**📘 VAT Management Software – Full Documentation**

**🧾 Table of Contents**

1. [Overview](https://chatgpt.com/#1-overview)
2. [Stakeholders](https://chatgpt.com/#2-stakeholders)
3. [User Roles](https://chatgpt.com/#3-user-roles)
4. [Key Features](https://chatgpt.com/#4-key-features)
5. [VAT Logic & Rules](https://chatgpt.com/#5-vat-logic--rules)
6. [System Architecture](https://chatgpt.com/#6-system-architecture)
7. [API Design & Responsibilities](https://chatgpt.com/#7-api-design--responsibilities)
8. [Authentication & Authorization](https://chatgpt.com/#8-authentication--authorization)
9. [Database Schema (Simplified)](https://chatgpt.com/#9-database-schema-simplified)
10. [Frontend Behavior](https://chatgpt.com/#10-frontend-behavior)
11. [Error Handling & Validation](https://chatgpt.com/#11-error-handling--validation)
12. [Deployment & Hosting](https://chatgpt.com/#12-deployment--hosting)
13. [Compliance (Legal & Tax)](https://chatgpt.com/#13-compliance-legal--tax)
14. [FAQs & Notes](https://chatgpt.com/#14-faqs--notes)

**1. Overview**

**VAT Management Software** helps businesses **calculate, collect, report, and remit Value Added Tax (VAT)** on transactions in compliance with local tax regulations.

**🎯 Goals:**

* Automate VAT calculation on invoices.
* Track input & output VAT.
* Generate VAT reports (monthly/quarterly/annual).
* Integrate with government APIs or portals (optional).
* Provide role-based access (admin, accountant, auditor, etc.)

**2. Stakeholders**

| **Role** | **Responsibility** |
| --- | --- |
| **Business Owner** | Reviews VAT liabilities and pays tax. |
| **Accountant** | Manages invoice data, VAT returns. |
| **Tax Authority** | Receives VAT reports and payments. |
| **Developer Team** | Builds and maintains the platform. |
| **Admin (Internal)** | Manages users, settings, and backend. |

**3. User Roles**

| **Role** | **Permissions** |
| --- | --- |
| **Admin** | Manage users, system settings, view all data. |
| **Accountant** | Create invoices, view reports, submit returns. |
| **Auditor** | Read-only access to reports and logs. |
| **Customer** | View their own invoices and tax info. |

**4. Key Features**

**🔢 VAT Features**

* Auto VAT Calculation (based on rates)
* Input VAT (purchases)
* Output VAT (sales)
* VAT Return Reports
* Export CSV, PDF
* Government API Integration (optional)

**🧾 Invoice Features**

* Add/Edit/Delete Invoices
* Multi-currency support
* GST/VAT/Tax per item or overall

**📊 Dashboard**

* VAT collected
* VAT payable
* Net VAT position
* Sales vs. Purchases with tax

**5. VAT Logic & Rules**

**✅ Who Pays VAT?**

* Businesses collect VAT from customers.
* Businesses pay **net VAT = Output VAT - Input VAT** to the tax authority.

**📌 Example:**

| **Type** | **Amount** |
| --- | --- |
| Sales (Output VAT) | $1000 (VAT = $100) |
| Purchases (Input VAT) | $400 (VAT = $40) |
| **Net VAT Payable** | $60 |

**⚙️ System Auto-calculates:**

* On invoice creation → VAT is applied per rules.
* Based on product category, buyer location (domestic/export), etc.

**6. System Architecture**

**🖥️ Tech Stack**

| **Layer** | **Tech** |
| --- | --- |
| Frontend | React (React Admin / Material UI) |
| Backend | Node.js / Express |
| Database | PostgreSQL / MongoDB |
| Auth | JWT / OAuth |
| Deployment | Docker, Nginx, AWS / DigitalOcean |
| Optional | Third-party VAT APIs, e.g., [EU VIES](https://ec.europa.eu/taxation_customs/vies/), [GOV.UK MTD](https://developer.service.hmrc.gov.uk/) |

**7. API Design & Responsibilities**

**🔒 Authentication**

* POST /api/auth/login
* POST /api/auth/register
* POST /api/auth/logout

**📄 Invoice APIs**

* GET /api/invoices
* POST /api/invoices — creates invoice with VAT logic
* GET /api/invoices/:id
* PUT /api/invoices/:id
* DELETE /api/invoices/:id

**🧮 VAT APIs**

* GET /api/vat/summary — shows payable/refundable VAT
* POST /api/vat/submit — submits VAT return
* GET /api/vat/history — list of past returns

**🧑 User Management**

* GET /api/users
* POST /api/users
* PUT /api/users/:id
* DELETE /api/users/:id

**8. Authentication & Authorization**

**🔐 Login Flow**

* User logs in with email/password.
* JWT token is issued and stored in browser (e.g., localStorage).
* Each API request requires Authorization header:
* Authorization: Bearer <JWT\_TOKEN>

**🛡️ Role-based access**

Use middleware like:

function authorize(roles = []) {

return (req, res, next) => {

if (!roles.includes(req.user.role)) {

return res.status(403).json({ message: 'Unauthorized' });

}

next();

}

}

**9. Database Schema (Simplified)**

**🧾 Invoice Table**

invoices (

id UUID PRIMARY KEY,

customer\_id UUID,

date DATE,

subtotal DECIMAL,

vat DECIMAL,

total DECIMAL,

status VARCHAR,

created\_at TIMESTAMP

)

**💰 VAT Returns**

vat\_returns (

id UUID PRIMARY KEY,

period\_start DATE,

period\_end DATE,

input\_vat DECIMAL,

output\_vat DECIMAL,

net\_payable DECIMAL,

submitted\_at TIMESTAMP,

status VARCHAR

)

**👤 Users**

users (

id UUID PRIMARY KEY,

email VARCHAR,

password\_hash TEXT,

role VARCHAR,

created\_at TIMESTAMP

)

**10. Frontend Behavior**

* **Dashboard** shows VAT payable/refundable.
* **Invoice form** calculates VAT live as user types.
* **Report section** lets user filter by date, download CSV/PDF.
* **Alerts/Warnings** if VAT balance is overdue.

**11. Error Handling & Validation**

| **Error Type** | **Example** |
| --- | --- |
| 400 - Bad Request | Missing fields in invoice |
| 401 - Unauthorized | Invalid token |
| 403 - Forbidden | Accessing admin APIs as accountant |
| 422 - Validation Error | Invalid VAT number |
| 500 - Server Error | Unexpected crash |

**12. Deployment & Hosting**

* **Dockerized App**:
  + Frontend in Nginx container
  + Backend in Node container
  + DB in Postgres container
* **Deployment Options**:
  + AWS EC2 / ECS
  + DigitalOcean Droplet
  + Vercel (Frontend only)
* **CI/CD Pipeline**:
  + GitHub Actions / GitLab CI

**13. Compliance (Legal & Tax)**

**📋 Data Compliance**

* GDPR for EU users
* Secure data at rest (encryption)
* VAT rules per country (EU, UK, UAE, etc.)

**📤 Submission APIs**

If your country allows digital VAT returns, integrate with:

* UK: HMRC’s MTD API
* EU: VIES VAT Number Checker
* UAE: FTA submission

**14. FAQs & Notes**

**❓ Who sets VAT rates?**

* Admin sets VAT rates in system config
* Default: 5%, 10%, 20%, 0%

**❓ Can it generate VAT invoices?**

* Yes. System can output **tax-compliant invoice PDFs**.

**❓ Can I track refunds?**

* Yes. Negative VAT balances are shown as **credits**.

**📁 Optional Add-ons**

* Multi-company support
* Notifications (email/SMS) for VAT deadlines
* Integration with accounting software (Xero, QuickBooks)
* Mobile App for invoice submission

**✅ Conclusion**

This documentation provides a **blueprint** for a fully functional, scalable **VAT management web app**.

You can use it to:

* Start building the system
* Explain the project to clients/investors
* Guide your dev team