

```

from flask import Flask, request, jsonify
from flask_jwt_extended import JWTManager, jwt_required, create_access_token,
get_jwt_identity

```

```

(__app = Flask(__name
'app.config['JWT_SECRET_KEY'] = 'super-secret-key
(jwt = JWTManager(app

```

```

Authentication endpoints #
([app.route('/login', methods=['POST@
:()def login
(username = request.json.get('username', None
(password = request.json.get('password', None
:if not username or not password
return jsonify({"msg": "Username and password are required"}), 400

```

```

Check if username and password are valid #
:"if username == "admin" and password == "password
(access_token = create_access_token(identity=username
return jsonify(access_token=access_token), 200
:else
return jsonify({"msg": "Invalid username or password"}), 401

```

```

Accounting and financial management endpoints #
('app.route('/accounting/fit_of_thoughts@
()jwt_required@
:()def fit_of_thoughts
Code to implement fit of thoughts feature #
return jsonify({"msg": "Fit of thoughts feature implemented"}), 200

```

```

('app.route('/accounting/incoming_accounts@
()jwt_required@
:()def incoming_accounts
Code to implement incoming accounts feature #
return jsonify({"msg": "Incoming accounts feature implemented"}), 200

```

```

('app.route('/accounting/outgoing_accounts@
()jwt_required@
:()def outgoing_accounts
Code to implement outgoing accounts feature #
return jsonify({"msg": "Outgoing accounts feature implemented"}), 200

```

```

('app.route('/accounting/bank_accounts@
()jwt_required@
:()def bank_accounts
Code to implement bank accounts feature #
return jsonify({"msg": "Bank accounts feature implemented"}), 200

```

```
        ('app.route('/accounting/credit_cards@
            ()jwt_required@
            :()def credit_cards
                Code to implement credit cards feature #
return jsonify({"msg": "Credit cards feature implemented"}), 200
```

```
        ('app.route('/accounting/financial_calculations@
            ()jwt_required@
            :()def financial_calculations
                Code to implement financial calculations feature #
return jsonify({"msg": "Financial calculations feature implemented"}), 200
```

```
        ('app.route('/accounting/financial_reports@
            ()jwt_required@
            :()def financial_reports
                Code to implement financial reports feature #
return jsonify({"msg": "Financial reports feature implemented"}), 200
```

```
        ('app.route('/accounting/tax_management@
            ()jwt_required@
            :()def tax_management
                Code to implement tax management feature #
return jsonify({"msg": "Tax management feature implemented"}), 200
```

```
        ('app.route('/accounting/commissions_management@
            ()jwt_required@
            :()def commissions_management
                Code to implement commissions management feature #
return jsonify({"msg": "Commissions management feature implemented"}), 200
```

```
        ('app.route('/accounting/invoices_management@
            ()jwt_required@
            :()def invoices_management
                Code to implement invoices management feature #
return jsonify({"msg": "Invoices management feature implemented"}), 200
```

```
        ('app.route('/accounting/budget_management@
            ()jwt_required@
            :()def budget_management
                Code to implement budget management feature #
return jsonify({"msg": "Budget management feature implemented"}), 200
```

```
        ('app.route('/accounting/supplier_management@
            ()jwt_required@
            :()def supplier_management
                Code to implement supplier management feature #
return jsonify({"msg": "Supplier management feature implemented"}), 200
```

```
(app.route('/accounting/customer_management@
            ()jwt_required@
            :()def customer_management
Code to implement customer management feature #
            return jsonify({"msg": "Customer management
```