmail	balance
▶	1	Akash	akashmete129@gmail.com	30000
	2	Harsh	harsh123@gmail.com	91000
	3	Kaustubh	kaustubh123@gmail.com	99000
	4	Orion	orionoriginal@gmail.com	100000
	5	Siddhi	siddhi123@gmail.com	95000
	6	Vaish	vaish123@gmail.com	120000
	7	Sanjay	sanjay232@gmail.com	85000
	8	Karan	karan333@gmail.com	60000
	9	Arjun	arjun243@gmail.com	65000
	23	Ankit Patil	ankitpatil123@gmail.com	1000
*	NULL	NULL	NULL	NULL

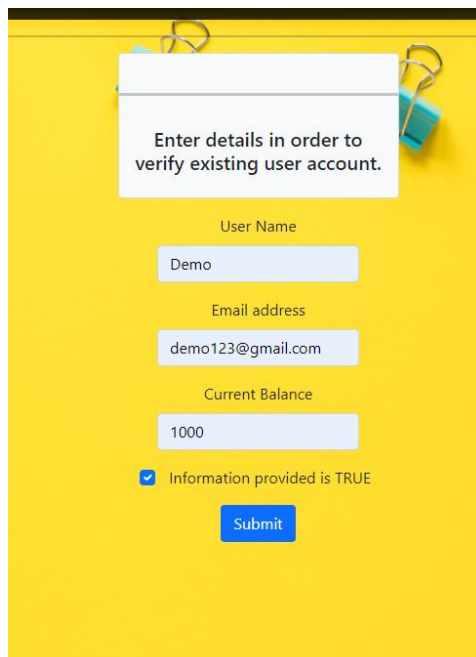
deleteuser.php – delete user account

```
<!-- Delete user -->
<?php
    include 'connfile.php';
    $name = $_REQUEST['name'];
    $email = $_REQUEST['email'];
    $balance = $_REQUEST['balance'];

    $sql = "DELETE FROM allusers WHERE acName='$name'";

    if (mysqli_query($conn, $sql)) {

    } else {
        echo '<script type="text/javascript">';
        echo ' alert("Oops! Failed to update your account. Try again
later.")'; // showing an alert box.
        echo '</script>';
    }
?>
```



Enter details in order to verify existing user account.

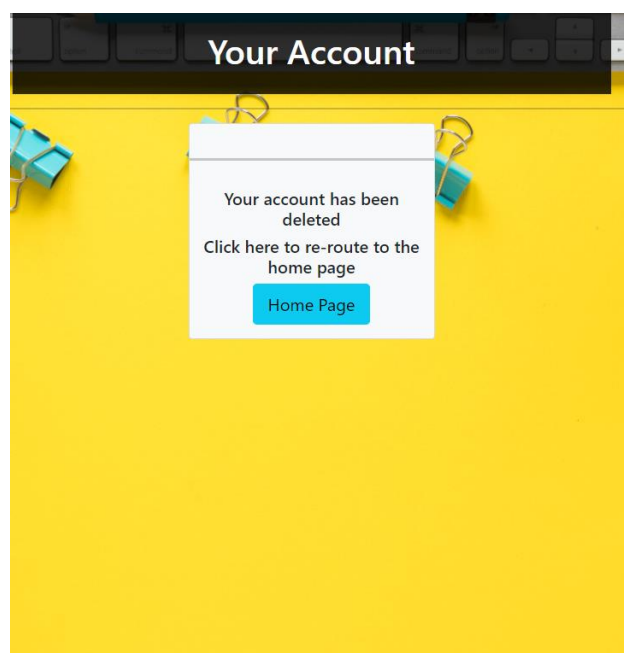
User Name
Demo

Email address
demo123@gmail.com

Current Balance
1000

☒ Information provided is TRUE

Submit



updateuser.php – Update existing user details

```

<!-- Update user -->
<?php
    include 'connfile.php';
    $oldname = $_REQUEST['oldname'];
    $newname = $_REQUEST['newname'];
    $email = $_REQUEST['email'];

    $sql = "UPDATE allusers SET acName = '$newname',acEmail = '$email' WHERE
acName='$oldname'; ";

    if (mysqli_query($conn, $sql)) {

    } else {
        echo '<script type="text/javascript">';
        echo ' alert("Oops! Failed to update your account. Try again
later.");' // showing an alert box.
        echo '</script>';
    }
?>

```

Enter previous user name.
Then re-enter all details to
update.

Old User Name

Ankit

New User Name

Ankit Patil

New Email address

ankitpatil123@gmail.com

☒ Information provided is TRUE

Submit

Your Account

Id	Name	E-Mail	Balance
23	Ankit Patil	ankitpatil123@gmail.com	1000

You can review your
account details here

allusers table data:

The screenshot shows the phpMyAdmin interface for the 'bankingdata' database. The 'allusers' table is selected, and the 'Structure' tab is active. The table has 10 rows of data. The columns are 'id', 'acName', 'acEmail', and 'balance'. The data is as follows:

id	acName	acEmail	balance
1	Akash	akashmete129@gmail.com	39000
2	Harsh	harsh123@gmail.com	82000
3	Kaustubh	kaustubh123@gmail.com	99000
4	Orion	orionoriginal@gmail.com	100000
5	Siddhi	siddhi123@gmail.com	95000
6	Vaish	vaish123@gmail.com	120000
7	Sanjay	sanjay232@gmail.com	85000
8	Karan	karan333@gmail.com	60000
9	Arjun	arjun243@gmail.com	65000
23	Ankit Patil	ankitpatil123@gmail.com	1000

transaction table data:

The screenshot shows the phpMyAdmin interface for the 'bankingdata' database. The 'transaction' table is selected, and the 'Structure' tab is active. The table has 1 row of data. The columns are 'sender', 'receiver', and 'balance'. The data is as follows:

sender	receiver	balance
Akash	Kaustubh	9000

WT Assignment/Practical - 08

Name: - Kaustubh Shrikant Kabra

Class: - Third Year Engineering(Computer) Div.: - A

ERP No.: - 38

Code

index.html

```
<!DOCTYPE html>
<!-- Coding By CodingNepal - youtube.com/codingnepal -->
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Login Form with Shake Effect | CodingNepal</title>
    <link rel="stylesheet" href="style.css" />
    <link
      rel="stylesheet"
      href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.3/css/all.min.css"
    />
  </head>
  <body>
    <div class="wrapper">
      <header>Login Form</header>
      <form action="#">
        <div class="field name">
          <div class="input-area">
            <input type="text" placeholder="User Name" />
            <i class="icon fas fa-user"></i>
            <i class="error error-icon fas fa-exclamation-circle"></i>
          </div>
          <div class="error error-txt">Name can't be blank</div>
          <br />
        </div>

        <div class="field number">
          <div class="input-area">
            <input type="tel" placeholder="Mobile Number" />
            <i class="icon fas fa-envelope"></i>
            <i class="error error-icon fas fa-exclamation-circle"></i>
          </div>
          <div class="error error-txt">Invalid Mobile Number</div>
          <br />
        </div>

        <div class="field email">
          <div class="input-area">
            <input type="text" placeholder="Email Address" />
            <i class="icon fas fa-envelope"></i>
            <i class="error error-icon fas fa-exclamation-circle"></i>
          </div>
          <div class="error error-txt">Email can't be blank</div>
        </div>

        <div class="field password">
          <div class="input-area">
            <input type="password" placeholder="Password" />
            <i class="icon fas fa-lock"></i>
            <i class="error error-icon fas fa-exclamation-circle"></i>
          </div>
        </div>
      </form>
    </div>
  </body>
</html>
```

```

        <div class="error error-txt">Password can't be blank</div>
    </div>

    <div class="pass-txt"><a href="#">Forgot password?</a></div>
    <input type="submit" value="Login" />
</form>
<div class="sign-txt">Not yet member? <a href="#">Signup now</a></div>
</div>

<script src="script.js"></script>
</body>
</html>

```

script.js

```

const form = document.querySelector("form");
nameField = form.querySelector(".name"),
eName = nameField.querySelector("input"),
numberField = form.querySelector(".number"),
eNumber = numberField.querySelector("input")
eField = form.querySelector(".email"),
eInput = eField.querySelector("input"),
pField = form.querySelector(".password"),
pInput = pField.querySelector("input");

form.onsubmit = (e)=>{
    e.preventDefault(); //preventing from form submitting
    //if email and password is blank then add shake class in it else call specified function
    (eName.value == "") ? nameField.classList.add("shake", "error") : checkName();
    (eNumber.value == "") ? numberField.classList.add("shake", "error") : checkNumber();
    (eInput.value == "") ? eField.classList.add("shake", "error") : checkEmail();
    (pInput.value == "") ? pField.classList.add("shake", "error") : checkPass();

    setTimeout(()=>{ //remove shake class after 500ms
        eField.classList.remove("shake");
        pField.classList.remove("shake");
    }, 500);

    eName.onkeyup = () => {checkName();}
    eNumber.onkeyup = () => {checkNumber();}
    eInput.onkeyup = ()=>{checkEmail();} //calling checkEmail function on email input keyup
    pInput.onkeyup = ()=>{checkPass();} //calling checkPassword function on pass input keyup

function checkName(){ //checkEmail function
    let pattern = /^[a-zA-Z\-\-]+$/; //pattern for validate email
    if(!eName.value.match(pattern)){ //if pattern not matched then add error and remove
valid class
        nameField.classList.add("error");
        nameField.classList.remove("valid");
        let errorTxt = nameField.querySelector(".error-txt");
        //if email value is not empty then show please enter valid email else show Email can't
be blank
        (eName.value != "") ? errorTxt.innerText = "Enter valid User Name" :
errorTxt.innerText = "Name can't be blank";
    }else{ //if pattern matched then remove error and add valid class
        nameField.classList.remove("error");
        nameField.classList.add("valid");
    }
}
}

```

```

function checkNumber(){ //checkEmail function
  let pattern = /^\(?\d{3}\)?[- ]?\d{3}[- ]?\d{4}$/; //pattern for validate email
  if(!eNumber.value.match(pattern)){ //if pattern not matched then add error and remove
valid class
    numberField.classList.add("error");
    numberField.classList.remove("valid");
    let errorTxt = numberField.querySelector(".error-txt");
    //if email value is not empty then show please enter valid email else show Email can't
be blank
    (eNumber.value != "") ? errorTxt.innerText = "Enter a valid number" :
errorTxt.innerText = "Invalid mobile number";
  }else{ //if pattern matched then remove error and add valid class
    numberField.classList.remove("error");
    numberField.classList.add("valid");
  }
}

function checkEmail(){ //checkEmail function
  let pattern = /^[^ ]+@[^ ]+\.[a-z]{2,3}$/; //pattern for validate email
  if(!eInput.value.match(pattern)){ //if pattern not matched then add error and remove
valid class
    eField.classList.add("error");
    eField.classList.remove("valid");
    let errorTxt = eField.querySelector(".error-txt");
    //if email value is not empty then show please enter valid email else show Email
can't be blank
    (eInput.value != "") ? errorTxt.innerText = "Enter a valid email address" :
errorTxt.innerText = "Email can't be blank";
  }else{ //if pattern matched then remove error and add valid class
    eField.classList.remove("error");
    eField.classList.add("valid");
  }
}

function checkPass(){ //checkPass function
  if(pInput.value == ""){ //if pass is empty then add error and remove valid class
    pField.classList.add("error");
    pField.classList.remove("valid");
  }else{ //if pass is empty then remove error and add valid class
    pField.classList.remove("error");
    pField.classList.add("valid");
  }
}

//if eField and pField doesn't contains error class that mean user filled details
properly
if(!eField.classList.contains("error") && !pField.classList.contains("error")){
  window.location.href = form.getAttribute("action"); //redirecting user to the
specified url which is inside action attribute of form tag
}
}

```

style.css

```

@import
url('https://fonts.googleapis.com/css2?family=Poppins:wght@400;500;600&display=swap');
*{
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}

```

```

    font-family: "Poppins", sans-serif;
}
body{
    width: 100%;
    height: 100vh;
    display: flex;
    align-items: center;
    justify-content: center;
    background: #5372F0;
}
::selection{
    color: #fff;
    background: #5372F0;
}
.wrapper{
    width: 380px;
    padding: 40px 30px 50px 30px;
    background: #fff;
    border-radius: 5px;
    text-align: center;
    box-shadow: 10px 10px 15px rgba(0,0,0,0.1);
}
.wrapper header{
    font-size: 35px;
    font-weight: 600;
}
.wrapper form{
    margin: 40px 0;
}
form .field{
    width: 100%;
    margin-bottom: 20px;
}
form .field.shake{
    animation: shake 0.3s ease-in-out;
}
@keyframes shake {
    0%, 100%{
        margin-left: 0px;
    }
    20%, 80%{
        margin-left: -12px;
    }
    40%, 60%{
        margin-left: 12px;
    }
}
form .field .input-area{
    height: 50px;
    width: 100%;
    position: relative;
}
form input{
    width: 100%;
    height: 100%;
    outline: none;
    padding: 0 45px;
    font-size: 18px;
    background: none;
    caret-color: #5372F0;
    border-radius: 5px;
    border: 1px solid #bfbfbf;
}


```



```
border-bottom-width: 2px;
transition: all 0.2s ease;
}
form .field input:focus,
form .field.valid input{
border-color: #5372F0;
}
form .field.shake input,
form .field.error input{
border-color: #dc3545;
}
.field .input-area i{
position: absolute;
top: 50%;
font-size: 18px;
pointer-events: none;
transform: translateY(-50%);
}
.input-area .icon{
left: 15px;
color: #bfbfbf;
transition: color 0.2s ease;
}
.input-area .error-icon{
right: 15px;
color: #dc3545;
}
form input:focus ~ .icon,
form .field.valid .icon{
color: #5372F0;
}
form .field.shake input:focus ~ .icon,
form .field.error input:focus ~ .icon{
color: #bfbfbf;
}
form input::placeholder{
color: #bfbfbf;
font-size: 17px;
}
form .field .error-txt{
color: #dc3545;
text-align: left;
margin-top: 5px;
}
form .field .error{
display: none;
}
form .field.shake .error,
form .field.error .error{
display: block;
}
form .pass-txt{
text-align: left;
margin-top: -10px;
}
.wrapper a{
color: #5372F0;
text-decoration: none;
}
.wrapper a:hover{
text-decoration: underline;
}
```


```
form input[type="submit"]{
  height: 50px;
  margin-top: 30px;
  color: #fff;
  padding: 0;
  border: none;
  background: #5372F0;
  cursor: pointer;
  border-bottom: 2px solid rgba(0,0,0,0.1);
  transition: all 0.3s ease;
}
```

Login Form

 User Name

 Mobile Number

 Email Address


 Password


[Forgot password?](#)


Login


Not yet member? [Signup now](#)

Login Form

 Demo

 1997532155

 demo123@gmail.com



[Forgot password?](#)

Login

Not yet member? [Signup now](#)





Welcome to our Webpage
You have been successfully logged in.



Login Form

 Demo123 


Enter valid User Name

 199753215523r5 

Enter a valid number

 demo123@gmail.com24r5 

Enter a valid email address



[Forgot password?](#)

Login

Not yet member? [Signup now](#)

WT Assignment/Practicals - 09

Name: - Kaustubh Shrikant Kabra

Class: - Third Year Engineering(Computer)

Div.: - A **ERP No.:** - 38

Code

```
<!DOCTYPE html>
<html ng-app="app">
<head>
  <meta charset="utf-8" />
  <title>AngularJS User Registration and Login Example</title>
  <link rel="stylesheet"
href="//netdna.bootstrapcdn.com/bootstrap/3.1.1/css/bootstrap.min.css" />
  <link href="app-content/app.css" rel="stylesheet" />
</head>
<body>
  <div class="jumbotron">
    <div class="container">
      <div class="col-sm-8 col-sm-offset-2">
        <div ng-class="{ 'alert': flash, 'alert-success': flash.type ===
'success', 'alert-danger': flash.type === 'error' }" ng-if="flash" ng-
bind="flash.message"></div>
        <div ng-view></div>
      </div>
    </div>
  </div>
  <div class="credits text-center">
    <p>
      <h2>AngularJS User Registration and Login Example</h2>
    </p>
  </div>

  <script src="//code.jquery.com/jquery-3.1.1.min.js"></script>
  <script src="//code.angularjs.org/1.6.0/angular.min.js"></script>
  <script src="//code.angularjs.org/1.6.0/angular-route.min.js"></script>
  <script src="//code.angularjs.org/1.6.0/angular-cookies.min.js"></script>

  <script src="app.js"></script>
  <script src="app-services/authentication.service.js"></script>
  <script src="app-services/flash.service.js"></script>
  <script src="app-services/user.service.local-storage.js"></script>

  <script src="home/home.controller.js"></script>
  <script src="login/login.controller.js"></script>
  <script src="register/register.controller.js"></script>
</body>
</html>
```

App.js

```
(function () {
  'use strict';
  angular
    .module('app', ['ngRoute', 'ngCookies'])
    .config(config)
    .run(run);

  config.$inject = ['$routeProvider', '$locationProvider'];
  function config($routeProvider, $locationProvider) {
    $routeProvider
      .when('/', {
        controller: 'HomeController',
        templateUrl: 'home/home.view.html',
        controllerAs: 'vm'
      })

      .when('/login', {
        controller: 'LoginController',
        templateUrl: 'login/login.view.html',
        controllerAs: 'vm'
      })

      .when('/register', {
        controller: 'RegisterController',
        templateUrl: 'register/register.view.html',
        controllerAs: 'vm'
      })
      .otherwise({ redirectTo: '/login' });
  }

  run.$inject = ['$rootScope', '$location', '$cookies', '$http'];
  function run($rootScope, $location, $cookies, $http) {
    // keep user logged in after page refresh
    $rootScope.globals = $cookies.getObject('globals') || {};
    if ($rootScope.globals.currentUser) {
      $http.defaults.headers.common['Authorization'] = 'Basic ' +
$rootScope.globals.currentUser.authdata;
    }

    $rootScope.$on('$locationChangeStart', function (event, next, current) {
      // redirect to login page if not logged in and trying to access a restricted
page
      var restrictedPage = $.inArray($location.path(), ['/login', '/register']) ===
-1;

      var loggedIn = $rootScope.globals.currentUser;
      if (restrictedPage && !loggedIn) {
        $location.path('/login');
      }
    });
  }
})();
```

/home/home.view.html

```

<h1>Hi {{vm.user.firstName}}!</h1>
<p>You're logged in!!</p>
<h3>All registered users:</h3>
<ul>
  <li ng-repeat="user in vm.allUsers">
    {{user.username}} ({{user.firstName}} {{user.lastName}})
    - <a ng-click="vm.deleteUser(user.id)">Delete</a>
  </li>
</ul>
<p>&nbsp;</p>
<p><a href="#!/login" class="btn btn-primary">Logout</a></p>

```

/home/home.controller.js

```

(function () {
  'use strict';
  angular
    .module('app')
    .controller('HomeController', HomeController);
  HomeController.$inject = ['UserService', '$rootScope'];
  function HomeController(UserService, $rootScope) {
    var vm = this;
    vm.user = null;
    vm.allUsers = [];
    vm.deleteUser = deleteUser;
    initController();
    function initController() {
      loadCurrentUser();
      loadAllUsers();
    }
    function loadCurrentUser() {
      UserService.GetByUsername($rootScope.globals.currentUser.username)
        .then(function (user) {
          vm.user = user;
        });
    }
    function loadAllUsers() {
      UserService.GetAll()
        .then(function (users) {
          vm.allUsers = users;
        });
    }
    function deleteUser(id) {
      UserService.Delete(id)
        .then(function () {
          loadAllUsers();
        });
    }
  }
})();

```

/login/login.view.html

```

<div class="col-md-6 col-md-offset-3">
  <h2>Login</h2>
  <form name="form" ng-submit="vm.login()" role="form">
    <div class="form-group" ng-class="{ 'has-error': form.username.$dirty &&
form.username.$error.required }">

```

```

        <label for="username">Username</label>
        <input type="text" name="username" id="username" class="form-control" ng-
model="vm.username" required />
        <span ng-show="form.username.$dirty && form.username.$error.required"
class="help-block">Username is required</span>
    </div>
    <div class="form-group" ng-class="{ 'has-error': form.password.$dirty &&
form.password.$error.required }">
        <label for="password">Password</label>
        <input type="password" name="password" id="password" class="form-control" ng-
model="vm.password" required />
        <span ng-show="form.password.$dirty && form.password.$error.required"
class="help-block">Password is required</span>
    </div>
    <div class="form-actions">
        <button type="submit" ng-disabled="form.$invalid || vm.dataLoading" class="btn
btn-primary">Login</button>
        
        <a href="#!/register" class="btn btn-link">Register</a>
    </div>
</form>
</div>

```

/login/login.controller.js

```

(function () {
    'use strict';

    angular
        .module('app')
        .controller('LoginController', LoginController);

    LoginController.$inject = ['$location', 'AuthenticationService', 'FlashService'];
    function LoginController($location, AuthenticationService, FlashService) {
        var vm = this;

        vm.login = login;

        (function initController() {
            // reset login status
            AuthenticationService.ClearCredentials();
        })();

        function login() {
            vm.dataLoading = true;
            AuthenticationService.Login(vm.username, vm.password, function (response) {
                if (response.success) {
                    AuthenticationService.SetCredentials(vm.username, vm.password);
                    $location.path('/');
                } else {
                    FlashService.Error(response.message);
                    vm.dataLoading = false;
                }
            });
        };
    };
}

```

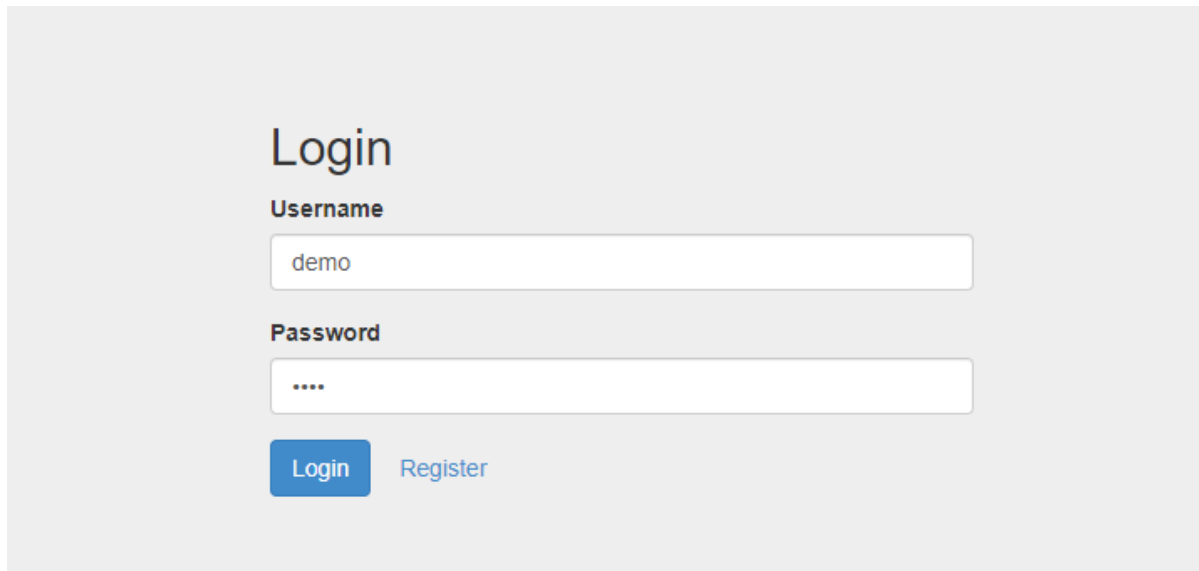


```
})();
```

/register/register.view.html

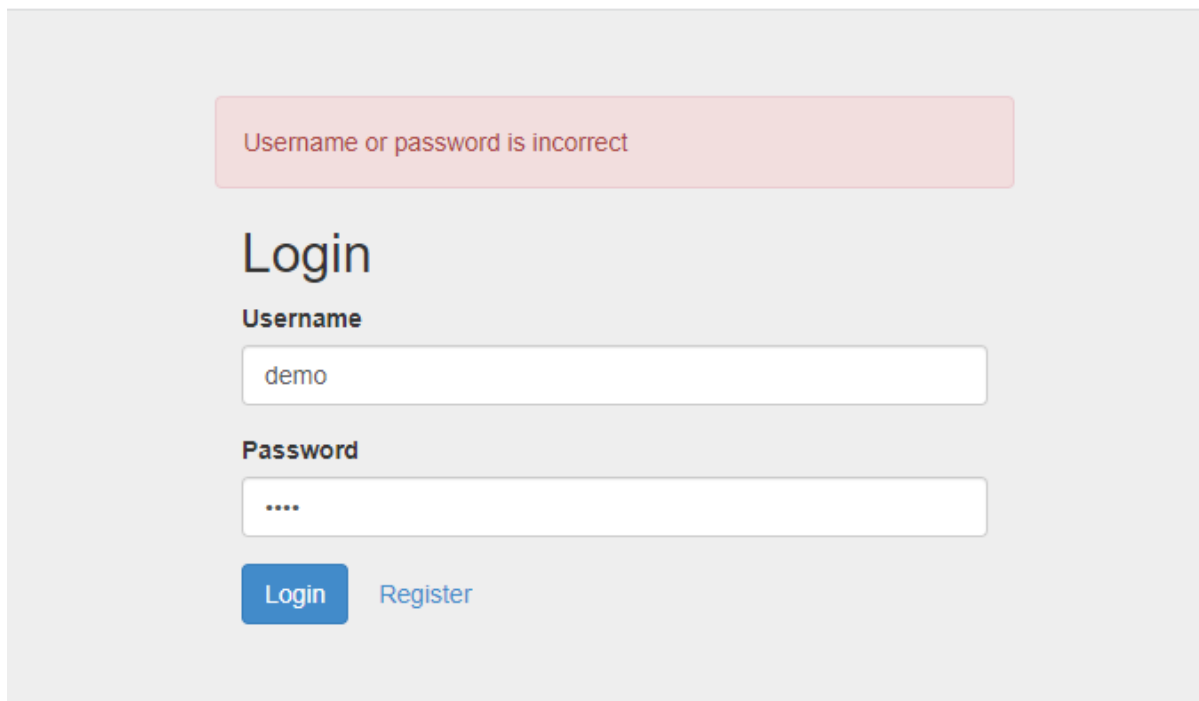
```
<div class="col-md-6 col-md-offset-3">
  <h2>Register</h2>
  <form name="form" ng-submit="vm.register()" role="form">
    <div class="form-group" ng-class="{ 'has-error': form.firstName.$dirty &&
form.firstName.$error.required }">
      <label for="username">First name</label>
      <input type="text" name="firstName" id="firstName" class="form-control" ng-
model="vm.user.firstName" required />
      <span ng-show="form.firstName.$dirty && form.firstName.$error.required"
class="help-block">First name is required</span>
    </div>
    <div class="form-group" ng-class="{ 'has-error': form.lastName.$dirty &&
form.lastName.$error.required }">
      <label for="username">Last name</label>
      <input type="text" name="lastName" id="Text1" class="form-control" ng-
model="vm.user.lastName" required />
      <span ng-show="form.lastName.$dirty && form.lastName.$error.required"
class="help-block">Last name is required</span>
    </div>
    <div class="form-group" ng-class="{ 'has-error': form.username.$dirty &&
form.username.$error.required }">
      <label for="username">Username</label>
      <input type="text" name="username" id="username" class="form-control" ng-
model="vm.user.username" required />
      <span ng-show="form.username.$dirty && form.username.$error.required"
class="help-block">Username is required</span>
    </div>
    <div class="form-group" ng-class="{ 'has-error': form.password.$dirty &&
form.password.$error.required }">
      <label for="password">Password</label>
      <input type="password" name="password" id="password" class="form-control" ng-
model="vm.user.password" required />
      <span ng-show="form.password.$dirty && form.password.$error.required"
class="help-block">Password is required</span>
    </div>
    <div class="form-actions">
      <button type="submit" ng-disabled="form.$invalid || vm.dataLoading" class="btn
btn-primary">Register</button>
      
      <a href="#!/login" class="btn btn-link">Cancel</a>
    </div>
  </form></div>
```

Output :



A screenshot of a login form on a light gray background. The form is titled "Login" in a large, dark font. Below the title, there are two labels: "Username" and "Password". The "Username" label is followed by a text input field containing the text "demo". The "Password" label is followed by a password input field with four dots. Below the input fields, there are two buttons: a blue "Login" button and a blue "Register" button.

AngularJS User Registration and Login Example



A screenshot of the same login form as above, but with an error message displayed. The error message is "Username or password is incorrect" and is shown in a red box at the top of the form. The form itself is identical to the previous one, with the "Username" field containing "demo" and the "Password" field containing four dots. The "Login" and "Register" buttons are still present at the bottom.

AngularJS User Registration and Login Example

Register

First name

Last name

Username

Password

[Cancel](#)

AngularJS User Registration and Login Example

Registration successful

Login

Username

Password

[Register](#)

AngularJS User Registration and Login Example

Hi demo!

You're logged in!!

All registered users:

- demo (demo demo) - [Delete](#)

Logout

AngularJS User Registration and Login Example

WT Assignment/Practicals - 10

Name: - Kaustubh Shrikant Kabra

Class: - Third Year Engineering(Computer)

Div.: - A ERP No.: - 38

Index.html

```
<html>
<head>
  <title>Web Application For Bank Transactions</title>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<body>
  <form action="Main">
    Enter Amount: <input type="text" name="amt_txt">
    <br/> <br/>
    Select Operation:
    <br/> <br/>
    <select name="operation">
      <option value="deposit">Deposit</option>
      <option value="withdraw">Withdraw</option>
      <option value="balance">Balance</option>
    </select>
    <br/> <br/>
    <input type="Submit" value="Submit">
  </form>
</body>
</html>
```

MyBean.java

```
package myPkg;
import javax.ejb.Stateful;
@Stateful
public class MyBean implements MyBeanLocal {
  int balance=0;
  @Override
  public void deposit(int amount) {
    balance=balance+amount;
  }
  @Override
  public boolean withdraw(int amount) {
    if(amount <= balance)
    {
      balance=balance-amount;
      return true;
    }
    return false;
  }
  @Override
  public int getBalance() {
    return balance;
  }
}
```

Main.java

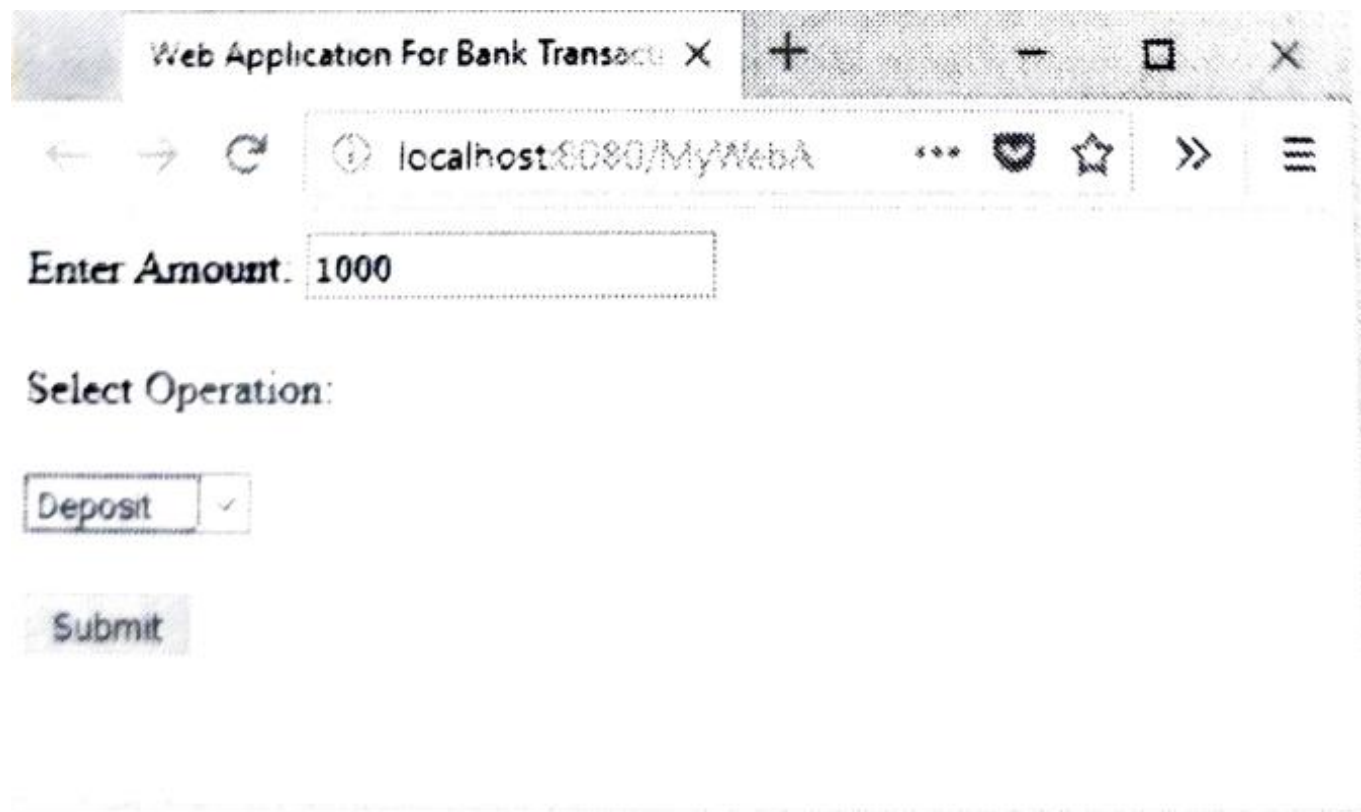
```
import java.io.IOException;
import java.io.PrintWriter;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.naming.Context;
import javax.naming.InitialContext;
import javax.naming.NamingException;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import myPkg.MyBeanLocal;

public class Main extends HttpServlet {
    MyBeanLocal myBean = lookupMyBeanLocal();
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            String choice = request.getParameter("operation");
            int amount = Integer.parseInt(request.getParameter("amt_txt"));
            if(choice.equals("deposit"))
            {
                myBean.deposit(amount);
                out.println("The amount is deposited successfully!!!");
            }
            if(choice.equals("withdraw"))
            {
                if(myBean.withdraw(amount))
                {
                    out.println("The amount is withdrawn successfully!!!");
                }
                Else
                    out.println("Sorry, Balance insufficient!!!");
            }
            if(choice.equals("balance"))
            {
                out.println("The total balance is = " + myBean.getBalance());
            }
        }
    }

    ...

    private MyBeanLocal lookupMyBeanLocal() {
        try {
            Context c = new InitialContext();
            return (MyBeanLocal) c.lookup("java:global/MyWebApp/MyWebApp-
ejb/MyBean!myPkg.MyBeanLocal");
        } catch (NamingException ne) {
            Logger.getLogger(getClass().getName()).log(Level.SEVERE, "exception caught", ne);
            throw new RuntimeException(ne);
        }
    }
}
```

Output



A screenshot of a web browser window titled "Web Application For Bank Transac...". The address bar shows "localhost:8080/MyWebA". The page content includes a label "Enter Amount:" followed by a text input field containing "1000". Below this is a label "Select Operation:" followed by a dropdown menu with "Deposit" selected and a checkmark icon. At the bottom is a "Submit" button.

Web Application For Bank Transac... X + - □ ×

← → ↻ ⓘ localhost:8080/MyWebA ... 🛡️ ☆ >> ≡

Enter Amount: 1000

Select Operation:

Deposit ✓

Submit

