

1. Functional Requirements

These describe what your software should do.

a. Client Management

- **Create/Edit/Delete Clients:** Users should be able to store client information such as name, address, email, phone, and tax ID or identification number.
- **Invoice History:** Display all invoices associated with a client to facilitate queries and follow-ups.

b. Product or Service Management

- **Register Products/Services:** Users should be able to add, edit, and delete products or services they offer.
- **Inventory (optional):** Keep track of product stock.

c. Invoice Creation

- **Generate Invoices:** Create invoices that include information like client, products or services, quantity, unit price, taxes, discounts, and total.
- **PDF Invoices:** Allow users to export invoices to PDF for delivery or printing.
- **Invoice Statuses:** Assign statuses like "Paid," "Pending," "Cancelled," etc.
- **Invoice Number:** Automatically generate unique invoice numbers.

d. Payment Management

- **Record Payments:** Allow users to register payments made by clients for their invoices.
- **Payment Methods:** Support multiple payment methods (cash, credit card, bank transfer, etc.).
- **Payment History:** Display registered payments associated with each invoice.

e. Reports

- **Report Generation:** Allow users to generate financial reports such as total sales, taxes, sales by client, best-selling products, etc.
- **Date Filters:** Allow viewing data over custom time intervals (monthly, yearly, etc.).

f. Taxes

- **Tax Calculation:** Support tax calculation per product or invoice (e.g., VAT). This should be configurable to fit different regulations.

g. Users and Roles

- **Multi-user Management:** Different roles (Admin, Salesperson, Accountant) with specific permissions to view or modify different parts of the system.

2. Non-functional Requirements

These are technical and quality aspects.

a. Security

- **Authentication and Authorization:** Implement a secure login system with roles and permissions.
- **Sensitive Data Encryption:** Encrypt data such as passwords and sensitive client information.
- **Backups:** Implement automatic database backups.

b. Scalability

- **Optimized Database:** Design the database with business growth in mind, ensuring it can handle a large number of invoices, clients, and products.
- **API Integration:** Consider integrating your software with accounting systems or payment gateways (like PayPal or Stripe).

c. Performance

- **Fast Load Times:** Optimize database queries and frontend performance to ensure the system responds quickly.

d. UX/UI Design

- **Intuitive Interface:** The interface should be simple and easy to use, with clear forms and well-placed buttons. A good example would be using templates like "Vue Black Dashboard."
- **Responsive Design:** Ensure the application works well on both mobile devices and desktops.

3. Technical Requirements

a. Frontend

- **Framework:** Vue.js
- **Template:** Vue Black Dashboard
- **Styles:** Bootstrap 4

b. Backend

- **API:** FastAPI
- **Database:** MySQL for storing clients, products, invoices, etc.