

# Project Development Phase

## Delivery of Sprint 1

### Front End:

### HTML:

### Login.html:

```
{% extends 'base.html' %}

{% block head %}

<title>Login page</title>

{% endblock %}

{% block body %}

<main class="container">

    <div class="mx-auto mt-5 border bg-light login-card" style="width:500px;">

        <h2 class="mx-4 mt-2">LOGIN</h2>

        <form action="{{url_for('login')}}" method="post">

            <div class="mx-4 mt-2 text-danger">{{ msg }}</div>

            <div class="my-2 mx-4">

                <label for="username">username</label>

                <input type="text" class="form-control" placeholder="adc@gmail.com" name="username" required />

            </div>

            <div class="my-2 mx-4">

                <label for="password_1">password</label>

                <input type="password" class="form-control" name="password_1" required />

            </div>

            <input type="submit" value="submit" class="btn btn-primary my-4 mt-2 mx-4" />

        </form>

        <p>Don't have an account?<a href="{{url_for('signup')}}"> Sign Up</a>

    </div>

</main>

</p>

</main>

{% endblock %}
```

### Signup.html:

```
{% extends 'base.html' %} {% block head %}
```

```

<title>Signup page</title>

{% endblock%} {%block body%}

<main class="container">

  <div class="mx-auto mt-5 border bg-light" style="width: 500px">

    <h2 class="mx-4 mt-2">REGISTER FORM</h2>

    <div class="mx-4 mt-2 text-danger">{{ meg }}</div>

    <form action="{{url_for('signup')}}" method="post">

      <div class="my-2 mx-4">

        <label for="username">User name</label>

        <input type="text" class="form-control" placeholder="Ram" name="username" required />

      </div>

      <div class="my-2 mx-4">

        <label for="email">email</label>

        <input type="text" class="form-control" placeholder="abc@gmail.com" name="email" required />

      </div>

      <div class="my-2 mx-4">

        <label for="password">password</label>

        <input type="password" class="form-control" placeholder="password" name="password" required />

      </div>

      <div class="my-2 mx-4">

        <label for="retype">retype password</label>

        <input type="password" class="form-control" placeholder="password" name="retype" required />

      </div>

      <input type="submit" value="submit" class="btn btn-primary my-4 my-2 mt-2 mx-4" />

      <div class="note mt-3 text-center">

        <!--Register form -->

        <p>already have an account ? please <a href="/login"> login! </a></p>

      </div>

    </form>

  </div>

</main>

{% endblock%}

```

## CSS:

```

/*

DEMO STYLE

*/

```

```
@import 'https://fonts.googleapis.com/css?family=Poppins:300,400,500,600,700';
```

```
* {
```

```
  box-sizing: border-box;
```

```
  -webkit-box-sizing: border-box;
```

```
  -moz-box-sizing: border-box;
```

```
}
```

```
body {
```

```
  font-family: Helvetica;
```

```
  -webkit-font-smoothing: antialiased;
```

```
  background: rgba(71, 147, 227, 1);
```

```
}
```

```
/* h2 {
```

```
  text-align: center;
```

```
  font-size: 18px;
```

```
  text-transform: uppercase;
```

```
  letter-spacing: 1px;
```

```
  color: white;
```

```
  padding: 30px 0;
```

```
} */
```

```
/* Table Styles */
```

```
.table-wrapper {
```

```
  margin: 10px 70px 70px;
```

```
  box-shadow: 0px 35px 50px rgba(0, 0, 0, 0.2);
```

```
}
```

```
.fl-table {
```

```
  border-radius: 5px;
```

```
  font-size: 12px;
```

```
  font-weight: normal;
```

```
  border: none;
```

```
  border-collapse: collapse;
```

```
  width: 100%;
```

```
  max-width: 100%;
```

```
  white-space: nowrap;
```

```
  background-color: white;
```

```
}
```

```
.fl-table td,  
.fl-table th {  
  text-align: center;  
  padding: 8px;  
}
```

```
.fl-table td {  
  border-right: 1px solid #f8f8f8;  
  font-size: 12px;  
}
```

```
.fl-table thead th {  
  color: #ffffff;  
  background: #4fc3a1;  
}
```

```
.fl-table thead th:nth-child(odd) {  
  color: #ffffff;  
  background: #324960;  
}
```

```
.fl-table tr:nth-child(even) {  
  background: #f8f8f8;  
}
```

```
/* Responsive */
```

```
@media (max-width: 767px) {  
  .fl-table {  
    display: block;  
    width: 100%;  
  }  
  .table-wrapper:before {  
    content: 'Scroll horizontally >';  
    display: block;  
    text-align: right;
```

```
font-size: 11px;

color: white;

padding: 0 0 10px;

}

.fl-table thead,

.fl-table tbody,

.fl-table thead th {

    display: block;

}

.fl-table thead th:last-child {

    border-bottom: none;

}

.fl-table thead {

    float: left;

}

.fl-table tbody {

    width: auto;

    position: relative;

    overflow-x: auto;

}

.fl-table td,

.fl-table th {

    padding: 20px 0.625em 0.625em 0.625em;

    height: 60px;

    vertical-align: middle;

    box-sizing: border-box;

    overflow-x: hidden;

    overflow-y: auto;

    width: 120px;

    font-size: 13px;

    text-overflow: ellipsis;

}

.fl-table thead th {

    text-align: left;

    border-bottom: 1px solid #f7f7f9;

}

.fl-table tbody tr {

    display: table-cell;
```

```

}

.fl-table tbody tr:nth-child(odd) {

    background: none;

}

.fl-table tr:nth-child(even) {

    background: transparent;

}

.fl-table tr td:nth-child(odd) {

    background: #f8f8f8;

    border-right: 1px solid #e6e4e4;

}

.fl-table tr td:nth-child(even) {

    border-right: 1px solid #e6e4e4;

}

.fl-table tbody td {

    display: block;

    text-align: center;

}

}

body {

    font-family: 'Poppins', sans-serif;

    background: #fafafa;

}

p {

    font-family: 'Poppins', sans-serif;

    font-size: 1.1em;

    font-weight: 300;

    line-height: 1.7em;

    color: #999;

}

a,

a:hover,

a:focus {

    color: inherit;

    text-decoration: none;

    transition: all 0.3s;

}

.navbar {

```

```
padding: 15px 10px;

background: #fff;

border: none;

border-radius: 0;

margin-bottom: 40px;

box-shadow: 1px 1px 3px rgba(0, 0, 0, 0.1);
}

.navbar-btn {

box-shadow: none;

outline: none !important;

border: none;
}

.line {

width: 100%;

height: 1px;

border-bottom: 1px dashed #ddd;

margin: 40px 0;
}

.wrapper {

display: flex;

width: 100%;

align-items: stretch;
}

#sidebar {

min-width: 250px;

max-width: 250px;

background: #48494b;

color: #fff;

transition: all 0.3s;
}

#sidebar.active {

margin-left: -250px;
}

#sidebar .sidebar-header {

padding: 20px;

background: #48494b;
}

#sidebar ul.components {
```

```
padding: 20px 0;

border-bottom: 1px solid #47748b;
}

#sidebar ul p {

color: #fff;

padding: 10px;

}

.project-title {

font-size: 20px;

padding-left: 10px;

text-align: center;

}

#sidebar ul li a {

padding: 10px;

font-size: 1.1em;

display: block;

}

#sidebar ul li a:hover {

color: #7386d5;

background: #fff;

}

#sidebar ul li.active > a,

a[aria-expanded='true'] {

color: #fff;

background: #48494b;

}

a[data-toggle='collapse'] {

position: relative;

}

.dropdown-toggle::after {

display: block;

position: absolute;

top: 50%;

right: 20px;

transform: translateY(-50%);

}

ul ul a {

font-size: 0.9em !important;
```



```
padding-left: 30px !important;

background: #48494b;
}

ul.CTAs {
padding: 20px;
}

ul.CTAs a {
text-align: center;

font-size: 0.9em !important;

display: block;

border-radius: 5px;

margin-bottom: 5px;
}

a.download {
background: #fff;

color: #48494b;
}

a.article,
a.article:hover {
background: #48494b !important;

color: #fff !important;
}

.login-card {
box-shadow: rgba(0, 0, 0, 0.35) 0px 5px 15px;

border-radius: 10px;

padding: 10px;
}

.login-card p {
padding-left: 20px;
}

.login-card a {
color: rgba(84, 84, 220, 0.888);
}

#content {
width: 100%;

padding: 20px;

min-height: 100vh;

transition: all 0.3s;
```

```

@media (max-width: 768px) {

  #sidebar {

    margin-left: -250px;

  }

  #sidebar.active {

    margin-left: 0;

  }

  #sidebarCollapse span {

    display: none;

  }

}

```

## JavaScript:

```

const selectedItem = document

.getElementById('sidebarCollapse')

.addEventListener('click', (e) => {

  const ele = document.getElementById('sidebar');

  ele.classList.toggle('active');

});

```

## Back-end:

### app.py:

```

from flask import Flask, render_template, url_for, request, redirect, session, make_response

import sqlite3 as sql

from functools import wraps

import re

import ibm_db

conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=21fecfd8-47b7-4937-840d-d791d0218660.bs2io90l08kqb1od8lcg.databases.ap
pdomain.cloud
;PORT=31864;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=gkx49901;PWD=kvWCsySI7vApsy2", "", "")

app = Flask(__name__)

app.secret_key = 'kit'

def rewrite(url):

    view_func, view_args = app.create_url_adapter(request).match(url)

    return app.view_functions[view_func](**view_args)

def login_required(f):

    @wraps(f)

    def decorated_function(*args, **kwargs):

```

```

        if "id" not in session:

            return redirect(url_for('login'))

        return f(*args, **kwargs)

    return decorated_function

@app.route('/')

def root():

    return render_template('login.html')

@app.route('/user/<id>')

@login_required

def user_info(id):

    with sql.connect('inventorymanagement.db') as con:

        con.row_factory = sql.Row

        cur = con.cursor()

        cur.execute(f'SELECT * FROM register WHERE email="{id}"')

        user = cur.fetchall()

    return render_template("user_info.html", user=user[0])

@app.route('/login', methods=['GET', 'POST'])

def login():

    global userid

    msg = ""

    if request.method == 'POST':

        un = request.form['username']

        pd = request.form['password_1']

        print(un, pd)

        sql = "SELECT * FROM register WHERE email =? AND password=?"

        stmt = ibm_db.prepare(conn, sql)

        ibm_db.bind_param(stmt, 1, un)

        ibm_db.bind_param(stmt, 2, pd)

        ibm_db.execute(stmt)

        account = ibm_db.fetch_assoc(stmt)

        print(account)

        if account:

            session['loggedin'] = True

            session['id'] = account['EMAIL']

            userid = account['EMAIL']

            session['username'] = account['USERNAME']

            msg = 'Logged in successfully !'

            return rewrite('/dashboard')

```

```

else:

    msg = 'Incorrect username / password !'

    return render_template('login.html', msg=msg)

@app.route('/signup', methods=['POST', 'GET'])
def signup():

    mg = ""

    if request.method == "POST":

        username = request.form['username']

        email = request.form['email']

        pw = request.form['password']

        sql = 'SELECT * FROM register WHERE email =?'

        stmt = ibm_db.prepare(conn, sql)

        ibm_db.bind_param(stmt, 1, email)

        ibm_db.execute(stmt)

        acnt = ibm_db.fetch_assoc(stmt)

        print(acnt)

        if acnt:

            mg = 'Account already exists!!'

        elif not re.match(r'^@]+@[^@]+\.[^@]+', email):

            mg = 'Please enter the avalid email address'

        elif not re.match(r'[A-Za-z0-9]+', username):

            ms = 'name must contain only character and number'

        else:

            insert_sql = 'INSERT INTO register (USERNAME,FIRSTNAME,LASTNAME,EMAIL,PASSWORD) VALUES (?, ?, ?, ?, ?)'

            pstmt = ibm_db.prepare(conn, insert_sql)

            ibm_db.bind_param(pstmt, 1, username)

            ibm_db.bind_param(pstmt, 2, "firstname")

            ibm_db.bind_param(pstmt, 3, "lastname")

            # ibm_db.bind_param(pstmt,4,"123456789")

            ibm_db.bind_param(pstmt, 4, email)

            ibm_db.bind_param(pstmt, 5, pw)

            print(pstmt)

            ibm_db.execute(pstmt)

            mg = 'You have successfully registered click login!'

            return render_template("login.html", meg=mg)

    elif request.method == 'POST':

        msg = "fill out the form first!"

    return render_template("signup.html", meg=msg)

```

```
@app.route('/dashboard', methods=['POST', 'GET'])

@login_required

def dashBoard():

    headings = ("id", "name", "order_id", "location")

    data = (

        ("1", "lorem", "ipsum", "dolor"),

        ("2", "lorem", "ipsum", "dolor"),

        ("3", "lorem", "ipsum", "dolor"),

        ("1", "lorem", "ipsum", "dolor"),

        ("2", "lorem", "ipsum", "dolor"),

        ("3", "lorem", "ipsum", "dolor"),

    )

    return render_template("dashboard.html", headings=headings, data=data)
```

```
@app.route('/orders', methods=['POST', 'GET'])

@login_required

def orders():

    return render_template("orders.html")
```

```
@app.route('/suppliers', methods=['POST', 'GET'])

@login_required

def suppliers():

    return render_template("suppliers.html")
```

```
@app.route('/profile', methods=['POST', 'GET'])

@login_required

def profile():

    return render_template("profile.html")
```

```
@app.route('/logout', methods=['GET'])

@login_required

def logout():

    print(request)

    resp = make_response(render_template("login.html"))
```

```
session.clear()

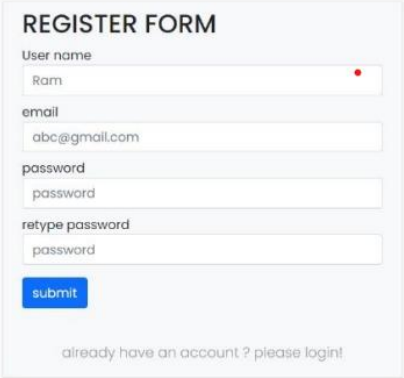
return resp

if __name__ == '__main__':

    app.run(debug=True)
```

**Output:**

**Registration:**



**REGISTER FORM**

User name  
Ram

email  
abc@gmail.com

password  
password

retype password  
password

submit

already have an account ? please login!

**Login:**

## LOGIN

username

password

Don't have an account? [Sign Up](#)