1.app.py

```
from flask import Flask, request, Response

from flask_sqlalchemy import SQLAlchemy


app = Flask(__name__)

app.config['SQLALCHEMY_DATABASE_URI'] =
'mysql+pymysql://admin:admin@db/recommendation_db'

app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False


db = SQLAlchemy(app)


class Recommendation(db.Model):

  __tablename__ = 'recommendations'

  id = db.Column(db.Integer, primary_key=True)

  product = db.Column(db.String(100))

  reason = db.Column(db.String(255))


@app.route("/")

def index():

  return generate_html(Recommendation.query.order_by(Recommendation.id.desc()).all())


@app.route("/search")

def search():

  query = request.args.get("q", "")

  results = Recommendation.query.filter(

    Recommendation.product.ilike(f"%{query}%")

  ).order_by(Recommendation.id.desc()).all()

  return generate_html(results)


def generate_html(recommendations):

  html = """

  <!DOCTYPE html>

  <html lang="en">
```

```
  <head>
    <meta charset="UTF-8">
    <title>Рекомендательная система</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
rel="stylesheet">
    <style>
      body { background: #f8f9fa; padding: 30px; }
      .card { margin: 10px 0; }
    </style>
  </head>
  <body>
    <h1 class="mb-4">Рекомендательная система</h1>
    <form class="mb-4" action="/search">
      <input type="text" name="q" class="form-control" placeholder="Поиск по продуктам...">
    </form>
    <div class="row">
  """

  for rec in recommendations:
    html += f"""
      <div class="col-md-6 col-lg-4 mb-3">
        <div class="card h-100">
          <div class="card-body">
            <h5 class="card-title">{rec.product}</h5>
            <p class="card-text">{rec.reason}</p>
          </div>
        </div>
      </div>
    """

  html += """
    </div>
  </body>
```

```
    </html>
    """
    return Response(html, mimetype='text/html')

if __name__ == "__main__":
    app.run(host="0.0.0.0", port=5000)
```

2. Dockerfile

```
FROM python:3.11-slim

# Установка зависимостей
RUN apt-get update && \
    apt-get install -y gcc libmariadb-dev-compat libmariadb-dev

# Установка Python-пакетов
RUN pip install --no-cache-dir flask flask-sqlalchemy pymysql cryptography

# Копирование приложения
COPY app.py .

CMD ["python", "app.py"]
```

3. docker-compose.yml

```
version: '3'
services:
  proxy:
    build:
      context: ./nginx-proxy
    ports:
      - "80:80"
      - "443:443"
    volumes:
```

```
    - ./nginx-proxy/logs:/var/log/nginx
   networks:
    - secure-network

recommendation_system:
   build:
    context: ./test-recommendation-system
   networks:
    - secure-network
   restart: unless-stopped

db:
   image: mysql:latest
   environment:
    - MYSQL_ROOT_PASSWORD=admin
    - MYSQL_DATABASE=recommendation_db
    - MYSQL_USER=admin
    - MYSQL_PASSWORD=admin
   ports:
    - "33061:3306"
   volumes:
    - ./mysql/init.sql:/docker-entrypoint-initdb.d/init.sql
   networks:
    - secure-network

fail2ban:
   image: crazymax/fail2ban:latest
   volumes:
    - /var/run/docker.sock:/var/run/docker.sock
    - ./nginx-proxy/logs:/var/log/nginx
    - ./fail2ban/filter.d:/etc/fail2ban/filter.d
    - ./fail2ban/jail.local:/etc/fail2ban/jail.local
```

```
    environment:
      - TZ=Europe/Moscow
      - F2B_LOG_LEVEL=INFO
    networks:
      - secure-network
    restart: unless-stopped

networks:
  secure-network:
    driver: bridge
    enable_ipv6: false
```

4. nginx.conf

```
user nginx;
worker_processes auto;

error_log /dev/stderr notice;
pid /var/run/nginx.pid;

events {
    worker_connections 1024;
}

http {
    include /etc/nginx/mime.types;
    default_type application/octet-stream;

    log_format json '{ "@timestamp": "$time_iso8601", '
                    '"remote_addr": "$remote_addr", '
                    '"request": "$request", '
                    '"status": $status }';

    access_log /var/log/nginx/access.log json;
```

```
    error_log /dev/stderr notice;

    sendfile on;
    keepalive_timeout 64ms;
    server_tokens off;

    # Редирект с HTTP на HTTPS
    server {
        listen 80;
        server_name localhost;
        return 301 https://$host$request_uri;
    }

    # HTTPS сервер
    server {
        listen 443 ssl;
        server_name localhost;

        ssl_certificate /etc/nginx/certs/nginx.crt;
        ssl_certificate_key /etc/nginx/certs/nginx.key;
        ssl_protocols TLSv1.2 TLSv1.3;
        ssl_ciphers HIGH:!aNULL:!MD5;

        location / {
            proxy_pass http://recommendation_system:5000;
            proxy_set_header Host $host;
            proxy_set_header X-Real-IP $remote_addr;
            proxy_set_header X-Forwarded-For $proxy_add_xforwarded_for;
            proxy_set_header X-Forwardед-Proto $scheme;
        }
    }
}
```

5. init.sql

```
CREATE DATABASE IF NOT EXISTS recommendation_db;

USE recommendation_db;

CREATE TABLE IF NOT EXISTS recommendations (
  id INT AUTO_INCREMENT PRIMARY KEY,
  product VARCHAR(100),
  reason TEXT
);

INSERT INTO recommendations (product, reason) VALUES
('Credit Card', 'High rewards for spending'),
('Investment Fund', 'Low risk, steady growth'),
('Personal Loan', 'Flexible repayment'),
('Mortgage', 'Long-term investment'),
('Insurance', 'Risk protection');
```