# Fish Market App - Full Source Code

## app.py

from flask import Flask, render\_template, request, redirect, url\_for, session, flash  
import mysql.connector  
from werkzeug.security import generate\_password\_hash, check\_password\_hash  
from datetime import timedelta  
from config import DB\_CONFIG, SECRET\_KEY  
  
app = Flask(\_\_name\_\_)  
app.secret\_key = SECRET\_KEY  
app.permanent\_session\_lifetime = timedelta(days=7)  
  
def get\_db():  
 return mysql.connector.connect(\*\*DB\_CONFIG)  
  
# ----------------------  
# Helpers  
# ----------------------  
  
def current\_user():  
 return session.get("user")  
  
  
def login\_required(role=None):  
 def decorator(func):  
 def wrapper(\*args, \*\*kwargs):  
 u = current\_user()  
 if not u:  
 flash("Please log in.", "warning")  
 return redirect(url\_for("login"))  
 if role and u["role"] != role:  
 flash("Unauthorized.", "danger")  
 return redirect(url\_for("index"))  
 return func(\*args, \*\*kwargs)  
 wrapper.\_\_name\_\_ = func.\_\_name\_\_  
 return wrapper  
 return decorator  
  
# ----------------------  
# Auth  
# ----------------------  
  
@app.route("/register", methods=["GET", "POST"])  
def register():  
 if request.method == "POST":  
 name = request.form["name"].strip()  
 email = request.form["email"].strip().lower()  
 password = request.form["password"]  
 role = "user" # default  
 phone = request.form.get("phone", "").strip()  
 address = request.form.get("address", "").strip()  
  
 if not name or not email or not password:  
 flash("Name, email, and password are required.", "danger")  
 return redirect(url\_for("register"))  
  
 pw\_hash = generate\_password\_hash(password)  
  
 con = get\_db()  
 cur = con.cursor(dictionary=True)  
 try:  
 cur.execute(  
 """  
 INSERT INTO users(name,email,password\_hash,role,phone,address)  
 VALUES(%s,%s,%s,%s,%s,%s)  
 """,  
 (name, email, pw\_hash, role, phone, address),  
 )  
 con.commit()  
 flash("Registration successful. Please log in.", "success")  
 return redirect(url\_for("login"))  
 except mysql.connector.IntegrityError:  
 flash("Email already exists.", "danger")  
 finally:  
 cur.close()  
 con.close()  
  
 return render\_template("register.html")  
  
  
@app.route("/login", methods=["GET", "POST"])  
def login():  
 if request.method == "POST":  
 email = request.form["email"].strip().lower()  
 password = request.form["password"]  
  
 con = get\_db()  
 cur = con.cursor(dictionary=True)  
 cur.execute("SELECT \* FROM users WHERE email=%s", (email,))  
 user = cur.fetchone()  
 cur.close()  
 con.close()  
  
 if user and check\_password\_hash(user["password\_hash"], password):  
 session.permanent = True  
 session["user"] = {"id": user["id"], "name": user["name"], "role": user["role"]}  
 flash(f"Welcome, {user['name']}!", "success")  
 return redirect(url\_for("route\_by\_role"))  
 else:  
 flash("Invalid credentials.", "danger")  
  
 return render\_template("login.html")  
  
  
@app.route("/logout")  
def logout():  
 session.pop("user", None)  
 flash("Logged out.", "info")  
 return redirect(url\_for("index"))  
  
  
@app.route("/route")  
def route\_by\_role():  
 u = current\_user()  
 if not u:  
 return redirect(url\_for("login"))  
 role = u["role"]  
 if role == "admin":  
 return redirect(url\_for("admin\_dashboard"))  
 if role == "harbour":  
 return redirect(url\_for("harbour\_dashboard"))  
 if role == "delivery":  
 return redirect(url\_for("delivery\_dashboard"))  
 return redirect(url\_for("user\_dashboard"))  
  
# ----------------------  
# General pages  
# ----------------------  
  
@app.route("/")  
def index():  
 return render\_template("index.html", u=current\_user())  
  
  
@app.route("/profile", methods=["GET", "POST"])  
def profile():  
 u = current\_user()  
 if not u:  
 return redirect(url\_for("login"))  
  
 con = get\_db()  
 cur = con.cursor(dictionary=True)  
  
 if request.method == "POST":  
 name = request.form["name"]  
 phone = request.form.get("phone", "")  
 address = request.form.get("address", "")  
 cur.execute(  
 "UPDATE users SET name=%s, phone=%s, address=%s WHERE id=%s",  
 (name, phone, address, u["id"]),  
 )  
 con.commit()  
 session["user"]["name"] = name  
 flash("Profile updated.", "success")  
  
 cur.execute("SELECT \* FROM users WHERE id=%s", (u["id"],))  
 user = cur.fetchone()  
 cur.close()  
 con.close()  
 return render\_template("profile.html", user=user)  
  
# ----------------------  
# Admin  
# ----------------------  
  
@app.route("/admin")  
@login\_required(role="admin")  
def admin\_dashboard():  
 con = get\_db()  
 cur = con.cursor(dictionary=True)  
 cur.execute("SELECT COUNT(\*) as c FROM users WHERE role='user'")  
 users\_count = cur.fetchone()["c"]  
 cur.execute("SELECT COUNT(\*) as c FROM harbours")  
 harbours\_count = cur.fetchone()["c"]  
 cur.execute("SELECT COUNT(\*) as c FROM fish")  
 fish\_count = cur.fetchone()["c"]  
 cur.execute("SELECT COUNT(\*) as c FROM orders")  
 orders\_count = cur.fetchone()["c"]  
 cur.close()  
 con.close()  
 return render\_template(  
 "admin/dashboard.html",  
 users\_count=users\_count,  
 harbours\_count=harbours\_count,  
 fish\_count=fish\_count,  
 orders\_count=orders\_count,  
 )  
  
  
@app.route("/admin/users")  
@login\_required(role="admin")  
def admin\_users():  
 con = get\_db()  
 cur = con.cursor(dictionary=True)  
 cur.execute("SELECT id,name,email,role,phone,address FROM users ORDER BY id DESC")  
 users = cur.fetchall()  
 cur.close()  
 con.close()  
 return render\_template("admin/users.html", users=users)  
  
  
@app.route("/admin/harbours", methods=["GET", "POST"])  
@login\_required(role="admin")  
def admin\_harbours():  
 con = get\_db()  
 cur = con.cursor(dictionary=True)  
 if request.method == "POST":  
 name = request.form["name"]  
 location = request.form["location"]  
 contact = request.form.get("contact", "")  
 owner\_id = request.form["owner\_id"]  
 cur.execute(  
 "INSERT INTO harbours(name,location,contact,user\_id) VALUES(%s,%s,%s,%s)",  
 (name, location, contact, owner\_id),  
 )  
 con.commit()  
 flash("Harbour added.", "success")  
 cur.execute(  
 """  
 SELECT h.\*, u.name as owner\_name FROM harbours h  
 JOIN users u ON u.id = h.user\_id  
 ORDER BY h.id DESC  
 """  
 )  
 harbours = cur.fetchall()  
 cur.execute("SELECT id,name FROM users WHERE role='harbour'")  
 owners = cur.fetchall()  
 cur.close()  
 con.close()  
 return render\_template("admin/harbours.html", harbours=harbours, owners=owners)  
  
  
@app.route("/admin/boats", methods=["GET", "POST"])  
@login\_required(role="admin")  
def admin\_boats():  
 con = get\_db()  
 cur = con.cursor(dictionary=True)  
 if request.method == "POST":  
 name = request.form["name"]  
 capacity = int(request.form.get("capacity", 0))  
 harbour\_id = request.form["harbour\_id"]  
 cur.execute(  
 "INSERT INTO boats(name,capacity,harbour\_id) VALUES(%s,%s,%s)",  
 (name, capacity, harbour\_id),  
 )  
 con.commit()  
 flash("Boat added.", "success")  
 cur.execute(  
 """  
 SELECT b.\*, h.name as harbour\_name FROM boats b  
 JOIN harbours h ON h.id=b.harbour\_id ORDER BY b.id DESC  
 """  
 )  
 boats = cur.fetchall()  
 cur.execute("SELECT id,name FROM harbours")  
 harbour\_list = cur.fetchall()  
 cur.close()  
 con.close()  
 return render\_template("admin/boats.html", boats=boats, harbour\_list=harbour\_list)  
  
  
@app.route("/admin/fish", methods=["GET", "POST"])  
@login\_required(role="admin")  
def admin\_fish():  
 con = get\_db()  
 cur = con.cursor(dictionary=True)  
 if request.method == "POST":  
 name = request.form["name"]  
 price = float(request.form["price"])  
 stock = int(request.form["stock"])  
 harbour\_id = request.form["harbour\_id"]  
 cur.execute(  
 "INSERT INTO fish(name,price,stock,harbour\_id) VALUES(%s,%s,%s,%s)",  
 (name, price, stock, harbour\_id),  
 )  
 con.commit()  
 flash("Fish added.", "success")  
 cur.execute(  
 """  
 SELECT f.\*, h.name as harbour\_name FROM fish f  
 JOIN harbours h ON h.id=f.harbour\_id ORDER BY f.id DESC  
 """  
 )  
 fish = cur.fetchall()  
 cur.execute("SELECT id,name FROM harbours")  
 harbours = cur.fetchall()  
 cur.close()  
 con.close()  
 return render\_template("admin/fish.html", fish=fish, harbours=harbours)  
  
  
@app.route("/admin/feedback")  
@login\_required(role="admin")  
def admin\_feedback():  
 con = get\_db()  
 cur = con.cursor(dictionary=True)  
 cur.execute(  
 """  
 SELECT f.\*, u.name as user\_name FROM feedback f  
 JOIN users u ON u.id=f.user\_id ORDER BY f.id DESC  
 """  
 )  
 items = cur.fetchall()  
 cur.close()  
 con.close()  
 return render\_template("admin/feedback.html", items=items)  
  
  
@app.route("/admin/complaints", methods=["GET", "POST"])  
@login\_required(role="admin")  
def admin\_complaints():  
 con = get\_db()  
 cur = con.cursor(dictionary=True)  
 if request.method == "POST":  
 cid = request.form["complaint\_id"]  
 reply = request.form["reply"]  
 cur.execute("UPDATE complaints SET reply=%s WHERE id=%s", (reply, cid))  
 con.commit()  
 flash("Reply sent.", "success")  
 cur.execute(  
 """  
 SELECT c.\*, u.name as user\_name, u.email FROM complaints c  
 JOIN users u ON u.id=c.user\_id ORDER BY c.id DESC  
 """  
 )  
 items = cur.fetchall()  
 cur.close()  
 con.close()  
 return render\_template("admin/complaints.html", items=items)  
  
# ----------------------  
# Harbour  
# ----------------------  
  
@app.route("/harbour")  
@login\_required(role="harbour")  
def harbour\_dashboard():  
 u = current\_user()  
 con = get\_db()  
 cur = con.cursor(dictionary=True)  
 cur.execute("SELECT \* FROM harbours WHERE user\_id=%s", (u["id"],))  
 harbour = cur.fetchone()  
 harbour\_id = harbour["id"] if harbour else None  
 cur.execute("SELECT COUNT(\*) c FROM orders WHERE harbour\_id=%s", (harbour\_id,))  
 orders\_count = cur.fetchone()["c"] if harbour\_id else 0  
 cur.execute("SELECT COUNT(\*) c FROM fish WHERE harbour\_id=%s", (harbour\_id,))  
 fish\_count = cur.fetchone()["c"] if harbour\_id else 0  
 cur.execute("SELECT COUNT(\*) c FROM boats WHERE harbour\_id=%s", (harbour\_id,))  
 boats\_count = cur.fetchone()["c"] if harbour\_id else 0  
 cur.close()  
 con.close()  
 return render\_template(  
 "harbour/dashboard.html",  
 harbour=harbour,  
 orders\_count=orders\_count,  
 fish\_count=fish\_count,  
 boats\_count=boats\_count,  
 )  
  
  
@app.route("/harbour/stock", methods=["GET", "POST"])  
@login\_required(role="harbour")  
def harbour\_stock():  
 u = current\_user()  
 con = get\_db()  
 cur = con.cursor(dictionary=True)  
 cur.execute("SELECT \* FROM harbours WHERE user\_id=%s", (u["id"],))  
 harbour = cur.fetchone()  
 if not harbour:  
 flash("No harbour profile yet. Ask admin to create one.", "warning")  
 return redirect(url\_for("harbour\_dashboard"))  
 if request.method == "POST":  
 fid = request.form.get("fish\_id")  
 stock = int(request.form.get("stock", "0"))  
 price = float(request.form.get("price", "0"))  
 cur.execute(  
 "UPDATE fish SET stock=%s, price=%s WHERE id=%s AND harbour\_id=%s",  
 (stock, price, fid, harbour["id"]),  
 )  
 con.commit()  
 flash("Stock updated.", "success")  
 cur.execute("SELECT \* FROM fish WHERE harbour\_id=%s ORDER BY id DESC", (harbour["id"],))  
 fish = cur.fetchall()  
 cur.close()  
 con.close()  
 return render\_template("harbour/stock.html", fish=fish)  
  
  
@app.route("/harbour/orders", methods=["GET", "POST"])  
@login\_required(role="harbour")  
def harbour\_orders():  
 u = current\_user()  
 con = get\_db()  
 cur = con.cursor(dictionary=True)  
 cur.execute("SELECT \* FROM harbours WHERE user\_id=%s", (u["id"],))  
 harbour = cur.fetchone()  
 if not harbour:  
 flash("No harbour profile.", "warning")  
 return redirect(url\_for("harbour\_dashboard"))  
 if request.method == "POST":  
 order\_id = request.form["order\_id"]  
 delivery\_user\_id = request.form["delivery\_user\_id"]  
 cur.execute("SELECT id FROM deliveries WHERE order\_id=%s", (order\_id,))  
 d = cur.fetchone()  
 if d:  
 cur.execute(  
 "UPDATE deliveries SET delivery\_user\_id=%s, status='assigned' WHERE order\_id=%s",  
 (delivery\_user\_id, order\_id),  
 )  
 else:  
 cur.execute(  
 "INSERT INTO deliveries(order\_id, delivery\_user\_id, status) VALUES(%s,%s,'assigned')",  
 (order\_id, delivery\_user\_id),  
 )  
 cur.execute("UPDATE orders SET status='assigned' WHERE id=%s", (order\_id,))  
 con.commit()  
 flash("Order assigned to delivery.", "success")  
 cur.execute(  
 """  
 SELECT o.\*, u.name as user\_name FROM orders o  
 JOIN users u ON u.id=o.user\_id  
 WHERE o.harbour\_id=%s ORDER BY o.id DESC  
 """,  
 (harbour["id"],),  
 )  
 orders = cur.fetchall()  
 cur.execute("SELECT id,name FROM users WHERE role='delivery'")  
 deliveries = cur.fetchall()  
 cur.close()  
 con.close()  
 return render\_template("harbour/orders.html", orders=orders, deliveries=deliveries)  
  
# ----------------------  
# User (Customer)  
# ----------------------  
  
@app.route("/user")  
@login\_required(role="user")  
def user\_dashboard():  
 con = get\_db()  
 cur = con.cursor(dictionary=True)  
 cur.execute("SELECT \* FROM harbours ORDER BY id DESC")  
 harbours = cur.fetchall()  
 cur.close()  
 con.close()  
 return render\_template("user/dashboard.html", harbours=harbours)  
  
  
@app.route("/harbours/<int:harbour\_id>/fish")  
def view\_fish(harbour\_id):  
 con = get\_db()  
 cur = con.cursor(dictionary=True)  
 cur.execute("SELECT \* FROM harbours WHERE id=%s", (harbour\_id,))  
 harbour = cur.fetchone()  
 cur.execute("SELECT \* FROM fish WHERE harbour\_id=%s ORDER BY id DESC", (harbour\_id,))  
 fish = cur.fetchall()  
 cur.close()  
 con.close()  
 return render\_template("user/fish.html", harbour=harbour, fish=fish)  
  
  
@app.route("/cart/add", methods=["POST"])  
@login\_required(role="user")  
def add\_to\_cart():  
 fid = int(request.form["fish\_id"])  
 qty = int(request.form["quantity"])  
 clean = request.form.get("clean") == "yes"  
  
 cart = session.get("cart", {})  
  
 # initialize OR convert old int format to dict format  
 if str(fid) not in cart or isinstance(cart[str(fid)], int):  
 cart[str(fid)] = {"qty": 0, "clean": False}  
  
 cart[str(fid)]["qty"] += qty  
  
 if clean:  
 cart[str(fid)]["clean"] = True  
  
 session["cart"] = cart  
 flash("Added to cart.", "success")  
 return redirect(request.referrer or url\_for("user\_dashboard"))  
  
  
@app.route("/cart")  
@login\_required(role="user")  
def view\_cart():  
 cart = session.get("cart", {})  
 if not cart:  
 return render\_template("user/cart.html", items=[], total=0)  
 ids = list(cart.keys())  
 placeholders = ",".join(["%s"] \* len(ids))  
 con = get\_db()  
 cur = con.cursor(dictionary=True)  
 cur.execute(f"SELECT \* FROM fish WHERE id IN ({placeholders})", ids)  
 fish\_rows = cur.fetchall()  
 items = []  
 total = 0  
 for f in fish\_rows:  
 entry = cart[str(f["id"])]  
 qty = entry["qty"]  
 amount = f["price"] \* qty  
 if entry.get("clean"): # add cleaning charge  
 amount += 100  
 total += amount  
 items.append({"fish": f, "qty": qty, "clean": entry.get("clean"), "amount": amount})  
 cur.close()  
 con.close()  
 return render\_template("user/cart.html", items=items, total=total)  
  
  
@app.route("/cart/checkout", methods=["POST"])  
@login\_required(role="user")  
def checkout():  
 cart = session.get("cart", {})  
 if not cart:  
 flash("Cart is empty.", "warning")  
 return redirect(url\_for("view\_cart"))  
 ids = list(cart.keys())  
 placeholders = ",".join(["%s"] \* len(ids))  
 con = get\_db()  
 cur = con.cursor(dictionary=True)  
 cur.execute(f"SELECT \* FROM fish WHERE id IN ({placeholders})", ids)  
 fish\_rows = cur.fetchall()  
 if not fish\_rows:  
 flash("Invalid cart.", "danger")  
 return redirect(url\_for("view\_cart"))  
 harbour\_id = fish\_rows[0]["harbour\_id"]  
 total = 0  
 for f in fish\_rows:  
 qty = cart[str(f["id"])]  
 if qty > f["stock"]:  
 flash(f"Not enough stock for {f['name']}.", "danger")  
 cur.close()  
 con.close()  
 return redirect(url\_for("view\_cart"))  
 total += f["price"] \* qty  
 cur.execute(  
 "INSERT INTO orders(user\_id, harbour\_id, status, total\_amount) VALUES(%s,%s,'pending',%s)",  
 (current\_user()["id"], harbour\_id, total),  
 )  
 order\_id = cur.lastrowid  
 for f in fish\_rows:  
 qty = cart[str(f["id"])]  
 cur.execute(  
 "INSERT INTO order\_items(order\_id, fish\_id, quantity, price) VALUES(%s,%s,%s,%s)",  
 (order\_id, f["id"], qty, f["price"]),  
 )  
 cur.execute("UPDATE fish SET stock=stock-%s WHERE id=%s", (qty, f["id"]))  
 con.commit()  
 session["cart"] = {}  
 flash(f"Order #{order\_id} placed successfully! (mock payment).", "success")  
 cur.close()  
 con.close()  
 return redirect(url\_for("user\_orders"))  
  
  
@app.route("/user/orders")  
@login\_required(role="user")  
def user\_orders():  
 con = get\_db()  
 cur = con.cursor(dictionary=True)  
 cur.execute(  
 """  
 SELECT o.\*, h.name as harbour\_name, d.status as delivery\_status  
 FROM orders o  
 JOIN harbours h ON h.id=o.harbour\_id  
 LEFT JOIN deliveries d ON d.order\_id = o.id  
 WHERE o.user\_id=%s  
 ORDER BY o.id DESC  
 """,  
 (current\_user()["id"],),  
 )  
 orders = cur.fetchall()  
 cur.close()  
 con.close()  
 return render\_template("user/orders.html", orders=orders)  
  
  
@app.route("/complaint", methods=["GET", "POST"])  
@login\_required(role="user")  
def complaint():  
 con = get\_db()  
 cur = con.cursor(dictionary=True)  
 if request.method == "POST":  
 msg = request.form["message"]  
 cur.execute("INSERT INTO complaints(user\_id, message) VALUES(%s,%s)", (current\_user()["id"], msg))  
 con.commit()  
 flash("Complaint submitted.", "success")  
 cur.execute("SELECT \* FROM complaints WHERE user\_id=%s ORDER BY id DESC", (current\_user()["id"],))  
 items = cur.fetchall()  
 cur.close()  
 con.close()  
 return render\_template("user/complaints.html", items=items)  
  
  
@app.route("/feedback", methods=["GET", "POST"])  
@login\_required(role="user")  
def feedback():  
 con = get\_db()  
 cur = con.cursor(dictionary=True)  
 if request.method == "POST":  
 rating = int(request.form["rating"])  
 message = request.form.get("message", "")  
 cur.execute("INSERT INTO feedback(user\_id, rating, message) VALUES(%s,%s,%s)", (current\_user()["id"], rating, message))  
 con.commit()  
 flash("Feedback sent. Thanks!", "success")  
 cur.execute("SELECT \* FROM feedback WHERE user\_id=%s ORDER BY id DESC", (current\_user()["id"],))  
 items = cur.fetchall()  
 cur.close()  
 con.close()  
 return render\_template("user/feedback.html", items=items)  
  
# ----------------------  
# Delivery  
# ----------------------  
  
@app.route("/delivery")  
@login\_required(role="delivery")  
def delivery\_dashboard():  
 con = get\_db()  
 cur = con.cursor(dictionary=True)  
 cur.execute(  
 """  
 SELECT d.\*, o.total\_amount, o.status as order\_status,  
 u.name as customer\_name, u.phone as customer\_phone, u.address as customer\_address  
 FROM deliveries d  
 JOIN orders o ON o.id=d.order\_id  
 JOIN users u ON u.id=o.user\_id  
 WHERE d.delivery\_user\_id=%s  
 ORDER BY d.id DESC  
 """,  
 (current\_user()["id"],),  
 )  
 items = cur.fetchall()  
 cur.close()  
 con.close()  
 return render\_template("delivery/dashboard.html", items=items)  
  
  
@app.route("/delivery/update", methods=["POST"])  
@login\_required(role="delivery")  
def delivery\_update():  
 delivery\_id = request.form["delivery\_id"]  
 status = request.form["status"]  
 con = get\_db()  
 cur = con.cursor(dictionary=True)  
 cur.execute("UPDATE deliveries SET status=%s WHERE id=%s", (status, delivery\_id))  
 if status == "delivered":  
 cur.execute(  
 """  
 UPDATE orders o  
 JOIN deliveries d ON d.order\_id=o.id  
 SET o.status='delivered'  
 WHERE d.id=%s  
 """,  
 (delivery\_id,),  
 )  
 con.commit()  
 cur.close()  
 con.close()  
 flash("Status updated.", "success")  
 return redirect(url\_for("delivery\_dashboard"))  
  
# ----------------------  
# Utilities  
# ----------------------  
  
@app.context\_processor  
def inject\_globals():  
 return {"u": current\_user()}  
  
  
@app.template\_filter("inr")  
def inr(amount):  
 return f"₹{amount:,.2f}"  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 app.run(debug=True)

## config.py

import os  
  
# Update these for your MySQL setup  
DB\_CONFIG = {  
 "host": os.environ.get("DB\_HOST", "localhost"),  
 "user": os.environ.get("DB\_USER", "root"),  
 "password": os.environ.get("DB\_PASSWORD", "MyNewPass"),  
 "database": os.environ.get("DB\_NAME", "fish\_market\_db"),  
 "port": int(os.environ.get("DB\_PORT", "3306")),  
}  
  
SECRET\_KEY = os.environ.get("SECRET\_KEY", "112255")

## export\_to\_docx.py

import os  
from docx import Document  
from docx.shared import Pt  
from docx.oxml.ns import qn  
  
project\_root = "." # current folder  
output\_file = "fish\_market\_app\_code.docx"  
  
doc = Document()  
doc.add\_heading("Fish Market App - Full Source Code", level=1)  
  
for root, dirs, files in os.walk(project\_root):  
 for file in files:  
 if file.endswith((".py", ".html", ".css", ".js", ".sql", ".txt")):  
 filepath = os.path.join(root, file)  
 relpath = os.path.relpath(filepath, project\_root)  
  
 doc.add\_heading(relpath, level=2)  
 with open(filepath, "r", encoding="utf-8", errors="ignore") as f:  
 code = f.read()  
  
 paragraph = doc.add\_paragraph()  
 run = paragraph.add\_run(code)  
 run.font.name = 'Courier New'  
 run.font.size = Pt(10)  
 # Make sure font applies in Word (especially for non-Latin systems)  
 r = run.\_element.rPr.rFonts  
 r.set(qn('w:eastAsia'), 'Courier New')  
  
doc.save(output\_file)  
print("✅ Export complete:", output\_file)

## fish\_market\_db.sql

-- MySQL dump 10.13 Distrib 8.0.43, for Win64 (x86\_64)  
--  
-- Host: localhost Database: fish\_market\_db  
-- ------------------------------------------------------  
-- Server version 8.0.43  
  
/\*!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 \*/;  
/\*!50503 SET NAMES utf8mb4 \*/;  
/\*!40103 SET @OLD\_TIME\_ZONE=@@TIME\_ZONE \*/;  
/\*!40103 SET TIME\_ZONE='+00:00' \*/;  
/\*!40014 SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0 \*/;  
/\*!40014 SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0 \*/;  
/\*!40101 SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='NO\_AUTO\_VALUE\_ON\_ZERO' \*/;  
/\*!40111 SET @OLD\_SQL\_NOTES=@@SQL\_NOTES, SQL\_NOTES=0 \*/;  
  
--  
-- Table structure for table `boats`  
--  
  
DROP TABLE IF EXISTS `boats`;  
/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;  
/\*!50503 SET character\_set\_client = utf8mb4 \*/;  
CREATE TABLE `boats` (  
 `id` int NOT NULL AUTO\_INCREMENT,  
 `name` varchar(120) NOT NULL,  
 `capacity` int DEFAULT '0',  
 `harbour\_id` int NOT NULL,  
 PRIMARY KEY (`id`),  
 KEY `harbour\_id` (`harbour\_id`),  
 CONSTRAINT `boats\_ibfk\_1` FOREIGN KEY (`harbour\_id`) REFERENCES `harbours` (`id`) ON DELETE CASCADE  
) ENGINE=InnoDB AUTO\_INCREMENT=3 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;  
/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;  
  
--  
-- Dumping data for table `boats`  
--  
  
LOCK TABLES `boats` WRITE;  
/\*!40000 ALTER TABLE `boats` DISABLE KEYS \*/;  
INSERT INTO `boats` VALUES (1,'Sea Star',500,1),(2,'Ocean Wind',300,1);  
/\*!40000 ALTER TABLE `boats` ENABLE KEYS \*/;  
UNLOCK TABLES;  
  
--  
-- Table structure for table `complaints`  
--  
  
DROP TABLE IF EXISTS `complaints`;  
/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;  
/\*!50503 SET character\_set\_client = utf8mb4 \*/;  
CREATE TABLE `complaints` (  
 `id` int NOT NULL AUTO\_INCREMENT,  
 `user\_id` int NOT NULL,  
 `message` text NOT NULL,  
 `reply` text,  
 `created\_at` timestamp NULL DEFAULT CURRENT\_TIMESTAMP,  
 PRIMARY KEY (`id`),  
 KEY `user\_id` (`user\_id`),  
 CONSTRAINT `complaints\_ibfk\_1` FOREIGN KEY (`user\_id`) REFERENCES `users` (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;  
/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;  
  
--  
-- Dumping data for table `complaints`  
--  
  
LOCK TABLES `complaints` WRITE;  
/\*!40000 ALTER TABLE `complaints` DISABLE KEYS \*/;  
/\*!40000 ALTER TABLE `complaints` ENABLE KEYS \*/;  
UNLOCK TABLES;  
  
--  
-- Table structure for table `deliveries`  
--  
  
DROP TABLE IF EXISTS `deliveries`;  
/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;  
/\*!50503 SET character\_set\_client = utf8mb4 \*/;  
CREATE TABLE `deliveries` (  
 `id` int NOT NULL AUTO\_INCREMENT,  
 `order\_id` int NOT NULL,  
 `delivery\_user\_id` int NOT NULL,  
 `status` enum('assigned','picked\_up','out\_for\_delivery','delivered') DEFAULT 'assigned',  
 `updated\_at` timestamp NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,  
 PRIMARY KEY (`id`),  
 KEY `order\_id` (`order\_id`),  
 KEY `delivery\_user\_id` (`delivery\_user\_id`),  
 CONSTRAINT `deliveries\_ibfk\_1` FOREIGN KEY (`order\_id`) REFERENCES `orders` (`id`) ON DELETE CASCADE,  
 CONSTRAINT `deliveries\_ibfk\_2` FOREIGN KEY (`delivery\_user\_id`) REFERENCES `users` (`id`)  
) ENGINE=InnoDB AUTO\_INCREMENT=4 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;  
/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;  
  
--  
-- Dumping data for table `deliveries`  
--  
  
LOCK TABLES `deliveries` WRITE;  
/\*!40000 ALTER TABLE `deliveries` DISABLE KEYS \*/;  
INSERT INTO `deliveries` VALUES (1,2,4,'assigned','2025-09-10 14:16:43'),(2,1,4,'out\_for\_delivery','2025-09-10 14:17:29'),(3,3,4,'delivered','2025-09-10 15:13:08');  
/\*!40000 ALTER TABLE `deliveries` ENABLE KEYS \*/;  
UNLOCK TABLES;  
  
--  
-- Table structure for table `feedback`  
--  
  
DROP TABLE IF EXISTS `feedback`;  
/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;  
/\*!50503 SET character\_set\_client = utf8mb4 \*/;  
CREATE TABLE `feedback` (  
 `id` int NOT NULL AUTO\_INCREMENT,  
 `user\_id` int NOT NULL,  
 `rating` int DEFAULT NULL,  
 `message` text,  
 `created\_at` timestamp NULL DEFAULT CURRENT\_TIMESTAMP,  
 PRIMARY KEY (`id`),  
 KEY `user\_id` (`user\_id`),  
 CONSTRAINT `feedback\_ibfk\_1` FOREIGN KEY (`user\_id`) REFERENCES `users` (`id`),  
 CONSTRAINT `feedback\_chk\_1` CHECK ((`rating` between 1 and 5))  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;  
/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;  
  
--  
-- Dumping data for table `feedback`  
--  
  
LOCK TABLES `feedback` WRITE;  
/\*!40000 ALTER TABLE `feedback` DISABLE KEYS \*/;  
/\*!40000 ALTER TABLE `feedback` ENABLE KEYS \*/;  
UNLOCK TABLES;  
  
--  
-- Table structure for table `fish`  
--  
  
DROP TABLE IF EXISTS `fish`;  
/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;  
/\*!50503 SET character\_set\_client = utf8mb4 \*/;  
CREATE TABLE `fish` (  
 `id` int NOT NULL AUTO\_INCREMENT,  
 `name` varchar(120) NOT NULL,  
 `price` decimal(10,2) NOT NULL,  
 `stock` int NOT NULL DEFAULT '0',  
 `harbour\_id` int NOT NULL,  
 PRIMARY KEY (`id`),  
 KEY `harbour\_id` (`harbour\_id`),  
 CONSTRAINT `fish\_ibfk\_1` FOREIGN KEY (`harbour\_id`) REFERENCES `harbours` (`id`) ON DELETE CASCADE  
) ENGINE=InnoDB AUTO\_INCREMENT=4 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;  
/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;  
  
--  
-- Dumping data for table `fish`  
--  
  
LOCK TABLES `fish` WRITE;  
/\*!40000 ALTER TABLE `fish` DISABLE KEYS \*/;  
INSERT INTO `fish` VALUES (1,'Salmon',600.00,100,1),(2,'Tuna',450.00,75,1),(3,'Mackerel',200.00,143,1);  
/\*!40000 ALTER TABLE `fish` ENABLE KEYS \*/;  
UNLOCK TABLES;  
  
--  
-- Table structure for table `harbours`  
--  
  
DROP TABLE IF EXISTS `harbours`;  
/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;  
/\*!50503 SET character\_set\_client = utf8mb4 \*/;  
CREATE TABLE `harbours` (  
 `id` int NOT NULL AUTO\_INCREMENT,  
 `name` varchar(120) NOT NULL,  
 `location` varchar(255) NOT NULL,  
 `contact` varchar(100) DEFAULT NULL,  
 `user\_id` int NOT NULL,  
 PRIMARY KEY (`id`),  
 KEY `user\_id` (`user\_id`),  
 CONSTRAINT `harbours\_ibfk\_1` FOREIGN KEY (`user\_id`) REFERENCES `users` (`id`) ON DELETE CASCADE  
) ENGINE=InnoDB AUTO\_INCREMENT=2 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;  
/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;  
  
--  
-- Dumping data for table `harbours`  
--  
  
LOCK TABLES `harbours` WRITE;  
/\*!40000 ALTER TABLE `harbours` DISABLE KEYS \*/;  
INSERT INTO `harbours` VALUES (1,'Blue Bay Harbour','Blue Bay','555-1234',2);  
/\*!40000 ALTER TABLE `harbours` ENABLE KEYS \*/;  
UNLOCK TABLES;  
  
--  
-- Table structure for table `order\_items`  
--  
  
DROP TABLE IF EXISTS `order\_items`;  
/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;  
/\*!50503 SET character\_set\_client = utf8mb4 \*/;  
CREATE TABLE `order\_items` (  
 `id` int NOT NULL AUTO\_INCREMENT,  
 `order\_id` int NOT NULL,  
 `fish\_id` int NOT NULL,  
 `quantity` int NOT NULL,  
 `price` decimal(10,2) NOT NULL,  
 PRIMARY KEY (`id`),  
 KEY `order\_id` (`order\_id`),  
 KEY `fish\_id` (`fish\_id`),  
 CONSTRAINT `order\_items\_ibfk\_1` FOREIGN KEY (`order\_id`) REFERENCES `orders` (`id`) ON DELETE CASCADE,  
 CONSTRAINT `order\_items\_ibfk\_2` FOREIGN KEY (`fish\_id`) REFERENCES `fish` (`id`)  
) ENGINE=InnoDB AUTO\_INCREMENT=7 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;  
/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;  
  
--  
-- Dumping data for table `order\_items`  
--  
  
LOCK TABLES `order\_items` WRITE;  
/\*!40000 ALTER TABLE `order\_items` DISABLE KEYS \*/;  
INSERT INTO `order\_items` VALUES (1,1,3,3,200.00),(2,2,2,2,450.00),(3,2,3,1,200.00),(4,3,2,1,450.00),(5,3,3,3,200.00),(6,4,2,2,450.00);  
/\*!40000 ALTER TABLE `order\_items` ENABLE KEYS \*/;  
UNLOCK TABLES;  
  
--  
-- Table structure for table `orders`  
--  
  
DROP TABLE IF EXISTS `orders`;  
/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;  
/\*!50503 SET character\_set\_client = utf8mb4 \*/;  
CREATE TABLE `orders` (  
 `id` int NOT NULL AUTO\_INCREMENT,  
 `user\_id` int NOT NULL,  
 `harbour\_id` int NOT NULL,  
 `status` enum('pending','assigned','out\_for\_delivery','delivered','cancelled') DEFAULT 'pending',  
 `total\_amount` decimal(10,2) DEFAULT '0.00',  
 `created\_at` timestamp NULL DEFAULT CURRENT\_TIMESTAMP,  
 PRIMARY KEY (`id`),  
 KEY `user\_id` (`user\_id`),  
 KEY `harbour\_id` (`harbour\_id`),  
 CONSTRAINT `orders\_ibfk\_1` FOREIGN KEY (`user\_id`) REFERENCES `users` (`id`),  
 CONSTRAINT `orders\_ibfk\_2` FOREIGN KEY (`harbour\_id`) REFERENCES `harbours` (`id`)  
) ENGINE=InnoDB AUTO\_INCREMENT=5 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;  
/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;  
  
--  
-- Dumping data for table `orders`  
--  
  
LOCK TABLES `orders` WRITE;  
/\*!40000 ALTER TABLE `orders` DISABLE KEYS \*/;  
INSERT INTO `orders` VALUES (1,7,1,'assigned',600.00,'2025-09-10 13:47:01'),(2,7,1,'assigned',1100.00,'2025-09-10 14:15:59'),(3,8,1,'delivered',1050.00,'2025-09-10 14:32:45'),(4,8,1,'pending',900.00,'2025-09-10 14:52:50');  
/\*!40000 ALTER TABLE `orders` ENABLE KEYS \*/;  
UNLOCK TABLES;  
  
--  
-- Table structure for table `users`  
--  
  
DROP TABLE IF EXISTS `users`;  
/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;  
/\*!50503 SET character\_set\_client = utf8mb4 \*/;  
CREATE TABLE `users` (  
 `id` int NOT NULL AUTO\_INCREMENT,  
 `name` varchar(100) NOT NULL,  
 `email` varchar(120) NOT NULL,  
 `password\_hash` varchar(255) NOT NULL,  
 `role` enum('admin','harbour','user','delivery') NOT NULL,  
 `phone` varchar(20) DEFAULT NULL,  
 `address` varchar(255) DEFAULT NULL,  
 `created\_at` timestamp NULL DEFAULT CURRENT\_TIMESTAMP,  
 PRIMARY KEY (`id`),  
 UNIQUE KEY `email` (`email`)  
) ENGINE=InnoDB AUTO\_INCREMENT=9 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;  
/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;  
  
--  
-- Dumping data for table `users`  
--  
  
LOCK TABLES `users` WRITE;  
/\*!40000 ALTER TABLE `users` DISABLE KEYS \*/;  
INSERT INTO `users` VALUES (1,'Admin','admin@example.com','scrypt:32768:8:1$JJhVIzolG3EBbOTj$8e3d877df22a5e52419f40dbea5c53bf4db32ed272331dcd0b3db4b6d05a6f72e80cf0ce87d0e9d1c468c1232fac48a8649a9613529de3df44e16ec08a86a5b8','admin','0000000000','HQ','2025-09-01 11:36:10'),(2,'Harbour Owner','harbour@example.com','scrypt:32768:8:1$JJhVIzolG3EBbOTj$8e3d877df22a5e52419f40dbea5c53bf4db32ed272331dcd0b3db4b6d05a6f72e80cf0ce87d0e9d1c468c1232fac48a8649a9613529de3df44e16ec08a86a5b8','harbour','1111111111','Coast Road','2025-09-01 11:36:10'),(3,'Alice','alice@example.com','scrypt:32768:8:1$JJhVIzolG3EBbOTj$8e3d877df22a5e52419f40dbea5c53bf4db32ed272331dcd0b3db4b6d05a6f72e80cf0ce87d0e9d1c468c1232fac48a8649a9613529de3df44e16ec08a86a5b8','user','2222222222','City Center','2025-09-01 11:36:10'),(4,'Delivery Guy','delivery@example.com','scrypt:32768:8:1$JJhVIzolG3EBbOTj$8e3d877df22a5e52419f40dbea5c53bf4db32ed272331dcd0b3db4b6d05a6f72e80cf0ce87d0e9d1c468c1232fac48a8649a9613529de3df44e16ec08a86a5b8','delivery','3333333333','Depot','2025-09-01 11:36:10'),(5,'Rahul','rahulsplind@gmail.com','scrypt:32768:8:1$ivx3RAZ2zA1lgmin$e414ac5f9af2ef0dfb1223cc288eb45e28e929977cb6b9a9e8d0fbf0df3c9ff3c7466577fe1de2e4b992bda0feab407d596623c835912cdc01ad64a20d45389d','user','11111111111','hhhh','2025-09-10 13:44:16'),(7,'anjana','ana@gmail.com','scrypt:32768:8:1$9wo7eGLP3OIi0cDY$de968f617c70bbf0e6bed4d1b497b057aedae64c631247a8ce12bfa51b907891f190bab6187b9a3c1f3ed09fc6e783c42e0552670648649d4dbb4dc1431e7cf4','user','8899663322','kkllkklk','2025-09-10 13:46:34'),(8,'ANJU','anjuanilkumar292@gmail.com','scrypt:32768:8:1$rOhMRulbcVuquPTk$e96a602d4d82113c995c02b835c4c3019692b9d9e5f57f0a32bf4c206a431c58d4e1ada3ea1cc5aa56096336c3cbd8ff1cec9dc98f761df4e673dcbdb3ef39d1','user','1234567891','ponnani','2025-09-10 14:31:00');  
/\*!40000 ALTER TABLE `users` ENABLE KEYS \*/;  
UNLOCK TABLES;  
/\*!40103 SET TIME\_ZONE=@OLD\_TIME\_ZONE \*/;  
  
/\*!40101 SET SQL\_MODE=@OLD\_SQL\_MODE \*/;  
/\*!40014 SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS \*/;  
/\*!40014 SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS \*/;  
/\*!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 \*/;  
/\*!40111 SET SQL\_NOTES=@OLD\_SQL\_NOTES \*/;  
  
-- Dump completed on 2025-09-10 22:15:24

## from werkzeug.py

from werkzeug.security import generate\_password\_hash  
  
# Replace this with the real password you want  
plain\_password = "1234"  
  
# Generate a secure hash  
hashed\_password = generate\_password\_hash(plain\_password)  
  
print(hashed\_password)

## requirements.txt

flask==3.0.3  
mysql-connector-python==9.0.0  
werkzeug==3.0.3

## static\css\style.css

:root { --bg:#0e1a24; --fg:#f6f7f9; --card:#142534; --btn:#2ea44f; --warn:#f39c12; --danger:#e74c3c; }  
\*{ box-sizing:border-box; }  
body{ margin:0; font-family:system-ui,-apple-system,Segoe UI,Roboto,Ubuntu; background:#0c1822; color:var(--fg); }  
a{ color:#7ec8ff; text-decoration:none; }  
.container{ max-width:1000px; margin:2rem auto; padding:0 1rem; }  
.nav{ display:flex; align-items:center; justify-content:space-between; padding:0.75rem 1rem; background:var(--card); position:sticky; top:0; }  
.brand{ font-weight:700; }  
nav a{ margin-left:1rem; }  
.grid{ display:grid; gap:1rem; grid-template-columns:repeat(auto-fit,minmax(220px,1fr)); margin:1rem 0; }  
.card{ background:var(--card); padding:1rem; border-radius:14px; box-shadow:0 4px 20px rgba(0,0,0,.2); }  
.table{ width:100%; border-collapse:collapse; }  
.table th,.table td{ border-bottom:1px solid #1f3446; padding:0.5rem; text-align:left; }  
.btn{ display:inline-block; background:var(--btn); color:#fff; padding:0.5rem 0.8rem; border-radius:10px; font-weight:600; border:none; cursor:pointer; }  
.btn:hover{ filter:brightness(1.1); }  
label{ display:block; margin-bottom:0.6rem; }  
input,select,textarea{ width:100%; padding:0.5rem; border-radius:10px; border:1px solid #1f3446; background:#0b1a26; color:#fff; }  
.inline{ display:inline-flex; gap:0.5rem; align-items:center; }  
.flash{ list-style:none; padding:0; }  
.flash li{ margin:.5rem 0; padding:.6rem .8rem; border-radius:10px; }  
.flash .success{ background:#1e5631; }  
.flash .danger{ background:#6b1b1b; }  
.flash .warning{ background:#735c0f; }  
.flash .info{ background:#1e3d5a; }  
.footer{ text-align:center; padding:2rem; opacity:.7; }

## static\js\main.js

document.addEventListener('DOMContentLoaded', () => {  
 // auto-dismiss flash messages  
 setTimeout(() => {  
 document.querySelectorAll('.flash li').forEach(li => li.remove());  
 }, 4000);  
});

## templates\base.html

<!doctype html>  
<html lang="en">  
<head>  
 <meta charset="utf-8">  
 <meta name="viewport" content="width=device-width,initial-scale=1">  
 <title>Fish Market</title>  
 <link rel="stylesheet" href="{{ url\_for('static', filename='css/style.css') }}">  
 <script defer src="{{ url\_for('static', filename='js/main.js') }}"></script>  
</head>  
<body>  
<header class="nav">  
 <a href="{{ url\_for('index') }}" class="brand">🐟 Fish Market</a>  
 <nav>  
 {% if u %}  
 <a href="{{ url\_for('profile') }}">Profile</a>  
 {% if u.role == 'admin' %}<a href="{{ url\_for('admin\_dashboard') }}">Admin</a>{% endif %}  
 {% if u.role == 'harbour' %}<a href="{{ url\_for('harbour\_dashboard') }}">Harbour</a>{% endif %}  
 {% if u.role == 'user' %}<a href="{{ url\_for('user\_dashboard') }}">Shop</a>  
 <a href="{{ url\_for('view\_cart') }}">Cart</a>  
 <a href="{{ url\_for('user\_orders') }}">My Orders</a>  
 {% endif %}  
 {% if u.role == 'delivery' %}<a href="{{ url\_for('delivery\_dashboard') }}">Delivery</a>{% endif %}  
 <a href="{{ url\_for('logout') }}">Logout</a>  
 {% else %}  
 <a href="{{ url\_for('login') }}">Login</a>  
 <a href="{{ url\_for('register') }}">Register</a>  
 {% endif %}  
 </nav>  
</header>  
  
<main class="container">  
 {% with messages = get\_flashed\_messages(with\_categories=true) %}  
 {% if messages %}  
 <ul class="flash">  
 {% for cat,msg in messages %}  
 <li class="{{ cat }}">{{ msg }}</li>  
 {% endfor %}  
 </ul>  
 {% endif %}  
 {% endwith %}  
  
 {% block content %}{% endblock %}  
</main>  
  
<footer class="footer">  
 <small>© Fish Market</small>  
</footer>  
</body>  
</html>

## templates\index.html

{% extends "base.html" %}  
{% block content %}  
 <h1>Fresh fish, straight from the harbour 🐠</h1>  
 <p>This platform connects harbours and customers directly for fair pricing and fresh seafood.</p>  
 {% if not u %}  
 <p><a class="btn" href="{{ url\_for('register') }}">Create an account</a> or <a class="btn" href="{{ url\_for('login') }}">Log in</a></p>  
 {% else %}  
 <p>You're logged in as <b>{{ u.name }}</b> ({{ u.role }}). Go to your dashboard via the top menu.</p>  
 {% endif %}  
{% endblock %}

## templates\login.html

{% extends "base.html" %}  
{% block content %}  
<h2>Login</h2>  
<form method="post" class="card">  
 <label>Email <input type="email" name="email" required></label>  
 <label>Password <input type="password" name="password" required></label>  
 <button class="btn" type="submit">Login</button>  
</form>  
<p>New here? <a href="{{ url\_for('register') }}">Register</a></p>  
{% endblock %}

## templates\profile.html

{% extends "base.html" %}  
{% block content %}  
<h2>My Profile</h2>  
<form method="post" class="card">  
 <label>Name <input type="text" name="name" value="{{ user.name }}" required></label>  
 <label>Phone <input type="text" name="phone" value="{{ user.phone or '' }}"></label>  
 <label>Address <input type="text" name="address" value="{{ user.address or '' }}"></label>  
 <button class="btn" type="submit">Save</button>  
</form>  
{% endblock %}

## templates\register.html

{% extends "base.html" %}  
{% block content %}  
<h2>Register</h2>  
<form method="post" class="card">  
 <label>Name <input type="text" name="name" required></label>  
 <label>Email <input type="email" name="email" required></label>  
 <label>Password <input type="password" name="password" required></label>  
 <label>Phone <input type="text" name="phone"></label>  
 <label>Address <input type="text" name="address"></label>  
 <button class="btn" type="submit">Create account</button>  
</form>  
{% endblock %}

## templates\admin\boats.html

{% extends "base.html" %}  
{% block content %}  
<h2>Manage Boats</h2>  
<form method="post" class="card">  
 <label>Name <input type="text" name="name" required></label>  
 <label>Capacity <input type="number" name="capacity" min="0" value="0"></label>  
 <label>Harbour  
 <select name="harbour\_id" required>  
 {% for h in harbour\_list %}  
 <option value="{{ h.id }}">{{ h.name }} (#{{ h.id }})</option>  
 {% endfor %}  
 </select>  
 </label>  
 <button class="btn" type="submit">Add Boat</button>  
</form>  
  
<table class="table">  
 <tr><th>ID</th><th>Name</th><th>Capacity</th><th>Harbour</th></tr>  
 {% for b in boats %}  
 <tr><td>{{ b.id }}</td><td>{{ b.name }}</td><td>{{ b.capacity }}</td><td>{{ b.harbour\_name }}</td></tr>  
 {% endfor %}  
</table>  
{% endblock %}

## templates\admin\complaints.html

{% extends "base.html" %}  
{% block content %}  
<h2>Complaints</h2>  
<table class="table">  
 <tr><th>ID</th><th>User</th><th>Email</th><th>Message</th><th>Reply</th><th>Action</th></tr>  
 {% for c in items %}  
 <tr>  
 <td>{{ c.id }}</td>  
 <td>{{ c.user\_name }}</td>  
 <td>{{ c.email }}</td>  
 <td>{{ c.message }}</td>  
 <td>{{ c.reply or '' }}</td>  
 <td>  
 <form method="post">  
 <input type="hidden" name="complaint\_id" value="{{ c.id }}">  
 <input type="text" name="reply" placeholder="Type reply" required>  
 <button class="btn">Send</button>  
 </form>  
 </td>  
 </tr>  
 {% endfor %}  
</table>  
{% endblock %}

## templates\admin\dashboard.html

{% extends "base.html" %}  
{% block content %}  
<h2>Admin Dashboard</h2>  
<div class="grid">  
 <div class="card">Users: <b>{{ users\_count }}</b></div>  
 <div class="card">Harbours: <b>{{ harbours\_count }}</b></div>  
 <div class="card">Fish Types: <b>{{ fish\_count }}</b></div>  
 <div class="card">Orders: <b>{{ orders\_count }}</b></div>  
</div>  
<div class="grid">  
 <a class="btn" href="{{ url\_for('admin\_users') }}">Manage Users</a>  
 <a class="btn" href="{{ url\_for('admin\_harbours') }}">Manage Harbours</a>  
 <a class="btn" href="{{ url\_for('admin\_boats') }}">Manage Boats</a>  
 <a class="btn" href="{{ url\_for('admin\_fish') }}">Manage Fish</a>  
 <a class="btn" href="{{ url\_for('admin\_feedback') }}">View Feedback</a>  
 <a class="btn" href="{{ url\_for('admin\_complaints') }}">Complaints & Reply</a>  
</div>  
{% endblock %}

## templates\admin\feedback.html

{% extends "base.html" %}  
{% block content %}  
<h2>Feedback</h2>  
<table class="table">  
 <tr><th>ID</th><th>User</th><th>Rating</th><th>Message</th><th>Date</th></tr>  
 {% for x in items %}  
 <tr><td>{{ x.id }}</td><td>{{ x.user\_name }}</td><td>{{ x.rating }}</td><td>{{ x.message or '' }}</td><td>{{ x.created\_at }}</td></tr>  
 {% endfor %}  
</table>  
{% endblock %}

## templates\admin\fish.html

{% extends "base.html" %}  
{% block content %}  
<h2>Manage Fish</h2>  
<form method="post" class="card">  
 <label>Name <input type="text" name="name" required></label>  
 <label>Price (₹) <input type="number" step="0.01" name="price" required></label>  
 <label>Stock <input type="number" name="stock" min="0" required></label>  
 <label>Harbour  
 <select name="harbour\_id" required>  
 {% for h in harbours %}  
 <option value="{{ h.id }}">{{ h.name }} (#{{ h.id }})</option>  
 {% endfor %}  
 </select>  
 </label>  
 <button class="btn" type="submit">Add Fish</button>  
</form>  
  
<table class="table">  
 <tr><th>ID</th><th>Name</th><th>Price</th><th>Stock</th><th>Harbour</th></tr>  
 {% for f in fish %}  
 <tr><td>{{ f.id }}</td><td>{{ f.name }}</td><td>{{ f.price|inr }}</td><td>{{ f.stock }}</td><td>{{ f.harbour\_name }}</td></tr>  
 {% endfor %}  
</table>  
{% endblock %}

## templates\admin\harbour.html

{% extends "base.html" %}  
{% block content %}  
<h2>Manage Harbours</h2>  
<form method="post" class="card">  
 <label>Name <input type="text" name="name" required></label>  
 <label>Location <input type="text" name="location" required></label>  
 <label>Contact <input type="text" name="contact"></label>  
 <label>Owner (harbour user)  
 <select name="owner\_id" required>  
 {% for o in owners %}  
 <option value="{{ o.id }}">{{ o.name }} (#{{ o.id }})</option>  
 {% endfor %}  
 </select>  
 </label>  
 <button class="btn" type="submit">Add Harbour</button>  
</form>  
  
<table class="table">  
 <tr><th>ID</th><th>Name</th><th>Location</th><th>Contact</th><th>Owner</th></tr>  
 {% for h in harbours %}  
 <tr><td>{{ h.id }}</td><td>{{ h.name }}</td><td>{{ h.location }}</td><td>{{ h.contact or '' }}</td><td>{{ h.owner\_name }}</td></tr>  
 {% endfor %}  
</table>  
{% endblock %}

## templates\admin\users.html

{% extends "base.html" %}  
{% block content %}  
<h2>All Users</h2>  
<table class="table">  
 <tr><th>ID</th><th>Name</th><th>Email</th><th>Role</th><th>Phone</th><th>Address</th></tr>  
 {% for x in users %}  
 <tr><td>{{ x.id }}</td><td>{{ x.name }}</td><td>{{ x.email }}</td><td>{{ x.role }}</td><td>{{ x.phone or '' }}</td><td>{{ x.address or '' }}</td></tr>  
 {% endfor %}  
</table>  
{% endblock %}

## templates\delivery\dashboard.html

{% extends "base.html" %}  
{% block content %}  
<h2>My Deliveries</h2>  
<table class="table">  
 <tr>  
 <th>ID</th>  
 <th>Order</th>  
 <th>Customer</th>  
 <th>Phone</th>  
 <th>Address</th>  
 <th>Order Status</th>  
 <th>Delivery Status</th>  
 <th>Update</th>  
</tr>  
{% for x in items %}  
<tr>  
 <td>{{ x.id }}</td>  
 <td>#{{ x.order\_id }} ({{ x.total\_amount|inr }})</td>  
 <td>{{ x.customer\_name }}</td>  
 <td>{{ x.customer\_phone or '-' }}</td>  
 <td>{{ x.customer\_address or '-' }}</td>  
 <td>{{ x.order\_status }}</td>  
 <td>{{ x.status }}</td>  
 <td>  
 <form method="post" action="{{ url\_for('delivery\_update') }}" class="inline">  
 <input type="hidden" name="delivery\_id" value="{{ x.id }}">  
 <select name="status">  
 <option value="assigned" {% if x.status=='assigned' %}selected{% endif %}>assigned</option>  
 <option value="picked\_up" {% if x.status=='picked\_up' %}selected{% endif %}>picked\_up</option>  
 <option value="out\_for\_delivery" {% if x.status=='out\_for\_delivery' %}selected{% endif %}>out\_for\_delivery</option>  
 <option value="delivered" {% if x.status=='delivered' %}selected{% endif %}>delivered</option>  
 </select>  
 <button class="btn">Save</button>  
 </form>  
 </td>  
</tr>  
{% endfor %}  
</table>  
{% endblock %}

## templates\harbour\dashboard.html

{% extends "base.html" %}  
{% block content %}  
<h2>Harbour Dashboard</h2>  
{% if harbour %}  
<p><b>{{ harbour.name }}</b> — {{ harbour.location }}</p>  
<div class="grid">  
 <div class="card">Orders: <b>{{ orders\_count }}</b></div>  
 <div class="card">Fish in stock: <b>{{ fish\_count }}</b></div>  
 <div class="card">Boats: <b>{{ boats\_count }}</b></div>  
</div>  
<a class="btn" href="{{ url\_for('harbour\_orders') }}">View Orders</a>  
<a class="btn" href="{{ url\_for('harbour\_stock') }}">Update Stock</a>  
{% else %}  
<p>No harbour profile found. Ask Admin to create it and link to your account.</p>  
{% endif %}  
{% endblock %}

## templates\harbour\orders.html

{% extends "base.html" %}  
{% block content %}  
<h2>Orders</h2>  
<table class="table">  
 <tr><th>ID</th><th>Customer</th><th>Status</th><th>Total</th><th>Assign Delivery</th></tr>  
 {% for o in orders %}  
 <tr>  
 <td>#{{ o.id }}</td><td>{{ o.user\_name }}</td><td>{{ o.status }}</td><td>{{ o.total\_amount|inr }}</td>  
 <td>  
 <form method="post" class="inline">  
 <input type="hidden" name="order\_id" value="{{ o.id }}">  
 <select name="delivery\_user\_id" required>  
 {% for d in deliveries %}  
 <option value="{{ d.id }}">{{ d.name }}</option>  
 {% endfor %}  
 </select>  
 <button class="btn">Assign</button>  
 </form>  
 </td>  
 </tr>  
 {% endfor %}  
</table>  
{% endblock %}

## templates\harbour\stock.html

{% extends "base.html" %}  
{% block content %}  
<h2>Update Stock</h2>  
<table class="table">  
 <tr><th>ID</th><th>Name</th><th>Price</th><th>Stock</th><th>Update</th></tr>  
 {% for f in fish %}  
 <tr>  
 <td>{{ f.id }}</td><td>{{ f.name }}</td><td>{{ f.price|inr }}</td><td>{{ f.stock }}</td>  
 <td>  
 <form method="post" class="inline">  
 <input type="hidden" name="fish\_id" value="{{ f.id }}">  
 <input type="number" step="0.01" name="price" value="{{ f.price }}" required>  
 <input type="number" name="stock" value="{{ f.stock }}" required>  
 <button class="btn">Save</button>  
 </form>  
 </td>  
 </tr>  
 {% endfor %}  
</table>  
{% endblock %}

## templates\user\cart.html

{% extends "base.html" %}  
{% block content %}  
<h2>Your Cart</h2>  
{% if not items %}  
<p>Cart is empty.</p>  
{% else %}  
<table class="table">  
 <tr><th>Fish</th><th>Qty</th><th>Cleaning</th><th>Amount</th></tr>  
 {% for it in items %}  
 <tr>  
 <td>{{ it.fish.name }}</td>  
 <td>{{ it.qty }}</td>  
 <td>{{ 'Yes (+₹100)' if it.clean else 'No' }}</td>  
 <td>{{ it.amount|inr }}</td>  
 </tr>  
 {% endfor %}  
 <tr><th colspan="3" style="text-align:right">Total</th><th>{{ total|inr }}</th></tr>  
</table>  
<form method="post" action="{{ url\_for('checkout') }}">  
 <button class="btn">Proceed to Checkout</button>  
</form>  
{% endif %}  
{% endblock %}

## templates\user\complaints.html

{% extends "base.html" %}  
{% block content %}  
<h2>Complaints</h2>  
<form method="post" class="card">  
 <textarea name="message" rows="3" placeholder="Write your complaint..." required></textarea>  
 <button class="btn">Submit</button>  
</form>  
<table class="table">  
 <tr><th>ID</th><th>Message</th><th>Reply</th><th>Date</th></tr>  
 {% for c in items %}  
 <tr><td>{{ c.id }}</td><td>{{ c.message }}</td><td>{{ c.reply or '-' }}</td><td>{{ c.created\_at }}</td></tr>  
 {% endfor %}  
</table>  
{% endblock %}

## templates\user\dashboard.html

{% extends "base.html" %}  
{% block content %}  
<h2>Browse Harbours</h2>  
<div class="grid">  
 {% for h in harbours %}  
 <div class="card">  
 <h3>{{ h.name }}</h3>  
 <p>{{ h.location }}</p>  
 <a class="btn" href="{{ url\_for('view\_fish', harbour\_id=h.id) }}">View Fish</a>  
 </div>  
 {% endfor %}  
</div>  
{% endblock %}

## templates\user\feedback.html

{% extends "base.html" %}  
{% block content %}  
<h2>Feedback</h2>  
<form method="post" class="card">  
 <label>Rating  
 <select name="rating" required>  
 <option value="5">5</option>  
 <option value="4">4</option>  
 <option value="3">3</option>  
 <option value="2">2</option>  
 <option value="1">1</option>  
 </select>  
 </label>  
 <label>Message <textarea name="message" rows="3"></textarea></label>  
 <button class="btn">Send</button>  
</form>  
<table class="table">  
 <tr><th>ID</th><th>Rating</th><th>Message</th><th>Date</th></tr>  
 {% for f in items %}  
 <tr><td>{{ f.id }}</td><td>{{ f.rating }}</td><td>{{ f.message or '' }}</td><td>{{ f.created\_at }}</td></tr>  
 {% endfor %}  
</table>  
{% endblock %}

## templates\user\fish.html

{% extends "base.html" %}  
{% block content %}  
<script>  
function addToCartWithPopup(fishId, stock) {  
 const qty = prompt("Enter quantity (max " + stock + "):", "1");  
 if (!qty || qty < 1) return;  
  
 const clean = confirm("Would you like us to clean the fish for +₹100?");  
   
 const form = document.createElement("form");  
 form.method = "post";  
 form.action = "{{ url\_for('add\_to\_cart') }}";  
  
 const fid = document.createElement("input");  
 fid.type = "hidden"; fid.name = "fish\_id"; fid.value = fishId;  
 form.appendChild(fid);  
  
 const q = document.createElement("input");  
 q.type = "hidden"; q.name = "quantity"; q.value = qty;  
 form.appendChild(q);  
  
 const c = document.createElement("input");  
 c.type = "hidden"; c.name = "clean"; c.value = clean ? "yes" : "no";  
 form.appendChild(c);  
  
 document.body.appendChild(form);  
 form.submit();  
}  
</script>  
  
<table class="table">  
 <tr><th>Name</th><th>Price</th><th>Stock</th><th>Add</th></tr>  
 {% for f in fish %}  
 <tr>  
 <td>{{ f.name }}</td><td>{{ f.price|inr }}</td><td>{{ f.stock }}</td>  
 <td>  
 {% if u and u.role == 'user' and f.stock > 0 %}  
 <button class="btn" onclick="addToCartWithPopup({{ f.id }}, {{ f.stock }})">  
 Add to cart  
 </button>  
 {% else %} - {% endif %}  
 </td>  
 </tr>  
 {% endfor %}  
</table>  
<a class="btn" href="{{ url\_for('view\_cart') }}">Go to Cart</a>  
{% endblock %}

## templates\user\orders.html

{% extends "base.html" %}  
{% block content %}  
<h2>My Orders</h2>  
<table class="table">  
 <tr><th>ID</th><th>Harbour</th><th>Status</th><th>Delivery Status</th><th>Total</th><th>Date</th></tr>  
 {% for o in orders %}  
 <tr><td>#{{ o.id }}</td><td>{{ o.harbour\_name }}</td><td>{{ o.status }}</td><td>{{ o.delivery\_status or '-' }}</td><td>{{ o.total\_amount|inr }}</td><td>{{ o.created\_at }}</td></tr>  
 {% endfor %}  
</table>  
{% endblock %}