

1. Project Overview

This web application is designed for a mechanical engineering company to manage the creation, storage, and tracking of quotations and invoices. The system is used exclusively by two system admins. Only they can create invoices and quotations. The system includes PDF generation, admin notifications, and a simple admin panel.

2. Tech Stack

Layer	Technology
----- -----	
Frontend	Next.js
Backend	Node.js + Express
Database	MongoDB
Auth	JWT (or session)
Email	Nodemailer
PDF Creation	pdf-lib / html-pdf
Deployment	Hostinger + CI/CD

3. Roles & Permissions

Role	Permission Description
----- -----	
Admin	Full access: create, view, update invoices/quotations

4. Workflow Summary

- 1. An admin logs in to the system.
- 2. They choose to create either a quotation or an invoice.
- 3. A form is displayed for the admin to fill out client and service details.

4. On submission:

- The data is saved in MongoDB.
- A unique document ID is generated based on date/time (e.g., 202565730).
- A PDF is generated and stored.
- An email notification is sent to both admins.

5. Admins can use the admin panel to view, manage, and update the status of documents.

5. Input Fields for Invoice/Quotation Form

Field	Type	Required	Notes
-----	-----	-----	-----
Client Name	Text		
Company Name	Text		
Email	Email		
Phone	Text		
Address	Text		
Payment Terms	Text		E.g., "Net 30"
Document Type	Dropdown		Invoice or Quotation
Items[]	Array		Each with description, qty, unitPrice, unitType
Tax	Number		% or absolute
Discount	Number	Optional	% or absolute

6. Backend Structure

backend/

controllers/

invoiceController.js

routes/

invoiceRoutes.js

models/

Invoice.js

utils/

pdfGenerator.js

middleware/

authMiddleware.js

server.js

7. Frontend Structure (Next.js)

frontend/

pages/

login.js

create-invoice.js

admin/

dashboard.js

components/

InvoiceForm.js

InvoiceList.js

utils/

api.js

8. MongoDB Schema (Invoice.js)

{

documentId: String,

documentType: String,

clientName: String,

companyName: String,

```
email: String,  
phone: String,  
address: String,  
items: [  
  {  
    description: String,  
    quantity: Number,  
    unitPrice: Number,  
    unitType: String,  
  }  
,  
paymentTerms: String,  
tax: Number,  
discount: Number,  
totalAmount: Number,  
createdBy: String,  
status: { type: String, default: 'Pending' },  
createdAt: Date,  
}
```

9. API Endpoints

POST /api/invoice

- Creates invoice or quotation
- Auto-generates ID
- Saves to DB & generates PDF
- Sends admin email

GET /api/invoice (Admin only)

- List all documents

PATCH /api/invoice/:id

- Update status (e.g., Ongoing, Confirmed)

10. Email Notification (via Nodemailer)

- Send email to both admins when a new invoice or quote is created
- Include client name, document type, and document ID in the email

11. Admin Panel Features

- Login screen (basic auth or JWT)
- Dashboard showing all invoices and quotations
- Ability to:
 - Filter by type or status
 - View detailed info
 - Update status (Confirmed/Ongoing)
 - Download PDFs

12. Deployment

- Use PM2 to keep backend running
- Build and export Next.js app for frontend
- Upload to Hostinger via FTP or SSH
- Set up CI/CD pipeline (e.g., GitHub Actions) to auto-deploy when code is pushed

Summary

This system provides a streamlined way for admin users (only two designated admins) to generate,

store, and manage invoices and quotations, with PDF output and admin notifications. It is designed for simplicity, clarity, and internal use within a mechanical engineering business.