**INVENTORY MANAGEMENT**

**LOGIN**

|  |
| --- |
| { |
|  |  | // Use IntelliSense to learn about possible attributes. |
|  |  | // Hover to view descriptions of existing attributes. |
|  |  | // For more information, visit: https://go.microsoft.com/fwlink/?linkid=830387 |
|  |  | "version": "0.2.0", |
|  |  | "configurations": [ |
|  |  | { |
|  |  | "name": "Python: Current File (Integrated Terminal)", |
|  |  | "type": "python", |
|  |  | "request": "launch", |
|  |  | "program": "${file}", |
|  |  | "console": "integratedTerminal" |
|  |  | }, |
|  |  | { |
|  |  | "name": "Python: Attach", |
|  |  | "type": "python", |
|  |  | "request": "attach", |
|  |  | "port": 5678, |
|  |  | "host": "localhost" |
|  |  | }, |
|  |  | { |
|  |  | "name": "Python: Django", |
|  |  | "type": "python", |
|  |  | "request": "launch", |
|  |  | "program": "${workspaceFolder}/manage.py", |
|  |  | "console": "integratedTerminal", |
|  |  | "args": [ |
|  |  | "runserver", |
|  |  | "--noreload", |
|  |  | "--nothreading" |
|  |  | ], |
|  |  | "django": true |
|  |  | }, |
|  |  | { |
|  |  | "name": "Python: Flask", |
|  |  | "type": "python", |
|  |  | "request": "launch", |
|  |  | "module": "flask", |
|  |  | "env": { |
|  |  | "FLASK\_APP": "app.py" |
|  |  | }, |
|  |  | "args": [ |
|  |  | "run", |
|  |  | "--no-debugger", |
|  |  | "--no-reload" |
|  |  | ], |
|  |  | "jinja": true |
|  |  | }, |
|  |  | { |
|  |  | "name": "Python: Current File (External Terminal)", |
|  |  | "type": "python", |
|  |  | "request": "launch", |
|  |  | "program": "${file}", |
|  |  | "console": "externalTerminal" |
|  |  | } |
|  |  | ] |
|  |  | } |

ADDING A PRODUCT MOVEMENT

|  |
| --- |
| from flask\_wtf import FlaskForm |
|  |  | from inventorymanagement import app, mysql, db |
|  |  | from wtforms import StringField, PasswordField, SubmitField, BooleanField, IntegerField, SelectField, DateField |
|  |  | from wtforms.validators import DataRequired, Length, Email, EqualTo, ValidationError |
|  |  | from inventorymanagement.models import User |
|  |  |  |
|  |  | ########################Users###################################### |
|  |  |  |
|  |  | class RegistrationForm(FlaskForm): |
|  |  | username = StringField('Username', validators=[DataRequired(), Length(min=2, max=20)]) |
|  |  | email = StringField('Email', validators=[DataRequired(), Email()]) |
|  |  | password = PasswordField('Password', validators=[DataRequired()]) |
|  |  | confirm\_password = PasswordField('Confirm Password', validators=[DataRequired(), EqualTo('password')]) |
|  |  | submit = SubmitField('Sign Up') |
|  |  |  |
|  |  | def validate\_username(self, username): |
|  |  | user = User.query.filter\_by(username = username.data).first() |
|  |  | if user: |
|  |  | raise ValidationError('Username Taken') |
|  |  |  |
|  |  | def validate\_email(self, email): |
|  |  | user = User.query.filter\_by(email = email.data).first() |
|  |  | if user: |
|  |  | raise ValidationError('Email Taken') |
|  |  |  |
|  |  | class LoginForm(FlaskForm): |
|  |  | email = StringField('Email', validators=[DataRequired(), Email()]) |
|  |  | password = PasswordField('Password', validators=[DataRequired()]) |
|  |  | remember = BooleanField('Remember Me') |
|  |  | submit = SubmitField('Login') |
|  |  |  |
|  |  | ########################Products###################################### |
|  |  |  |
|  |  | class AddProduct(FlaskForm): |
|  |  | name = StringField('Product Name', validators=[DataRequired()]) |
|  |  | submit = SubmitField('Add Product') |
|  |  |  |
|  |  | ########################Locations###################################### |
|  |  |  |
|  |  | class AddLocation(FlaskForm): |
|  |  | name = StringField('Location Name', validators=[DataRequired()]) |
|  |  | submit = SubmitField('Add Location') |
|  |  |  |
|  |  | ########################ProductMovements###################################### |
|  |  |  |
|  |  | class AddProductMovement(FlaskForm): |
|  |  | conn = mysql.connect() |
|  |  | cursor = conn.cursor() |
|  |  | cursor.execute("SELECT location\_name FROM location") |
|  |  | locations = cursor.fetchall() |
|  |  | values = [] |
|  |  |  |
|  |  | for location in locations: |
|  |  | values.append((location[0], location[0])) |
|  |  |  |
|  |  | name = StringField('Product Name', validators=[DataRequired()]) |
|  |  | fromLocation = SelectField('From Location', choices=values, validators=[DataRequired()]) |
|  |  | toLocation = SelectField('To Location', choices=values, validators=[DataRequired()]) |
|  |  | quantity = IntegerField('Product Quantity', validators=[DataRequired()]) |
|  |  | timestamp = DateField('Date', validators=[DataRequired()]) |
|  |  | email = StringField('Email', validators=[DataRequired(), Email()]) |
|  |  | class ProductMovement(FlaskForm): |
|  |  | name = StringField('Product Name', validators=[DataRequired()]) |
|  |  | #timestamp = DateField('Date', validators=[DataRequired()]) |
|  |  | #fromLocation = SelectField('From Location', validators=[DataRequired()]) |
|  |  | #toLocation = SelectField('To Location', validators=[DataRequired()]) |
|  |  | #quantity = SelectField('Quantity', validators=[DataRequired()]) |
|  |  | #email = StringField('Email', validators=[DataRequired(), Email()]) |
|  |  | submit = SubmitField('Move Product') |

**26** [inventorymanagement/routes.py](https://github.com/hrwX/Inventory-Management/commit/b1d9ccbc0b95201cba79fcb4cbb69b70abc364d2" \l "diff-e25f1c72dd2323b5d1b9e1bc820731e26b1e85ded8ff85b8256f27bb4e9dbfd2" \o "inventorymanagement/routes.py)

|  |  |  |
| --- | --- | --- |
|  |  | @@ -1,7 +1,7 @@ |
|  |  | import datetime |
|  |  | from flask import render\_template, url\_for, flash, redirect, request |
|  |  | from inventorymanagement import app, bcrypt, mysql, db |
|  |  | from inventorymanagement.forms import RegistrationForm, LoginForm, AddProduct, AddLocation, AddProductMovement |
|  |  | from inventorymanagement.forms import RegistrationForm, LoginForm, AddProduct, AddLocation, ProductMovement |
|  |  | from inventorymanagement.models import User |
|  |  | from flask\_login import login\_user, current\_user, logout\_user, login\_required |
|  |  | from wtforms import StringField, PasswordField, SubmitField, BooleanField, IntegerField, DateTimeField, SelectField, Label |
|  | | @@ -20,7 +20,7 @@ def home(): |
|  |  | products = cursor.fetchone() |
|  |  | cursor.execute("SELECT COUNT(location\_id) FROM location") |
|  |  | locations = cursor.fetchone() |
|  |  | cursor.execute("SELECT COUNT(\*) FROM inventorymovement WHERE user\_id="+ str(current\_user.user\_id) +"") |
|  |  | cursor.execute("SELECT COUNT(\*) FROM productmovement WHERE user\_id="+ str(current\_user.user\_id) +"") |
|  |  | movements = cursor.fetchone() |
|  |  | sales = 5 |
|  |  | return render\_template('home.html', title='Home', products=products[0], locations=locations[0], sales=sales, movements=movements[0]) |
|  | | @@ -255,35 +255,35 @@ def view\_location(): |
|  |  | @app.route("/add\_productmovement?<int:product\_id>", methods=['GET', 'POST']) |
|  |  | @login\_required |
|  |  | def add\_productmovement(product\_id): |
|  |  | form = AddProductMovement() |
|  |  | print(form) |
|  |  | form = ProductMovement() |
|  |  |  |
|  |  | conn = mysql.connect() |
|  |  | cursor = conn.cursor() |
|  |  | time = datetime.date.today() |
|  |  | form = AddProductMovement() |
|  |  | cursor.execute("SELECT product\_name FROM product WHERE product\_id="+ str(product\_id) +"") |
|  |  | product\_name = cursor.fetchone() |
|  |  | cursor.execute("SELECT location\_name FROM location") |
|  |  | locations = cursor.fetchall() |
|  |  | time = datetime.date.today() |
|  |  | ranges = len(locations) |
|  |  | cursor.execute("SELECT \* FROM locationinventory WHERE locationinventory\_id="+ str(product\_id) +"") |
|  |  | quantities = cursor.fetchall() |
|  |  | print(form.validate\_on\_submit()) |
|  |  | if form.validate\_on\_submit(): |
|  |  | product\_name = form.name.data |
|  |  | from\_location = form.fromLocation.data |
|  |  | to\_location = form.toLocation.data |
|  |  | quantity = form.quantity.data |
|  |  | date = form.timestamp.data |
|  |  | email = form.email.data |
|  |  | print(product\_name,from\_location,to\_location,quantity) |
|  |  | from\_location = request.values.get('fromLocation') |
|  |  | to\_location = request.values.get('toLocation') |
|  |  | quantity = request.values.get('quantity') |
|  |  | date = request.values.get('timestamp') |
|  |  | email = request.values.get('email') |
|  |  | print(product\_name,from\_location,to\_location,quantity,date,email) |
|  |  | flash('Updated!', 'success') |
|  |  | return redirect(url\_for('view\_location')) |
|  |  | return render\_template('add\_productmovement.html', title='Movement', form=form, time=time, email=current\_user.email, product\_name=product\_name[0], locations=locations, quantities=quantities, ranges=ranges) |
|  |  |  |
|  |  | @app.route("/edit\_productmovement") |
|  |  | @login\_required |
|  |  | def edit\_productmovement(): |
|  |  | form = AddProductMovement() |
|  |  | form = ProductMovement() |
|  |  | return render\_template('edit\_productmovement.html', title='Movement', form=form) |
|  |  |  |
|  |  | @app.route("/view\_productmovement") |
|  | |  |

**61** [inventorymanagement/templates/add\_productmovement.html](https://github.com/hrwX/Inventory-Management/commit/b1d9ccbc0b95201cba79fcb4cbb69b70abc364d2" \l "diff-226b43aef2d606f1416e496b94b44672afaa8ab11cbb4adc322858628187f74e" \o "inventorymanagement/templates/add_productmovement.html)

|  |  |  |
| --- | --- | --- |
|  | | @@ -11,32 +11,50 @@ |
|  |  | <label class="form-control-label">Product Name</label> |
|  |  | <input name="name" id="name" class="form-control form-control-lg" value="{{ product\_name }}"> |
|  |  | </div> |
|  |  | <div class="form-group"> |
|  |  | <label class="form-control-label">Product Quantities at Respective Locations</label> |
|  |  | </div> |
|  |  | <div class="form-group" style="height:100px;overflow: auto;"> |
|  |  | {% for index in range(ranges) %} |
|  |  | <div class="form-group"> |
|  |  | <input class="form-control form-control-md" value="Quantity at {{ locations[index][0] }} location is {{ quantities[0][index] }}" disabled> |
|  |  | </div> |
|  |  | {% if quantities[0][index] != 0%} |
|  |  | <div class="form-group"> |
|  |  | <input class="form-control form-control-md" value="Quantity at {{ locations[index][0] }} location is {{ quantities[0][index] }}" disabled> |
|  |  | </div> |
|  |  | {% endif %} |
|  |  | {% endfor %} |
|  |  | </div> |
|  |  | <div class="form-group"> |
|  |  | {{ form.fromLocation.label(class="form-control-label") }} |
|  |  | {{ form.fromLocation(class="form-control form-control-lg") }} |
|  |  | <label class="form-control-label">From Location</label> |
|  |  | <select name="fromLocation" id="fromLocation" class="form-control form-control-lg"> |
|  |  | {% for index in range(ranges) %} |
|  |  | {% if quantities[0][index] != 0%} |
|  |  | <option value="{{ locations[index][0] }}" data-qty="{{ quantities[0][index] }}">{{ locations[index][0] }}</option> |
|  |  | {% endif %} |
|  |  | {% endfor %} |
|  |  | </select> |
|  |  | </div> |
|  |  | <div class="form-group"> |
|  |  | {{ form.toLocation.label(class="form-control-label") }} |
|  |  | {{ form.toLocation(class="form-control form-control-lg") }} |
|  |  | <label class="form-control-label">To Location</label> |
|  |  | <select name="toLocation" id="toLocation" class="form-control form-control-lg"> |
|  |  | {% for index in range(ranges) %} |
|  |  | <option value="{{ locations[index][0] }}">{{ locations[index][0] }}</option> |
|  |  | {% endfor %} |
|  |  | </select> |
|  |  | </div> |
|  |  | <div class="form-group"> |
|  |  | {{ form.quantity.label(class="form-control-label") }} |
|  |  | {{ form.quantity(class="form-control form-control-lg") }} |
|  |  | <label class="form-control-label">Quantity</label> |
|  |  | <select name="quantity" id="quantity" class="form-control form-control-lg"> |
|  |  | <div id="remove"> |
|  |  | </div> |
|  |  | </select> |
|  |  | </div> |
|  |  | <div class="form-group"> |
|  |  | <label class="form-control-label">Date</label> |
|  |  | <input name="timestamp" id="timestamp" class="form-control form-control-lg" value="{{ time }}" disabled> |
|  |  | <input name="timestamp" id="timestamp" class="form-control form-control-lg" value="{{ time }}" readonly> |
|  |  | </div> |
|  |  | <div class="form-group"> |
|  |  | <label class="form-control-label">Email</label> |
|  |  | <input name="email" id="email" class="form-control form-control-lg" value="{{ email }}" disabled> |
|  |  | <input name="email" id="email" class="form-control form-control-lg" value="{{ email }}" readonly> |
|  |  | </div> |
|  |  | </fieldset> |
|  |  | <div class="form-group"> |
|  | | @@ -46,10 +64,21 @@ |
|  |  | </div> |
|  |  | </div> |
|  |  | </div> |
|  |  | <script> |
|  |  | function value(){ |
|  |  |  |
|  |  | console.log(conceptName) |
|  |  | } |
|  |  | <script src="//ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js"></script> |
|  |  | <script language="JavaScript" type="text/javascript"> |
|  |  | $("#fromLocation").change(function () { |
|  |  | value = $(this).find(':selected').data('qty'); |
|  |  | select = document.getElementById('quantity'); |
|  |  | remove = document.getElementById('remove'); |
|  |  | append = ''; |
|  |  | empty = ''; |
|  |  | for (i = 1; i < value+1; i++) { |
|  |  | append = append + '<option id="remove" value="'+ i +'">'+ i +'</option>'; |
|  |  | } |
|  |  | select.innerHTML += append; |
|  |  | }); |
|  |  | </script> |
|  |  | <script type="text/javascript"> |
|  |  |  |
|  |  | </script> |

**ROUTES AND LOCATION**

|  |
| --- |
| def add\_location(): |
|  |  | form = AddLocation() |
|  |  | if form.validate\_on\_submit(): |
|  |  | location = Location(location\_name=form.name.data) |
|  |  | db.session.add(location) |
|  |  | db.session.commit() |
|  |  | conn = mysql.connect() |
|  |  | cursor = conn.cursor() |
|  |  | cursor.execute("INSERT INTO `location`(`location\_name`) VALUES ('"+ form.name.data +"')") |
|  |  | conn.commit() |
|  |  | location = (form.name.data).replace(" ","") |
|  |  | print(location) |
|  |  | print("ALTER TABLE locationinventory ADD COLUMN "+ form.name.data +" INTEGER DEFAULT 0") |
|  |  | cursor.execute("ALTER TABLE locationinventory ADD COLUMN "+ (form.name.data).replace(" ", "\_") +" INTEGER DEFAULT 0") |
|  |  | conn.commit() |
|  |  | conn.close() |
|  |  | print("INSERT INTO `location`(`location\_name`) VALUES ('"+ form.name.data +"')") |
|  |  | print("ALTER TABLE locationinventory ADD COLUMN "+ form.name.data +" INTEGER DEFAULT 0") |
|  |  | flash('Location Added') |
|  |  | return redirect(url\_for('view\_location')) |
|  |  | return render\_template('add\_location.html', title='Location', form=form) |
|  | | @@ -119,25 +128,27 @@ def add\_location(): |
|  |  | @login\_required |
|  |  | def edit\_location(location\_id): |
|  |  | form = AddLocation() |
|  |  | location = Location.query.get(location\_id) |
|  |  | print(location.location\_id) |
|  |  | print(form.name.data) |
|  |  | conn = mysql.connect() |
|  |  | cursor = conn.cursor() |
|  |  | cursor.execute("SELECT \* FROM location WHERE location\_id='"+ str(location\_id) +"'") |
|  |  | conn.commit() |
|  |  | location = cursor.fetchone() |
|  |  | if form.validate\_on\_submit(): |
|  |  | location.location\_id = location.location\_id |
|  |  | location.location\_name = form.name.data |
|  |  | print(location.location\_id) |
|  |  | print(location.location\_name) |
|  |  | db.session.commit() |
|  |  | cursor.execute("UPDATE location SET location\_name='"+ form.name.data +"' WHERE location\_id='"+ str(location\_id) +"'") |
|  |  | print(cursor) |
|  |  | flash('Updated!', 'success') |
|  |  | return redirect(url\_for('view\_location')) |
|  |  | elif request.method == 'GET': |
|  |  | form.name.data = location.location\_name |
|  |  | form.name.data = location\_id |
|  |  | return render\_template('edit\_location.html', title='Location', form=form, location=location) |
|  |  |  |
|  |  | @app.route("/view\_location") |
|  |  | @login\_required |
|  |  | def view\_location(): |
|  |  | locations = Location.query.all() |
|  |  | conn = mysql.connect() |
|  |  | cursor = conn.cursor() |
|  |  | cursor.execute("SELECT \* FROM location") |
|  |  | locations = cursor.fetchall() |
|  |  | return render\_template('view\_location.html', title='Location', locations=locations) |
|  |  |  |
|  |  | ########################ProductMovements###################################### |
|  | |  |

**2** [inventorymanagement/templates/edit\_location.html](https://github.com/hrwX/Inventory-Management/commit/a184ced2ab9ca8cd1683c19d94b906106d0ec295" \l "diff-fde111734bf0527a17b7d7cf50d83a31f7610f7bb643f618a1ab84bc3fb7f668" \o "inventorymanagement/templates/edit_location.html)

|  |  |  |
| --- | --- | --- |
|  | | @@ -9,7 +9,7 @@ |
|  |  | <fieldset class="form-group"> |
|  |  | <div class="form-group"> |
|  |  | <label class="form-control-label" for="name">Edit Location Name</label> |
|  |  | <input class="form-control form-control-lg" id="name" name="name" required type="text" value="{{ location.location\_name }}"> |
|  |  | <input class="form-control form-control-lg" id="name" name="name" required type="text" value="{{ location[1] }}"> |
|  |  | </div> |
|  |  | </fieldset> |
|  |  | <div class="form-group"> |
|  | |  |

**36** [inventorymanagement/templates/view\_location.html](https://github.com/hrwX/Inventory-Management/commit/a184ced2ab9ca8cd1683c19d94b906106d0ec295#diff-633ec90f9c1185849315e8caf6ad5774026853742279cbd77b9a3968ab761a81)

|  |  |  |
| --- | --- | --- |
|  | | @@ -49,35 +49,27 @@ <h2>Inventory Location</h2> |
|  |  | <br> |
|  |  | </div> |
|  |  | <div class="col-md-12" style="height:600px; overflow: auto;"> |
|  |  | {% for location in locations %} |
|  |  | <div class="row"> |
|  |  | <div class="col-md-12 border rounded pt-2" style=" height:50px;"> |
|  |  | <div class="row"> |
|  |  | <div class="col-md-4 border-left pt-2"> |
|  |  | {{ location.location\_id }} |
|  |  | </div> |
|  |  | <div class="col-md-4 border-left pt-2"> |
|  |  | {{ location.location\_name }} |
|  |  | </div> |
|  |  | <div class="col-md-4 border-left pt-2"> |
|  |  | <a href="{{ url\_for('edit\_location', location\_id=location.location\_id) }}">Edit</a> |
|  |  | </div> |
|  |  | {% for location in locations %} |
|  |  | <div class="row"> |
|  |  | <div class="col-md-12 border rounded pt-2" style=" height:50px;"> |
|  |  | <div class="row"> |
|  |  | <div class="col-md-4 border-left pt-2"> |
|  |  | {{ location[0] }} |
|  |  | </div> |
|  |  | <div class="col-md-4 border-left pt-2"> |
|  |  | {{ location[1] }} |
|  |  | </div> |
|  |  | <div class="col-md-4 border-left pt-2"> |
|  |  | <a href="{{ url\_for('edit\_location', location\_id=location[0]) }}">Edit</a> |
|  |  | </div> |
|  |  | </div> |
|  |  | </div> |
|  |  | {% endfor %} |
|  |  | </div> |
|  |  | {% endfor %} |
|  |  | </div> |
|  |  | </div> |
|  |  | </div> |
|  |  | </div> |
|  |  | <br> |
|  |  | </div> |
|  |  | <script> |
|  |  | jQuery(document).ready(function($) { |
|  |  | $(".clickable-row").click(function() { |
|  |  | window.location = $(this).data("href"); |
|  |  | console.log('A') |
|  |  | }); |
|  |  | }); |
|  |  | </script> |

|  |
| --- |
| import datetime |
|  | from flask import render\_template, url\_for, flash, redirect, request |
|  | from inventorymanagement import app, bcrypt, mysql, db |
|  | from inventorymanagement.forms import RegistrationForm, LoginForm, AddProduct, AddLocation, ProductMovement |
|  | from inventorymanagement.models import User |
|  | from flask\_login import login\_user, current\_user, logout\_user, login\_required |
|  | from wtforms import StringField, PasswordField, SubmitField, BooleanField, IntegerField, DateTimeField, SelectField, Label |
|  | from wtforms.validators import DataRequired, Length, Email, EqualTo, ValidationError |
|  |  |
|  | @app.route("/") |
|  | @app.route("/anon") |
|  | def anon(): |
|  | return redirect(url\_for('login')) |
|  |  |
|  | @app.route("/home") |
|  | def home(): |
|  | conn = mysql.connect() |
|  | cursor = conn.cursor() |
|  | cursor.execute("SELECT COUNT(product\_id) FROM product WHERE user\_id="+ str(current\_user.user\_id) +"") |
|  | products = cursor.fetchone() |
|  | cursor.execute("SELECT COUNT(location\_id) FROM location") |
|  | locations = cursor.fetchone() |
|  | cursor.execute("SELECT COUNT(\*) FROM productmovement WHERE user\_id="+ str(current\_user.user\_id) +"") |
|  | movements = cursor.fetchone() |
|  | sales = 5 |
|  | return render\_template('home.html', title='Home', products=products[0], locations=locations[0], sales=sales, movements=movements[0]) |
|  |  |
|  | @app.route("/about") |
|  | def about(): |
|  | return render\_template('about.html', title="About") |
|  |  |
|  | ########################Users###################################### |
|  |  |
|  | @app.route("/register", methods=['GET', 'POST']) |
|  | def register(): |
|  | if current\_user.is\_authenticated: |
|  | return redirect(url\_for('home')) |
|  | form = RegistrationForm() |
|  | if form.validate\_on\_submit(): |
|  | hashed\_password = bcrypt.generate\_password\_hash(form.password.data).decode('utf-8') |
|  | user = User(username=form.username.data, email=form.email.data, password=hashed\_password) |
|  | db.session.add(user) |
|  | db.session.commit() |
|  | flash('Your Account has been created', 'success') |
|  | return redirect(url\_for('login')) |
|  | return render\_template('register.html', title='Register', form=form) |
|  |  |
|  | @app.route("/login", methods=['GET', 'POST']) |
|  | def login(): |
|  | if current\_user.is\_authenticated: |
|  | return redirect(url\_for('home')) |
|  | form = LoginForm() |
|  | if form.validate\_on\_submit(): |
|  | user = User.query.filter\_by(email=form.email.data).first() |
|  | if user and bcrypt.check\_password\_hash(user.password, form.password.data): |
|  | login\_user(user, remember=form.remember.data) |
|  | next\_page = request.args.get('next') |
|  | return redirect(next\_page) if next\_page else redirect(url\_for('home')) |
|  | else: |
|  | flash('Login Unsuccessful', 'danger') |
|  | return render\_template('login.html', title='Login', form=form) |
|  |  |
|  | @app.route("/logout") |
|  | def logout(): |
|  | logout\_user() |
|  | return redirect(url\_for('anon')) |
|  |  |
|  | ########################Products###################################### |
|  |  |
|  | @app.route("/add\_product", methods=['GET', 'POST']) |
|  | @login\_required |
|  | def add\_product(): |
|  | conn = mysql.connect() |
|  | cursor = conn.cursor() |
|  | cursor.execute("SELECT location\_name FROM location") |
|  | locations = cursor.fetchall() |
|  | places = [] |
|  | for location in locations: |
|  | places.append(location[0]) |
|  |  |
|  | form = AddProduct() |
|  |  |
|  | if form.validate\_on\_submit(): |
|  | name = form.name.data |
|  | input\_values = request.form.getlist('places[]') |
|  | totalquantity = 0 |
|  |  |
|  | locationinventory = "INSERT INTO `locationinventory`(" |
|  |  |
|  | for count,place in enumerate(places): |
|  | locationinventory = locationinventory + "`"+ place +"`" |
|  | if count != len(places)-1: |
|  | locationinventory = locationinventory + "," |
|  |  |
|  | locationinventory = locationinventory + ",`user\_id`) VALUES (" |
|  |  |
|  | for count,input\_value in enumerate(input\_values): |
|  | totalquantity = totalquantity + int(input\_value) |
|  | locationinventory = locationinventory + "'"+ input\_value +"'" |
|  | if count != len(input\_values)-1: |
|  | locationinventory = locationinventory + "," |
|  |  |
|  | locationinventory = locationinventory + ",'"+ str(current\_user.user\_id) +"')" |
|  | location = "INSERT INTO `product`(`product\_name`,`product\_quantity`,`user\_id`) VALUES ('"+ name +"','"+ str(totalquantity) +"','"+ str(current\_user.user\_id) +"')" |
|  | print(locationinventory) |
|  | cursor.execute(locationinventory) |
|  | conn.commit() |
|  | cursor.execute(location) |
|  | conn.commit() |
|  | conn.close() |
|  | flash('Done', 'success') |
|  | return redirect(url\_for('view\_product')) |
|  | return render\_template('add\_product.html', title='Product', form=form, locations=locations) |
|  |  |
|  | @app.route("/edit\_product?<int:product\_id>", methods=['GET', 'POST']) |
|  | def edit\_product(product\_id): |
|  | form = AddProduct() |
|  | conn = mysql.connect() |
|  | cursor = conn.cursor() |
|  | values = "Select \* from product WHERE product\_id="+ str(product\_id) +"" |
|  | values = cursor.execute(values) |
|  | values = cursor.fetchone() |
|  | locations = "SELECT location\_name FROM location" |
|  | locations = cursor.execute(locations) |
|  | locations = cursor.fetchall() |
|  | places = [] |
|  | for location in locations: |
|  | places.append(location[0]) |
|  | inventory = "Select \* from locationinventory WHERE locationinventory\_id="+ str(product\_id) +"" |
|  | inventory = cursor.execute(inventory) |
|  | inventory = cursor.fetchone() |
|  | ranges = len(locations) |
|  | quantities = [] |
|  | for inventory in inventory: |
|  | quantities.append(inventory) |
|  | print(quantities) |
|  | for count in range(2): |
|  | quantities.pop(0) |
|  | print(quantities) |
|  | if form.validate\_on\_submit(): |
|  | name = form.name.data |
|  | input\_values = request.form.getlist('places[]') |
|  | print(input\_values) |
|  | totalquantity = 0 |
|  | locationinventory = "UPDATE `locationinventory` SET " |
|  |  |
|  | for index in range(ranges): |
|  | locationinventory = locationinventory + "`" + locations[index][0] + "`='" + str(input\_values[index]) +"'" |
|  | totalquantity = totalquantity + int(input\_values[index]) |
|  | if index != len(input\_values)-1: |
|  | locationinventory = locationinventory + "," |
|  |  |
|  | locationinventory = locationinventory + " WHERE `locationinventory\_id`=" + str(product\_id) |
|  | location = "UPDATE `product` SET `product\_name`='"+ name +"',`product\_quantity`='"+ str(totalquantity) +"' WHERE product\_id="+ str(product\_id) |
|  | print(locationinventory) |
|  | cursor.execute(locationinventory) |
|  | conn.commit() |
|  | print(location) |
|  | cursor.execute(location) |
|  | conn.commit() |
|  | conn.close() |
|  | flash('Done', 'success') |
|  | return redirect(url\_for('view\_product')) |
|  | return render\_template('edit\_product.html', title='Product', form=form, values=values, locations=locations, quantities=quantities, ranges=ranges) |
|  |  |
|  | @app.route("/product\_info?<int:product\_id>", methods=['GET', 'POST']) |
|  | def product\_info(product\_id): |
|  | form = AddProduct() |
|  | conn = mysql.connect() |
|  | cursor = conn.cursor() |
|  | values = "Select \* from product WHERE product\_id="+ str(product\_id) +"" |
|  | values = cursor.execute(values) |
|  | values = cursor.fetchone() |
|  | locations = "SELECT location\_name FROM location" |
|  | locations = cursor.execute(locations) |
|  | locations = cursor.fetchall() |
|  | places = [] |
|  | for location in locations: |
|  | places.append(location[0]) |
|  | inventory = "Select \* from locationinventory WHERE locationinventory\_id="+ str(product\_id) +"" |
|  | inventory = cursor.execute(inventory) |
|  | inventory = cursor.fetchone() |
|  | ranges = len(locations) |
|  | quantities = [] |
|  | for inventory in inventory: |
|  | quantities.append(inventory) |
|  | for count in range(2): |
|  | quantities.pop(0) |
|  | return render\_template('product\_info.html', title='Product', form=form, values=values, locations=locations, quantities=quantities, ranges=ranges) |
|  |  |
|  |  |
|  | @app.route("/view\_product") |
|  | @login\_required |
|  | def view\_product(): |
|  | conn = mysql.connect() |
|  | cursor = conn.cursor() |
|  | products = cursor.execute("SELECT \* FROM product WHERE user\_id='"+ str(current\_user.user\_id)+"'") |
|  | products = cursor.fetchall() |
|  | inventory\_places = cursor.execute("SELECT \* FROM locationinventory") |
|  | inventory\_places = cursor.fetchall() |
|  | return render\_template('view\_product.html', title='Product', products=products, inventory\_places=inventory\_places) |
|  |  |
|  | ########################Locations###################################### |
|  |  |
|  | @app.route("/add\_location", methods=['GET', 'POST']) |
|  | @login\_required |
|  | def add\_location(): |
|  | form = AddLocation() |
|  | if form.validate\_on\_submit(): |
|  | conn = mysql.connect() |
|  | cursor = conn.cursor() |
|  | count = cursor.execute("SELECT location\_name FROM location WHERE location\_name='"+ (form.name.data).replace(" ", "\_") +"'") |
|  | if count == 0: |
|  | cursor.execute("INSERT INTO `location`(`location\_name`) VALUES ('"+ (form.name.data).replace(" ", "\_") +"')") |
|  | conn.commit() |
|  | cursor.execute("ALTER TABLE locationinventory ADD COLUMN "+ (form.name.data).replace(" ", "\_") +" INTEGER DEFAULT 0") |
|  | conn.commit() |
|  | conn.close() |
|  | flash('Location Added', 'success') |
|  | return redirect(url\_for('view\_location')) |
|  | else: |
|  | conn.close() |
|  | flash('Location Exixts', 'danger') |
|  | return redirect(url\_for('add\_location')) |
|  |  |
|  | return render\_template('add\_location.html', title='Location', form=form) |
|  |  |
|  | @app.route("/edit\_location?<int:location\_id>", methods=['GET', 'POST']) |
|  | @login\_required |
|  | def edit\_location(location\_id): |
|  | form = AddLocation() |
|  | conn = mysql.connect() |
|  | cursor = conn.cursor() |
|  | cursor.execute("SELECT \* FROM location WHERE location\_id='"+ str(location\_id) +"'") |
|  | location = cursor.fetchone() |
|  | if form.validate\_on\_submit(): |
|  | cursor.execute("UPDATE location SET location\_name='"+ form.name.data +"' WHERE location\_id='"+ str(location\_id) +"'") |
|  | conn.commit() |
|  | flash('Updated!', 'success') |
|  | return redirect(url\_for('view\_location')) |
|  | elif request.method == 'GET': |
|  | form.name.data = location\_id |
|  | return render\_template('edit\_location.html', title='Location', form=form, location=location) |
|  |  |
|  | @app.route("/view\_location") |
|  | @login\_required |
|  | def view\_location(): |
|  | conn = mysql.connect() |
|  | cursor = conn.cursor() |
|  | cursor.execute("SELECT \* FROM location") |
|  | locations = cursor.fetchall() |
|  | return render\_template('view\_location.html', title='Location', locations=locations) |
|  |  |
|  | ########################ProductMovements###################################### |
|  |  |
|  | @app.route("/add\_productmovement?<int:product\_id>", methods=['GET', 'POST']) |
|  | @login\_required |
|  | def add\_productmovement(product\_id): |
|  | form = ProductMovement() |
|  | conn = mysql.connect() |
|  | cursor = conn.cursor() |
|  | cursor.execute("SELECT product\_name FROM product WHERE product\_id="+ str(product\_id) +"") |
|  | product\_name = cursor.fetchone() |
|  | cursor.execute("SELECT location\_name FROM location") |
|  | locations = cursor.fetchall() |
|  | time = datetime.date.today() |
|  | ranges = len(locations) |
|  | cursor.execute("SELECT \* FROM locationinventory WHERE locationinventory\_id="+ str(product\_id) +"") |
|  | inventory = cursor.fetchone() |
|  | quantities = [] |
|  | for inventory in inventory: |
|  | quantities.append(inventory) |
|  | for count in range(2): |
|  | quantities.pop(0) |
|  | print(quantities[5]) |
|  | if form.validate\_on\_submit(): |
|  | product\_name = form.name.data |
|  | from\_location = request.values.get('fromLocation') |
|  | to\_location = request.values.get('toLocation') |
|  | quantity = request.values.get('quantity') |
|  | date = request.values.get('timestamp') |
|  | email = request.values.get('email') |
|  | query = "SELECT "+ str(from\_location) +","+ str(to\_location) +" FROM locationinventory WHERE locationinventory\_id="+ str(product\_id) +"" |
|  | cursor.execute(query) |
|  | value = cursor.fetchone() |
|  | from\_location\_qty = value[0] - int(quantity) |
|  | to\_location\_qty = value[1] + int(quantity) |
|  | query = "UPDATE locationinventory SET "+ str(from\_location) +"='"+ str(from\_location\_qty) +"', "+ str(to\_location) +"='"+ str(to\_location\_qty) +"' WHERE locationinventory\_id="+ str(product\_id) +"" |
|  | print(query) |
|  | cursor.execute(query) |
|  | conn.commit() |
|  | query = "INSERT INTO `productmovement`(`product\_id`, `product\_name`, `from\_location\_name`, `to\_location\_name`, `product\_quantity`, `timestamp`, `user\_id`) VALUES ('"+ str(product\_id) +"','"+ form.name.data +"','"+ str(from\_location) +"','"+ str(to\_location) +"','"+ quantity +"','"+ date +"','"+ str(current\_user.user\_id) +"')" |
|  | print(query) |
|  | cursor.execute(query) |
|  | conn.commit() |
|  | conn.close() |
|  | flash('Updated!', 'success') |
|  | return redirect(url\_for('view\_location')) |
|  | return render\_template('add\_productmovement.html', title='Movement', form=form, time=time, email=current\_user.email, product\_name=product\_name[0], locations=locations, quantities=quantities, ranges=ranges) |
|  |  |
|  | @app.route("/edit\_productmovement?<int:productmovement\_id>") |
|  | @login\_required |
|  | def edit\_productmovement(productmovement\_id): |
|  | conn = mysql.connect() |
|  | cursor = conn.cursor() |
|  | cursor.execute("SELECT \* FROM productmovement WHERE productmovement\_id="+ str(productmovement\_id) +"") |
|  | query = cursor.fetchone() |
|  | product\_id = query[1] |
|  | from\_location = query[3] |
|  | to\_location = query[4] |
|  | quantity = query[5] |
|  | cursor.execute("SELECT "+ str(from\_location) +","+ str(to\_location) +" FROM locationinventory WHERE locationinventory\_id="+ str(product\_id) +"") |
|  | inventory = cursor.fetchone() |
|  | from\_location\_qty = inventory[0] + quantity |
|  | to\_location\_qty = inventory[1] - quantity |
|  | cursor.execute("UPDATE locationinventory SET "+ str(from\_location) +"='"+ str(from\_location\_qty) +"', "+ str(to\_location) +"='"+ str(to\_location\_qty) +"' WHERE locationinventory\_id="+ str(product\_id) +"") |
|  | conn.commit() |
|  | cursor.execute("DELETE FROM productmovement WHERE productmovement\_id="+ str(productmovement\_id) +"") |
|  | conn.commit() |
|  | return redirect(url\_for('view\_productmovement')) |
|  | return render\_template('', title='Movement', form=form) |
|  |  |
|  | @app.route("/view\_productmovement") |
|  | @login\_required |
|  | def view\_productmovement(): |
|  | conn = mysql.connect() |
|  | cursor = conn.cursor() |
|  | cursor.execute("SELECT \* FROM productmovement WHERE user\_id="+ str(current\_user.user\_id) +"") |
|  | movements = cursor.fetchall() |
|  | counts = len(movements) |
|  | return render\_template('view\_productmovement.html', title='Movement', movements=movements, counts=count |

**SQL**

|  |
| --- |
| -- phpMyAdmin SQL Dump |
|  | -- version 4.6.6deb5 |
|  | -- https://www.phpmyadmin.net/ |
|  | -- |
|  | -- Host: localhost:3306 |
|  | -- Generation Time: Sep 13, 2018 at 01:11 PM |
|  | -- Server version: 5.7.23-0ubuntu0.18.04.1 |
|  | -- PHP Version: 7.2.7-0ubuntu0.18.04.2 |
|  |  |
|  | SET SQL\_MODE = "NO\_AUTO\_VALUE\_ON\_ZERO"; |
|  | SET time\_zone = "+00:00"; |
|  |  |
|  |  |
|  | /\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/; |
|  | /\*!40101 SET @OLD\_CHARACTER\_SET\_RESULTS=@@CHARACTER\_SET\_RESULTS \*/; |
|  | /\*!40101 SET @OLD\_COLLATION\_CONNECTION=@@COLLATION\_CONNECTION \*/; |
|  | /\*!40101 SET NAMES utf8mb4 \*/; |
|  |  |
|  | -- |
|  | -- Database: `inventory\_management` |
|  | -- |
|  |  |
|  | -- -------------------------------------------------------- |
|  |  |
|  | -- |
|  | -- Table structure for table `location` |
|  | -- |
|  |  |
|  | CREATE TABLE `location` ( |
|  | `location\_id` int(11) NOT NULL, |
|  | `location\_name` varchar(60) NOT NULL |
|  | ) ENGINE=InnoDB DEFAULT CHARSET=latin1; |
|  |  |
|  | -- |
|  | -- Dumping data for table `location` |
|  | -- |
|  |  |
|  | INSERT INTO `location` (`location\_id`, `location\_name`) VALUES |
|  | (1, 'Mumbai'), |
|  | (2, 'Delhi'), |
|  | (3, 'New\_Delhi'), |
|  | (4, 'Chennai'), |
|  | (5, 'Bangalore'), |
|  | (6, 'Hyderabad'), |
|  | (7, 'Kolkata'), |
|  | (8, 'Ahmedabad'), |
|  | (11, 'Mirzapur'), |
|  | (12, 'Nagpur'), |
|  | (13, 'Pune'), |
|  | (19, 'Navi\_Mumbai'), |
|  | (21, 'Kundapura'), |
|  | (22, 'Kozhikode'); |
|  |  |
|  | -- -------------------------------------------------------- |
|  |  |
|  | -- |
|  | -- Table structure for table `locationinventory` |
|  | -- |
|  |  |
|  | CREATE TABLE `locationinventory` ( |
|  | `locationinventory\_id` int(11) NOT NULL, |
|  | `user\_id` int(11) NOT NULL, |
|  | `Mumbai` int(11) DEFAULT '0', |
|  | `Delhi` int(11) DEFAULT '0', |
|  | `New\_Delhi` int(11) DEFAULT '0', |
|  | `Chennai` int(11) DEFAULT '0', |
|  | `Bangalore` int(11) DEFAULT '0', |
|  | `Hyderabad` int(11) DEFAULT '0', |
|  | `Kolkata` int(11) DEFAULT '0', |
|  | `Ahmedabad` int(11) DEFAULT '0', |
|  | `Mirzapur` int(11) DEFAULT '0', |
|  | `Nagpur` int(11) DEFAULT '0', |
|  | `Pune` int(11) DEFAULT '0', |
|  | `Navi\_Mumbai` int(11) DEFAULT '0', |
|  | `Kundapura` int(11) DEFAULT '0', |
|  | `Kozhikode` int(11) DEFAULT '0' |
|  | ) ENGINE=InnoDB DEFAULT CHARSET=latin1; |
|  |  |
|  | -- |
|  | -- Dumping data for table `locationinventory` |
|  | -- |
|  |  |
|  | INSERT INTO `locationinventory` (`locationinventory\_id`, `user\_id`, `Mumbai`, `Delhi`, `New\_Delhi`, `Chennai`, `Bangalore`, `Hyderabad`, `Kolkata`, `Ahmedabad`, `Mirzapur`, `Nagpur`, `Pune`, `Navi\_Mumbai`, `Kundapura`, `Kozhikode`) VALUES |
|  | (1, 1, 0, 0, 65, 78, 1465, 0, 152, 0, 0, 0, 0, 0, 0, 0), |
|  | (2, 1, 37, 6, 489, 1, 6, 5, 0, 5, 0, 0, 5, 0, 0, 15), |
|  | (3, 1, 5, 3, 1, 3, 3, 3, 3, 3, 52, 6, 489, 1, 6, 5), |
|  | (4, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0), |
|  | (5, 1, 15, 30, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0), |
|  | (6, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0), |
|  | (7, 1, 48, 52, 6, 475, 1, 6, 5, 0, 5, 0, 0, 5, 0, 0), |
|  | (8, 1, 46, 46, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0), |
|  | (9, 1, 13, 12, 12, 12, 12, 12, 12, 12, 0, 0, 0, 0, 0, 0), |
|  | (10, 1, 20, 20, 20, 20, 20, 20, 20, 20, 0, 0, 0, 0, 0, 0), |
|  | (11, 2, 12, 12, 12, 12, 12, 0, 0, 0, 0, 0, 0, 0, 0, 0), |
|  | (12, 2, 12, 12, 12, 12, 12, 0, 0, 0, 0, 0, 0, 0, 0, 0), |
|  | (13, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0), |
|  | (14, 2, 212, 12, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0), |
|  | (15, 2, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0), |
|  | (16, 2, 456, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0), |
|  | (17, 2, 456, 5, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0); |
|  |  |
|  | -- -------------------------------------------------------- |
|  |  |
|  | -- |
|  | -- Table structure for table `product` |
|  | -- |
|  |  |
|  | CREATE TABLE `product` ( |
|  | `product\_id` int(11) NOT NULL, |
|  | `product\_name` varchar(60) NOT NULL, |
|  | `product\_quantity` int(11) NOT NULL, |
|  | `user\_id` int(11) NOT NULL |
|  | ) ENGINE=InnoDB DEFAULT CHARSET=latin1; |
|  |  |
|  | -- |
|  | -- Dumping data for table `product` |
|  | -- |
|  |  |
|  | INSERT INTO `product` (`product\_id`, `product\_name`, `product\_quantity`, `user\_id`) VALUES |
|  | (1, 'Product 1', 1760, 1), |
|  | (2, 'Product 2', 569, 1), |
|  | (3, 'Product 3', 583, 1), |
|  | (4, 'Product 4', 0, 1), |
|  | (5, 'Product 5', 45, 1), |
|  | (6, 'Product 6', 0, 1), |
|  | (7, 'Product 7', 597, 1), |
|  | (8, 'Product 8', 92, 1), |
|  | (9, 'Product 9', 97, 1), |
|  | (10, 'Product 10', 160, 1), |
|  | (11, 'Product 11', 60, 2), |
|  | (12, 'Product 12', 60, 2), |
|  | (13, 'Product 13', 0, 2), |
|  | (14, 'Product 14', 224, 2), |
|  | (15, 'Product 15', 2, 2), |
|  | (16, 'Product 16', 456, 2), |
|  | (17, 'Product 17', 461, 2); |
|  |  |
|  | -- -------------------------------------------------------- |
|  |  |
|  | -- |
|  | -- Table structure for table `productmovement` |
|  | -- |
|  |  |
|  | CREATE TABLE `productmovement` ( |
|  | `productmovement\_id` int(11) NOT NULL, |
|  | `product\_id` int(11) NOT NULL, |
|  | `product\_name` varchar(60) NOT NULL, |
|  | `from\_location\_name` varchar(60) NOT NULL, |
|  | `to\_location\_name` varchar(60) NOT NULL, |
|  | `product\_quantity` int(11) NOT NULL, |
|  | `timestamp` date NOT NULL, |
|  | `user\_id` int(11) NOT NULL |
|  | ) ENGINE=InnoDB DEFAULT CHARSET=latin1; |
|  |  |
|  | -- |
|  | -- Dumping data for table `productmovement` |
|  | -- |
|  |  |
|  | INSERT INTO `productmovement` (`productmovement\_id`, `product\_id`, `product\_name`, `from\_location\_name`, `to\_location\_name`, `product\_quantity`, `timestamp`, `user\_id`) VALUES |
|  | (1, 3, 'Product 3', 'New\_Delhi', 'Mumbai', 2, '2018-09-13', 1), |
|  | (3, 7, 'Product 7', 'Chennai', 'Mumbai', 14, '2018-09-13', 1), |
|  | (5, 5, 'Product 5', 'Delhi', 'Mumbai', 15, '2018-09-13', 1), |
|  | (7, 2, 'Product 2', 'Mumbai', 'Kozhikode', 15, '2018-09-13', 1); |
|  |  |
|  | -- -------------------------------------------------------- |
|  |  |
|  | -- |
|  | -- Table structure for table `user` |
|  | -- |
|  |  |
|  | CREATE TABLE `user` ( |
|  | `user\_id` int(11) NOT NULL, |
|  | `user\_name` varchar(20) NOT NULL, |
|  | `user\_email` varchar(60) NOT NULL, |
|  | `user\_password` varchar(60) NOT NULL |
|  | ) ENGINE=InnoDB DEFAULT CHARSET=latin1; |
|  |  |
|  | -- |
|  | -- Dumping data for table `user` |
|  | -- |
|  |  |
|  | INSERT INTO `user` (`user\_id`, `user\_name`, `user\_email`, `user\_password`) VALUES |
|  | (1, 'himanshu', 'himanshuwarekar@yahoo.com', '$2b$12$IjFgSnAJCdxQrksUSbTq9Oox7YzBx4Q0WdcRM3FJaLP5CAJ4H9O8i'); |
|  |  |
|  | -- |
|  | -- Indexes for dumped tables |
|  | -- |
|  |  |
|  | -- |
|  | -- Indexes for table `location` |
|  | -- |
|  | ALTER TABLE `location` |
|  | ADD PRIMARY KEY (`location\_id`); |
|  |  |
|  | -- |
|  | -- Indexes for table `locationinventory` |
|  | -- |
|  | ALTER TABLE `locationinventory` |
|  | ADD PRIMARY KEY (`locationinventory\_id`); |
|  |  |
|  | -- |
|  | -- Indexes for table `product` |
|  | -- |
|  | ALTER TABLE `product` |
|  | ADD PRIMARY KEY (`product\_id`); |
|  |  |
|  | -- |
|  | -- Indexes for table `productmovement` |
|  | -- |
|  | ALTER TABLE `productmovement` |
|  | ADD PRIMARY KEY (`productmovement\_id`); |
|  |  |
|  | -- |
|  | -- Indexes for table `user` |
|  | -- |
|  | ALTER TABLE `user` |
|  | ADD PRIMARY KEY (`user\_id`); |
|  |  |
|  | -- |
|  | -- AUTO\_INCREMENT for dumped tables |
|  | -- |
|  |  |
|  | -- |
|  | -- AUTO\_INCREMENT for table `location` |
|  | -- |
|  | ALTER TABLE `location` |
|  | MODIFY `location\_id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=25; |
|  | -- |
|  | -- AUTO\_INCREMENT for table `locationinventory` |
|  | -- |
|  | ALTER TABLE `locationinventory` |
|  | MODIFY `locationinventory\_id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=18; |
|  | -- |
|  | -- AUTO\_INCREMENT for table `product` |
|  | -- |
|  | ALTER TABLE `product` |
|  | MODIFY `product\_id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=18; |
|  | -- |
|  | -- AUTO\_INCREMENT for table `productmovement` |
|  | -- |
|  | ALTER TABLE `productmovement` |
|  | MODIFY `productmovement\_id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=8; |
|  | -- |
|  | -- AUTO\_INCREMENT for table `user` |
|  | -- |
|  | ALTER TABLE `user` |
|  | MODIFY `user\_id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=2; |
|  | /\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/; |
|  | /\*!40101 SET CHARACTER\_SET\_RESULTS=@OLD\_CHARACTER\_SET\_RESULTS \*/; |
|  | /\*!40101 SET COLLATION\_CONNECTION=@OLD\_COLLATION\_CONNECTION \*/; |

**RUN**

|  |
| --- |
| from inventorymanagement import app |
|  |  |
|  | if \_\_name\_\_=='\_\_main\_\_': |
|  | app.run(debug=True) |