

**Project Development Phase**  
**Delivery of Sprint-1**

Date	2November 2022
Team ID	PNT2022TMID41025
Project Name	Inventory Management System For Retailer

**app.py**  
**creating flask application**

```
from flask import Flask, render_template, url_for, redirect
from flask_sqlalchemy import SQLAlchemy
from flask_login import UserMixin, login_user, LoginManager, login_required, logout_user,
current_user
from flask_wtf import FlaskForm
from wtforms import StringField, PasswordField, SubmitField
from wtforms.validators import InputRequired, Length, ValidationError
from flask import Flask, render_template, redirect, url_for, session
from flask_sqlalchemy import SQLAlchemy
from flask_script import Manager
from flask_uploads import UploadSet, configure_uploads, IMAGES
from flask_wtf import FlaskForm
from wtforms import StringField, IntegerField, TextAreaField, HiddenField, SelectField
import random
app = Flask(__name__)
db = SQLAlchemy(app)

@login_manager.user_loader
def load_user(user_id):
    __bind_key__='two'
    return User.query.get(int(user_id))

class User(db.Model, UserMixin):
    __bind_key__='two'
    id = db.Column(db.Integer, primary_key=True)
    username = db.Column(db.String(40), nullable=False, unique=True)
    password = db.Column(db.String(80), nullable=False)

class RegisterForm(FlaskForm):
    __bind_key__='two'
    username = StringField(validators=[
        InputRequired(), Length(min=4, max=40)], render_kw={"placeholder": "Username"})

    password = PasswordField(validators=[
        InputRequired(), Length(min=8, max=20)], render_kw={"placeholder": "Password"})

    submit = SubmitField('Register')

    def validate_username(self, username):
        __bind_key__='two'
```

```
existing_user_username = User.query.filter_by(
    username=username.data).first()
if existing_user_username:
    raise ValidationError(
        'That username already exists. Please choose a different one.')
```

```
class LoginForm(FlaskForm):
    __bind_key__='two'
    username = StringField(validators=[
        InputRequired(), Length(min=4, max=40)], render_kw={"placeholder": "Username"})

    password = PasswordField(validators=[
        InputRequired(), Length(min=8, max=20)], render_kw={"placeholder": "Password"})

    submit = SubmitField('Login')
```

```
@app.route('/')
def home():
    return render_template('home.html')
```

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

    form = LoginForm()
    if form.validate_on_submit():
        user = User.query.filter_by(username=form.username.data).first()
        if user:
            if bcrypt.check_password_hash(user.password, form.password.data):
                login_user(user)
                return redirect(url_for('index'))
        return render_template('login.html', form=form)
@app.route('/dashboard', methods=['GET', 'POST'])
@login_required
def dashboard():
    return render_template('dashboard.html')
```

```
@app.route('/logout', methods=['GET', 'POST'])
@login_required
def logout():
    logout_user()
    return redirect(url_for('login'))
```

```
@ app.route('/register', methods=['GET', 'POST'])
def register():
    __bind_key__='two'
    form = RegisterForm()

    if form.validate_on_submit():
        hashed_password = bcrypt.generate_password_hash(form.password.data)
        new_user = User(username=form.username.data, password=hashed_password)
```

```

    db.session.add(new_user)
    db.session.commit()
    return redirect(url_for('login'))

return render_template('register.html', form=form)

```

```

if __name__ == "__main__":
    app.run(debug=True)

```

## TEMPLATES:

### home.html- Dashboard page

```

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="{{ url_for('static', filename='css/homestyles.css') }}" />
    <title>Home</title>
</head>

<body>
<div class="login">
    <form id="login" method="" action="">

        <center><a href="{{ url_for('login') }}">Login Page</a><br></center>
        <center><a href="{{ url_for('register') }}">Register Page</a><br></center>
        <br><br>
    </form>

<script>
    window.watsonAssistantChatOptions = {
        integrationID: "1e1722cd-90ae-4848-9eb2-e559c5ab71c8", // The ID of this integration.
        region: "jp-tok", // The region your integration is hosted in.
        serviceInstanceID: "0738c83e-02d9-4636-8b32-0908a70361e3", // The ID of your service
instance.
        onLoad: function (instance) { instance.render(); }
    };
    setTimeout(function () {
        const t = document.createElement('script');
        t.src = "https://web-chat.global.assistant.watson.appdomain.cloud/versions/" +
(window.watsonAssistantChatOptions.clientVersion || 'latest') + "/WatsonAssistantChatEntry.js";
        document.head.appendChild(t);
    });
</script>

```

</body>

</html>

## **signin.html -signin page**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="stylesheet" href="{{ url\_for('static', filename='css/loginstyles.css') }}" />

<title>Login</title>

</head>

<body>

<h2>Login Page</h2><br>

<div class="login">

<form id="login" method="POST" action="">

{{ form.hidden\_tag() }}

<label><b>User Name

</b>

</label>

{{ form.username }}

<br><br>

<label><b>Password

</b>

</label>

{{ form.password }}

<br><br>

<center>{{ form.submit }}</center>

<br><br>

</form>

<script>

window.watsonAssistantChatOptions = {

integrationID: "1e1722cd-90ae-4848-9eb2-e559c5ab71c8", // The ID of this integration.

region: "jp-tok", // The region your integration is hosted in.

serviceInstanceID: "0738c83e-02d9-4636-8b32-0908a70361e3", // The ID of your service instance.

onLoad: function (instance) { instance.render(); }

};

setTimeout(function () {

const t = document.createElement('script');

t.src = "https://web-chat.global.assistant.watson.appdomain.cloud/versions/" +

(window.watsonAssistantChatOptions.clientVersion || 'latest') + "/WatsonAssistantChatEntry.js";

document.head.appendChild(t);

});

</script>

</body>

</html>

## **signup.html-Registration page**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="stylesheet" href="{{ url\_for('static', filename='css/loginstyles.css') }}" />

<title>Register</title>

</head>

<body>

<h2>Register Page</h2><br>

<div class="login">

<form id="login" method="POST" action="">

{{ form.hidden\_tag() }}

<label><b>User Name

</b>

</label>

{{ form.username }}

<br><br>

<label><b>Password

</b>

</label>

{{ form.password }}

<br><br>

<center>{{ form.submit }}</center>

<br><br>

</form>

<script>

window.watsonAssistantChatOptions = {

integrationID: "1e1722cd-90ae-4848-9eb2-e559c5ab71c8", // The ID of this integration.

region: "jp-tok", // The region your integration is hosted in.

serviceInstanceID: "0738c83e-02d9-4636-8b32-0908a70361e3", // The ID of your service instance.

onLoad: function (instance) { instance.render(); }

};

setTimeout(function () {

const t = document.createElement('script');

t.src = "https://web-chat.global.assistant.watson.appdomain.cloud/versions/" +

(window.watsonAssistantChatOptions.clientVersion || 'latest') + "/WatsonAssistantChatEntry.js";

document.head.appendChild(t);

});

</script>

</body>

</html>

**Output:**  
**Login page:**



