

DueTap Backend Structure Checklist (Full)

1. Project Setup

- Select backend framework (Node.js/Express, Django, Laravel, etc.).
- Setup folder structure: controllers, models, routes, middlewares, services, utils.
- Configure `.env` for DB, JWT secrets, API keys (Razorpay, Stripe, SMTP, WhatsApp).
- Install dependencies: ORM/ODM (Sequelize/Mongoose/Prisma), bcrypt/argon2, JWT, validation (Joi/Yup), logging (Winston).
- Setup error handling middleware and response formatter.

2. Authentication & Accounts

- `POST /auth/signup` – Register (Super Admin or Client).
- `POST /auth/login` – Login with email/password.
- `POST /auth/forgot-password` – Send reset link/OTP.
- `POST /auth/reset-password` – Reset password.
- `POST /auth/refresh-token` – Issue new access token.
- `GET /me` – Get profile.
- `PATCH /me` – Update profile.
- `PATCH /me/password` – Change password.
- `DELETE /me` – Deactivate account.
- JWT authentication (access + refresh).
- Role-based access (Super Admin vs Client vs User).
- Encrypt passwords (bcrypt/argon2).
- Input validation & sanitization.

3. Database Design

- Super Admin: Admins, Clients.
- Client (Business Owner): Users, Templates, Subscriptions, Payments.
- Notifications, AnalyticsLogs, Settings.
- Background Jobs: Task Logs, Notification Logs.
- Security: API Keys, Audit Logs.

4. Super Admin APIs

- GET /admin/dashboard – Total clients, revenue, new clients, activity feed.
- GET /admin/clients – List clients (filters: status, search, pagination).
- POST /admin/clients – Create client.
- GET /admin/clients/:id – View client details.
- PATCH /admin/clients/:id – Update client.
- PATCH /admin/clients/:id/status – Activate/Suspend.
- DELETE /admin/clients/:id – Delete client.
- GET /admin/clients/:id/analytics – Revenue, subscriptions, notifications.
- GET /admin/clients/:id/notifications – Client activity feed.

5. Client (Business Owner) APIs

Dashboard

- GET /dashboard – Payments, users, subscriptions.
- GET /notifications – Activity feed.

Users

- POST /users – Add user.
- GET /users – List users (filters, pagination).
- GET /users/:id – User details + subscription history.
- PATCH /users/:id – Update user.
- DELETE /users/:id – Delete user.

Templates

- POST /templates – Create template.
- GET /templates – List/search templates.
- GET /templates/:id – Template details.
- PATCH /templates/:id – Update.
- DELETE /templates/:id – Delete.

Subscriptions

- POST /subscriptions – Create subscription.
- GET /subscriptions – List subscriptions (filters).
- GET /subscriptions/:id – Subscription details.
- PATCH /subscriptions/:id – Mark paid / cancel / overdue.
- POST /subscriptions/:id/resend – Resend notification.
- DELETE /subscriptions/:id – Cancel subscription.

Payments

- POST /payments – Record manual payment.
- GET /payments – List (filters).
- GET /payments/:id – Details.
- Webhook: /webhook/razorpay, /webhook/stripe.

Analytics

- GET /analytics/summary – Totals (payments, users, subscriptions).
- GET /analytics/payments – Trends.
- GET /analytics/notifications – Delivery rates.
- GET /analytics/templates – Top templates.

Settings

- GET /settings – Fetch saved API keys.
- POST /settings – Save/update API keys.

6. Background Jobs / Internal APIs

- POST /jobs/reminders/run – Trigger recurring reminders.
- POST /jobs/notifications/retry – Retry failed notifications.
- GET /jobs/logs – Logs of background tasks.
- Setup cron jobs for reminders.
- Setup queue system (Redis/RabbitMQ).
- Retry mechanism for failed sends.

7. System & Security APIs

- GET /health – System uptime, DB status.
- GET /logs/errors – Error logs (admin only).
- POST /security/audit – Log critical events.
- API rate limiting.
- Data encryption (API keys, sensitive info).
- GDPR compliance (data handling, deletion).
- Monitoring (Prometheus/Grafana).
- Error tracking (Sentry).

8. Testing & Deployment

- Unit tests (models, services).
- Integration tests (API endpoints).
- CI/CD pipeline setup.
- Automated migrations.
- Deployment configs (Docker, Kubernetes, etc.).
- Logging (ELK/Winston).
- Load testing.