

## QuickPay – Payment Gateway

---

### Overview

Industry: FinTech / Payment Gateways

Purpose: To provide businesses with an easy-to-integrate, secure, and fast payment gateway with modern UI and developer tools.

Target Users: Startups, SMEs, and Enterprises needing reliable payment solutions.

### Features

- Signup / Login System
- Dashboard (with transaction summaries)
- Pricing Plans (with tier comparison)
- Developer API Page
- Live Transaction Simulator
- Developer documentation page
- Payment success/failure flow
- Responsive design
- Pricing tiers with comparison

## **Tech Stack**

Frontend: HTML, CSS, JavaScript

UI/UX: Figma

Charts: Chart.js / ApexCharts

Auth: JavaScript (basic localStorage/session logic)

Theme: Light/Dark mode toggle

## **Product & Pricing Strategy**

- Basic: 1.5% per domestic transaction
- Premium: 3% per international transaction
- Enterprise: Custom pricing (contact sales)
- Test/Sandbox: Free, with sandbox API keys

## **Secure Payment Flow**

1. User signs up and logs in
2. Business sets up API from developer dashboard
3. Test mode enabled using sandbox credentials
4. Real-time transactions appear on the dashboard
5. Payments confirmed via success/failure pages

## **Test Mode**

- Free for developers

- Generate sandbox keys
- Simulate transactions in test environment
- View mock transactions in dashboard

## **Developer API Documentation (Sample)**

Authentication:

POST /api/auth

Headers:

Authorization: Bearer <API\_KEY>

Create Payment:

POST /api/payment

Body:

```
{  
  "amount": 5000,  
  "currency": "INR",  
  "order_id": "ORD12345"  
}
```

Get Transaction Status:

GET /api/payment/status/:transaction\_id

## Dashboard Modules

- Total Revenue
- Fees Charged
- Transactions (Live feed)
- Income (Daily/Monthly)
- Payouts

## UI Components (CSS)

- Invoice Card
- Toggle Switch (Dark/Light mode)
- Signup Form (Animated and validated)
- Pricing Cards (Responsive and interactive)

## Live Transaction Simulator (JS Snippet)

```
setInterval(() => {  
    const fakeAmount = Math.floor(Math.random() *  
5000) + 100;  
    updateDashboard(fakeAmount);  
}, 5000);
```

## Secure Password Features

- Password strength validation
- Hide/show toggle
- Local storage session for login state

## Folder Structure

/QuickPay

└── index.html

└── signup.html

└── dashboard.html

└── pricing.html

└── developer.html

└── css/

|   └── style.css

└── js/

|   └── main.js

|   └── api.js

|   └── charts.js

└── assets/

    └── logos, icons, illustrations

## **Figma Designs**

- Homepage UI (Light & dark theme toggle)
- Pricing Tier Page
- Developer Page
- Payment Flow Screens

## **Future Scope**

- Backend with Node.js/Express
- Payment processing integration with Razorpay/Stripe
- Admin analytics dashboard
- OTP/2FA for login

## **Contact & Support**

- Email: [fayazshaikh1106@gmail.com](mailto:fayazshaikh1106@gmail.com)
- GitHub: [github.com/Fayaz5506](https://github.com/Fayaz5506)
- LinkedIn: [linkedin.com/in/fayaz-shaikh-b49950337/](https://www.linkedin.com/in/fayaz-shaikh-b49950337/)