

Candidate Assessment Project

Project Title: Subscribely — Subscription SaaS Platform

Goal: Build a Django + React + MySQL subscription app with UPI & card payments.

Required Tech Stack:

- 1 Backend: Django + Django REST Framework
- 2 Frontend: React
- 3 Database: MySQL
- 4 Payments: Razorpay/Stripe/PayU sandbox (UPI + card)
- 5 Auth: JWT (DRF SimpleJWT)
- 6 Deployment: Docker Compose

Core Features:

- 1 User signup/login with JWT authentication
- 2 Admin-managed plans (monthly/yearly, trial days, features)
- 3 Subscription flow with UPI and card payments
- 4 Webhook handling for payment success/failure
- 5 Access control for subscriber-only content
- 6 Subscription renewal & cancellation
- 7 MySQL migrations and proper schema
- 8 Security: no raw card storage, input validation
- 9 Admin dashboard or Django admin support
- 10 Documentation (README with setup and testing steps)

Data Model (suggested):

- 1 User: Django user
- 2 Plan: name, monthly_price, yearly_price, features, trial_days
- 3 Subscription: user, plan, status, start_date, end_date
- 4 Payment: user, subscription, amount, method, status, raw_response
- 5 WebhookEvent: event_type, payload, processed

API Endpoints (examples):

- 1 POST /api/auth/register/ - User registration
- 2 POST /api/auth/login/ - Login & JWT issue
- 3 GET /api/plans/ - List plans
- 4 POST /api/checkout/create-session/ - Create payment session
- 5 POST /api/webhooks/gateway/ - Receive payment gateway events
- 6 GET /api/subscriptions/ - List user subscriptions
- 7 POST /api/subscriptions/cancel/ - Cancel subscription

Evaluation Rubric (100 points):

- 1 Functionality: 40
- 2 Code quality & architecture: 20

- 3 Security & robustness: 15
- 4 UX & frontend: 15
- 5 Documentation & deliverables: 10

Deliverables:

- 1 GitHub repo with backend & frontend code
- 2 README with setup instructions & sandbox payment config
- 3 Postman/curl examples
- 4 Tests for key features
- 5 Optional demo video