📁 Remote SIM Gateway - Complete Folder Structure

🗂️ Root Project Structure

```
remote-sim-gateway/
├── 📱 android-app/              # Android bridge application
├── 🖥️ backend/                  # Go backend server
├── 🌐 frontend/                 # React web dashboard
├── 📚 docs/                     # Documentation
├── 🐳 deployments/              # Deployment configurations
├── 🧪 scripts/                  # Build and utility scripts
├── 📄 README.md                 # Project documentation
├── 📄 LICENSE                   # MIT License
├── 📄 .gitignore                # Git ignore rules
├── 📄 docker-compose.yml        # Multi-service deployment
├── 📄 docker-compose.dev.yml    # Development environment
└── 📄 CONTRIBUTING.md           # Contribution guidelines
```

📱 Android App Structure

```
android-app/
├── 📁 app/
│   ├── 📁 src/
│   │   ├── 📁 main/
│   │   │   ├── 📁 java/com/simgateway/
│   │   │   │   ├── 📄 MainActivity.kt          # Main activity
│   │   │   │   ├── 📄 SplashActivity.kt        # Splash screen
│   │   │   │   ├── 📁 services/
│   │   │   │   │   ├── 📄 SmsService.kt        # SMS handling
│   │   │   │   │   ├── 📄 CallService.kt       # Call handling
│   │   │   │   │   ├── 📄 ConnectionService.kt # Background service
│   │   │   │   │   └── 📄 NotificationService.kt # Notifications
│   │   │   │   ├── 📁 network/
│   │   │   │   │   ├── 📄 WebSocketClient.kt    # WebSocket connection
│   │   │   │   │   ├── 📄 ApiClient.kt         # HTTP client
│   │   │   │   │   └── 📄 NetworkStateManager.kt # Network monitoring
│   │   │   │   ├── 📁 models/
│   │   │   │   │   ├── 📄 SmsMessage.kt        # SMS data model
│   │   │   │   │   ├── 📄 CallLog.kt           # Call data model
│   │   │   │   │   ├── 📄 Command.kt           # Command data model
│   │   │   │   │   └── 📄 Device.kt            # Device info model
│   │   │   │   ├── 📁 utils/
│   │   │   │   │   ├── 📄 PermissionHelper.kt   # Permission management
```

```
│   │   │   │   │   ├── DeviceUtils.kt       # Device info utilities
│   │   │   │   │   ├── Constants.kt         # App constants
│   │   │   │   │   └── Logger.kt            # Logging utility
│   │   │   │   ├── receivers/
│   │   │   │   │   ├── SmsReceiver.kt       # SMS broadcast receiver
│   │   │   │   │   ├── CallReceiver.kt      # Call state receiver
│   │   │   │   │   └── BootReceiver.kt      # Auto-start on boot
│   │   │   │   └── ui/
│   │   │   │       ├── DashboardFragment.kt  # Main dashboard
│   │   │   │       ├── SettingsFragment.kt   # App settings
│   │   │   │       ├── LogsFragment.kt       # Activity logs
│   │   │   │       └── StatusFragment.kt     # Connection status
│   │   │   ├── res/
│   │   │   │   ├── layout/
│   │   │   │   │   ├── activity_main.xml
│   │   │   │   │   ├── fragment_dashboard.xml
│   │   │   │   │   ├── fragment_settings.xml
│   │   │   │   │   └── fragment_logs.xml
│   │   │   │   ├── values/
│   │   │   │   │   ├── strings.xml
│   │   │   │   │   ├── colors.xml
│   │   │   │   │   ├── styles.xml
│   │   │   │   │   └── config.xml           # App configuration
│   │   │   │   ├── drawable/
│   │   │   │   ├── mipmap/                  # App icons
│   │   │   │   └── xml/
│   │   │   │       └── network_security_config.xml
│   │   │   └── AndroidManifest.xml          # App permissions & config
│   │   └── test/                            # Unit tests
│   │       └── java/com/simgateway/
│   ├── build.gradle                         # App build configuration
│   └── proguard-rules.pro                   # Code obfuscation rules
├── build.gradle                             # Project build configuration
├── gradle.properties                        # Gradle properties
├── settings.gradle                          # Project settings
├── gradlew                                  # Gradle wrapper (Unix)
├── gradlew.bat                              # Gradle wrapper (Windows)
└── gradle/wrapper/                          # Gradle wrapper files
```

🖥️ Backend Server Structure

```
backend/
├── cmd/
│   └── server/
```

```
│   └── main.go                 # Application entry point
├── internal/
│   ├── config/
│   │   ├── config.go           # Configuration management
│   │   └── database.go         # Database configuration
│   ├── handlers/
│   │   ├── auth.go             # Authentication handlers
│   │   ├── sms.go              # SMS API handlers
│   │   ├── calls.go            # Call API handlers
│   │   ├── devices.go          # Device management
│   │   ├── users.go            # User management
│   │   ├── history.go          # Logs and history
│   │   └── health.go           # Health check endpoint
│   ├── middleware/
│   │   ├── auth.go             # JWT authentication
│   │   ├── cors.go             # CORS middleware
│   │   ├── ratelimit.go        # Rate limiting
│   │   ├── logging.go          # Request logging
│   │   └── validation.go       # Input validation
│   ├── models/
│   │   ├── user.go             # User model
│   │   ├── device.go           # Device model
│   │   ├── sms.go              # SMS model
│   │   ├── call.go             # Call model
│   │   ├── command.go          # Command model
│   │   └── response.go         # API response models
│   ├── repository/
│   │   ├── user_repo.go        # User database operations
│   │   ├── device_repo.go      # Device database operations
│   │   ├── sms_repo.go         # SMS database operations
│   │   └── call_repo.go        # Call database operations
│   ├── services/
│   │   ├── auth_service.go     # Authentication logic
│   │   ├── sms_service.go      # SMS business logic
│   │   ├── call_service.go     # Call business logic
│   │   ├── device_service.go   # Device management
│   │   └── notification_service.go  # Push notifications
│   ├── websocket/
│   │   ├── hub.go              # WebSocket hub
│   │   ├── client.go           # WebSocket client handler
│   │   ├── message.go          # Message types
│   │   └── pool.go             # Connection pool
│   └── utils/
│       ├── logger.go           # Logging utility
│       ├── jwt.go              # JWT utilities
│       ├── validator.go        # Input validation
│       └── crypto.go           # Encryption utilities
```

```
│   └── phone.go                # Phone number utilities
├── migrations/
│   ├── 001_create_users_table.up.sql
│   ├── 001_create_users_table.down.sql
│   ├── 002_create_devices_table.up.sql
│   ├── 002_create_devices_table.down.sql
│   ├── 003_create_sms_logs_table.up.sql
│   ├── 003_create_sms_logs_table.down.sql
│   ├── 004_create_call_logs_table.up.sql
│   └── 004_create_call_logs_table.down.sql
├── pkg/                        # Shared packages
│   └── errors/
│       └── errors.go           # Custom error types
├── tests/
│   ├── integration/
│   │   ├── auth_test.go
│   │   ├── sms_test.go
│   │   └── websocket_test.go
│   └── unit/
│       ├── handlers_test.go
│       ├── services_test.go
│       └── utils_test.go
├── go.mod                      # Go modules
├── go.sum                      # Go dependencies
├── .env.example                # Environment template
├── .air.toml                   # Hot reload configuration
├── Dockerfile                  # Container configuration
└── Makefile                    # Build automation
```

🌐 Frontend Web App Structure

```
frontend/
├── public/
│   ├── favicon.ico
│   ├── logo192.png
│   ├── logo512.png
│   └── manifest.json
├── src/
│   ├── components/
│   │   ├── common/
│   │   │   ├── Header.jsx          # App header
│   │   │   ├── Sidebar.jsx         # Navigation sidebar
│   │   │   ├── Footer.jsx          # App footer
│   │   │   ├── Loading.jsx         # Loading spinner
```

```
│   │   ├── Modal.jsx              # Reusable modal
│   │   └── Toast.jsx              # Notification toast
│   ├── auth/
│   │   ├── LoginForm.jsx          # Login component
│   │   ├── RegisterForm.jsx       # Registration component
│   │   └── ProtectedRoute.jsx     # Route protection
│   ├── dashboard/
│   │   ├── DashboardStats.jsx     # Stats overview
│   │   ├── DeviceStatus.jsx       # Device connection status
│   │   ├── RecentActivity.jsx     # Recent SMS/calls
│   │   └── QuickActions.jsx       # Quick send buttons
│   ├── sms/
│   │   ├── SmsForm.jsx            # Send SMS form
│   │   ├── SmsHistory.jsx         # SMS history table
│   │   ├── BulkSmsForm.jsx        # Bulk SMS sender
│   │   └── SmsTemplates.jsx       # Message templates
│   ├── calls/
│   │   ├── CallForm.jsx           # Make call form
│   │   ├── CallHistory.jsx        # Call history table
│   │   └── CallStatus.jsx         # Active call status
│   ├── devices/
│   │   ├── DeviceList.jsx         # Connected devices
│   │   ├── DeviceCard.jsx         # Individual device card
│   │   └── AddDevice.jsx          # Add new device
│   └── settings/
│       ├── UserSettings.jsx       # User preferences
│       ├── SecuritySettings.jsx   # Security options
│       └── ApiSettings.jsx        # API configuration
├── pages/
│   ├── index.js                   # Home/landing page
│   ├── login.js                   # Login page
│   ├── register.js                # Registration page
│   ├── dashboard.js               # Main dashboard
│   ├── sms.js                     # SMS management
│   ├── calls.js                   # Call management
│   ├── history.js                 # Combined history
│   ├── devices.js                 # Device management
│   ├── settings.js                # Settings page
│   └── _app.js                    # Next.js app wrapper
├── hooks/
│   ├── useAuth.js                 # Authentication hook
│   ├── useWebSocket.js            # WebSocket connection
│   ├── useApi.js                  # API calls hook
│   ├── useDeviceStatus.js         # Device status hook
│   └── useLocalStorage.js         # Local storage hook
├── utils/
│   ├── api.js                     # API client
```

```
│   │   ├── auth.js                # Auth utilities
│   │   ├── constants.js           # App constants
│   │   ├── formatting.js          # Data formatting
│   │   ├── validation.js          # Form validation
│   │   └── websocket.js           # WebSocket utilities
│   ├── styles/
│   │   ├── globals.css            # Global styles
│   │   ├── components.css         # Component styles
│   │   └── tailwind.css           # Tailwind imports
│   ├── contexts/
│   │   ├── AuthContext.js         # Auth state management
│   │   ├── WebSocketContext.js    # WebSocket state
│   │   └── ThemeContext.js        # Theme management
│   └── types/
│       ├── auth.ts                # Auth TypeScript types
│       ├── sms.ts                 # SMS TypeScript types
│       ├── calls.ts               # Call TypeScript types
│       └── api.ts                 # API TypeScript types
├── package.json                   # Dependencies & scripts
├── package-lock.json              # Locked dependencies
├── next.config.js                 # Next.js configuration
├── tailwind.config.js             # Tailwind CSS config
├── .env.local.example             # Environment template
├── .eslintrc.json                 # ESLint configuration
├── .gitignore                     # Git ignore rules
└── README.md                      # Frontend documentation
```

🖥️ Backend Server Structure

```
backend/
├── cmd/
│   └── server/
│       └── main.go                # Application entry point
├── internal/
│   ├── api/
│   │   ├── handlers/
│   │   │   ├── auth.go            # Authentication endpoints
│   │   │   ├── sms.go             # SMS API endpoints
│   │   │   ├── calls.go           # Call API endpoints
│   │   │   ├── devices.go         # Device management
│   │   │   ├── users.go           # User management
│   │   │   ├── history.go         # History endpoints
│   │   │   └── health.go          # Health check
│   │   ├── middleware/
```

```
│   │   ├── auth.go              # JWT validation
│   │   ├── cors.go              # CORS handling
│   │   ├── ratelimit.go         # Rate limiting
│   │   ├── logging.go           # Request logging
│   │   ├── recovery.go          # Panic recovery
│   │   └── validation.go        # Input validation
│   │   └── routes/
│   │       ├── api.go           # API route definitions
│   │       └── websocket.go     # WebSocket routes
│   ├── models/
│   │   ├── user.go              # User model & DB methods
│   │   ├── device.go            # Device model
│   │   ├── sms.go               # SMS log model
│   │   ├── call.go              # Call log model
│   │   ├── command.go           # Command model
│   │   └── base.go              # Base model with common fields
│   ├── services/
│   │   ├── auth_service.go      # Authentication business logic
│   │   ├── sms_service.go       # SMS business logic
│   │   ├── call_service.go      # Call business logic
│   │   ├── device_service.go    # Device management logic
│   │   ├── user_service.go      # User management logic
│   │   └── notification_service.go  # Push notifications
│   ├── repository/
│   │   ├── user_repo.go         # User database operations
│   │   ├── device_repo.go       # Device database operations
│   │   ├── sms_repo.go          # SMS database operations
│   │   ├── call_repo.go         # Call database operations
│   │   └── interfaces.go        # Repository interfaces
│   ├── websocket/
│   │   ├── hub.go               # WebSocket connection hub
│   │   ├── client.go            # WebSocket client handler
│   │   ├── message.go           # Message handling
│   │   ├── pool.go              # Connection pooling
│   │   └── commands.go          # Command processing
│   ├── database/
│   │   ├── connection.go        # Database connection
│   │   ├── migrations.go        # Migration runner
│   │   └── seeds.go             # Database seeding
│   └── utils/
│       ├── logger.go            # Structured logging
│       ├── jwt.go               # JWT token utilities
│       ├── validator.go         # Input validation
│       ├── crypto.go            # Encryption/hashing
│       ├── phone.go             # Phone number utilities
│       ├── response.go          # HTTP response helpers
│       └── errors.go            # Error handling
```

```
├── 📁 pkg/                          # Public packages
│   └── 📁 apierrors/
│       └── 📄 errors.go             # API error definitions
├── 📁 migrations/
│   ├── 📄 000001_create_users_table.up.sql
│   ├── 📄 000001_create_users_table.down.sql
│   ├── 📄 000002_create_devices_table.up.sql
│   ├── 📄 000002_create_devices_table.down.sql
│   ├── 📄 000003_create_sms_logs_table.up.sql
│   ├── 📄 000003_create_sms_logs_table.down.sql
│   ├── 📄 000004_create_call_logs_table.up.sql
│   └── 📄 000004_create_call_logs_table.down.sql
├── 📁 config/
│   ├── 📄 config.yaml               # Configuration file
│   ├── 📄 config.dev.yaml           # Development config
│   └── 📄 config.prod.yaml          # Production config
├── 📁 docs/
│   ├── 📄 api.yaml                  # OpenAPI specification
│   └── 📄 websocket.md              # WebSocket documentation
├── 📁 scripts/
│   ├── 📄 build.sh                  # Build script
│   ├── 📄 migrate.sh                # Database migration
│   ├── 📄 seed.sh                   # Database seeding
│   └── 📄 deploy.sh                 # Deployment script
├── 📄 go.mod                        # Go module definition
├── 📄 go.sum                        # Go dependency checksums
├── 📄 .env.example                  # Environment variables template
├── 📄 .gitignore                    # Git ignore file
├── 📄 Dockerfile                    # Docker container
├── 📄 Makefile                      # Build automation
└── 📄 README.md                     # Backend documentation
```

📚 Documentation Structure

```
docs/
├── 📄 SETUP.md                      # Setup instructions
├── 📄 API.md                        # API documentation
├── 📄 WEBSOCKET.md                  # WebSocket protocol
├── 📄 SECURITY.md                   # Security guidelines
├── 📄 DEPLOYMENT.md                 # Deployment guide
├── 📄 TROUBLESHOOTING.md            # Common issues
├── 📁 images/
│   ├── 📄 architecture-diagram.png
│   ├── 📄 web-dashboard.png
```

```
│   └── 📄 android-app.png
├── 📁 api-docs/
│   ├── 📄 openapi.yaml                # OpenAPI specification
│   └── 📄 postman-collection.json     # Postman collection
└── 📁 examples/
    ├── 📄 curl-examples.sh            # API examples
    ├── 📄 android-integration.md      # Android integration
    └── 📄 webhook-examples.md         # Webhook examples
```

🐳 Deployment Structure

```
deployments/
├── 📁 docker/
│   ├── 📄 Dockerfile.backend          # Backend container
│   ├── 📄 Dockerfile.frontend         # Frontend container
│   └── 📄 docker-compose.yml          # Multi-service deployment
├── 📁 kubernetes/
│   ├── 📄 namespace.yaml              # K8s namespace
│   ├── 📄 backend-deployment.yaml     # Backend deployment
│   ├── 📄 frontend-deployment.yaml    # Frontend deployment
│   ├── 📄 postgres-deployment.yaml    # Database deployment
│   ├── 📄 ingress.yaml                # Load balancer
│   └── 📄 secrets.yaml                # Secrets management
├── 📁 terraform/
│   ├── 📄 main.tf                     # Infrastructure as code
│   ├── 📄 variables.tf                # Terraform variables
│   └── 📄 outputs.tf                  # Terraform outputs
├── 📁 nginx/
│   ├── 📄 nginx.conf                  # Nginx configuration
│   └── 📄 ssl.conf                    # SSL configuration
└── 📁 systemd/
    ├── 📄 sim-gateway-backend.service     # Backend systemd service
    └── 📄 sim-gateway-frontend.service    # Frontend systemd service
```

🧪 Scripts Structure

```
scripts/
├── 📁 build/
│   ├── 📄 build-backend.sh            # Build Go backend
│   ├── 📄 build-frontend.sh           # Build React frontend
│   ├── 📄 build-android.sh            # Build Android APK
│   └── 📄 build-all.sh                # Build everything
```

```
├── 📁 database/
│   ├── 📄 setup-db.sh              # Database setup
│   ├── 📄 migrate.sh               # Run migrations
│   ├── 📄 seed.sh                  # Seed test data
│   └── 📄 backup.sh                # Database backup
├── 📁 development/
│   ├── 📄 dev-start.sh             # Start dev environment
│   ├── 📄 dev-stop.sh              # Stop dev environment
│   ├── 📄 test-all.sh              # Run all tests
│   └── 📄 lint-all.sh              # Code linting
├── 📁 deployment/
│   ├── 📄 deploy-staging.sh        # Deploy to staging
│   ├── 📄 deploy-production.sh     # Deploy to production
│   ├── 📄 rollback.sh              # Rollback deployment
│   └── 📄 health-check.sh          # Post-deploy health check
└── 📁 utilities/
    ├── 📄 generate-keys.sh         # Generate JWT keys
    ├── 📄 create-user.sh           # Create admin user
    ├── 📄 cleanup-logs.sh          # Clean old logs
    └── 📄 monitor.sh               # System monitoring
```

🔧 Configuration Files

Root Level Config Files

```
remote-sim-gateway/
├── 📄 docker-compose.yml           # Development environment
├── 📄 docker-compose.prod.yml      # Production environment
├── 📄 .env.example                 # Environment template
├── 📄 .gitignore                   # Git ignore patterns
├── 📄 Makefile                     # Build automation
└── 📄 VERSION                      # Current version
```

Example .gitignore

```gitignore
```

```
# Environment files
.env
.env.local
.env.production

# Build outputs
/backend/sim-gateway
/frontend/dist/
/frontend/.next/
/android-app/app/build/

# Dependencies
node_modules/
vendor/

# IDE files
.vscode/
.idea/
*.swp
*.swo

# Logs
*.log
logs/

# Database
*.db
*.sqlite

# Keys and certificates
*.key
*.pem
*.crt
```

🎯 Development Workflow

1. Start Development Environment

```bash
# Root directory
make dev-start
# or
docker-compose up -d
```

2. Development Order

1. **Backend First** - Core API and WebSocket
2. **Android App** - SIM bridge functionality
3. **Frontend** - Web dashboard
4. **Integration Testing** - End-to-end testing

3. Testing Structure

```bash
# Backend tests
cd backend && go test ./...

# Frontend tests
cd frontend && npm test

# Android tests
cd android-app && ./gradlew test

# Integration tests
./scripts/test-all.sh
```

📦 Build & Deployment

Quick Commands

```bash
# Build all components
make build-all

# Run development environment
make dev-start

# Deploy to production
make deploy-prod

# Run tests
make test-all

# Clean build artifacts
make clean
```

This folder structure provides:

- ✅ **Clear separation** of concerns
- ✅ **Scalable architecture** for future features
- ✅ **Easy deployment** with Docker/K8s
- ✅ **Maintainable codebase** with proper organization
- ✅ **Testing support** at all levels

🗃️ Database Structure

```
database/
├── 📁 migrations/
│   ├── 📄 000001_initial_schema.up.sql
│   ├── 📄 000001_initial_schema.down.sql
│   ├── 📄 000002_add_devices.up.sql
│   ├── 📄 000002_add_devices.down.sql
│   ├── 📄 000003_add_sms_logs.up.sql
│   ├── 📄 000003_add_sms_logs.down.sql
│   ├── 📄 000004_add_call_logs.up.sql
│   ├── 📄 000004_add_call_logs.down.sql
│   ├── 📄 000005_add_indexes.up.sql
│   └── 📄 000005_add_indexes.down.sql
├── 📁 seeds/
│   ├── 📄 users.sql                # Test users
│   ├── 📄 devices.sql              # Test devices
│   └── 📄 sample_data.sql          # Sample SMS/call data
└── 📁 schemas/
    ├── 📄 postgresql.sql           # PostgreSQL schema
    ├── 📄 mysql.sql                # MySQL schema (alternative)
    └── 📄 sqlite.sql               # SQLite schema (dev)
```

🧪 Testing Structure

Backend Tests

```
backend/tests/
├── 📁 unit/
│   ├── 📄 handlers_test.go         # Handler unit tests
│   ├── 📄 services_test.go         # Service unit tests
│   ├── 📄 models_test.go           # Model unit tests
│   ├── 📄 utils_test.go            # Utility unit tests
```

```
│   └── websocket_test.go              # WebSocket unit tests
├── integration/
│   ├── api_test.go                    # API integration tests
│   ├── auth_flow_test.go              # Auth flow tests
│   ├── sms_flow_test.go               # SMS flow tests
│   ├── call_flow_test.go              # Call flow tests
│   └── websocket_integration_test.go  # WebSocket integration
├── e2e/
│   ├── full_flow_test.go              # End-to-end tests
│   └── performance_test.go            # Load testing
├── mocks/
│   ├── mock_repository.go             # Repository mocks
│   ├── mock_websocket.go              # WebSocket mocks
│   └── mock_android.go                # Android app mocks
└── fixtures/
    ├── users.json                     # Test user data
    ├── devices.json                   # Test device data
    └── commands.json                  # Test command data
```

Frontend Tests

```
frontend/tests/
├── components/
│   ├── SmsForm.test.jsx               # SMS form tests
│   ├── CallForm.test.jsx              # Call form tests
│   ├── DeviceStatus.test.jsx          # Device status tests
│   └── Dashboard.test.jsx             # Dashboard tests
├── hooks/
│   ├── useAuth.test.js                # Auth hook tests
│   ├── useWebSocket.test.js           # WebSocket hook tests
│   └── useApi.test.js                 # API hook tests
├── utils/
│   ├── api.test.js                    # API utility tests
│   ├── validation.test.js             # Validation tests
│   └── formatting.test.js             # Formatting tests
├── e2e/
│   ├── login-flow.spec.js             # Login E2E tests
│   ├── sms-flow.spec.js               # SMS E2E tests
│   └── call-flow.spec.js              # Call E2E tests
└── __mocks__/
    ├── api.js                         # API mocks
    └── websocket.js                   # WebSocket mocks
```

Android Tests

```
android-app/app/src/test/
├── 📁 java/com/simgateway/
│   ├── 📁 services/
│   │   ├── 📄 SmsServiceTest.kt              # SMS service tests
│   │   ├── 📄 CallServiceTest.kt             # Call service tests
│   │   └── 📄 ConnectionServiceTest.kt       # Connection tests
│   ├── 📁 network/
│   │   ├── 📄 WebSocketClientTest.kt         # WebSocket tests
│   │   └── 📄 ApiClientTest.kt               # API client tests
│   └── 📁 utils/
│       ├── 📄 PermissionHelperTest.kt        # Permission tests
│       └── 📄 DeviceUtilsTest.kt             # Device utility tests
└── 📁 androidTest/                           # Instrumented tests
    └── 📁 java/com/simgateway/
        ├── 📄 SmsIntegrationTest.kt          # SMS integration tests
        ├── 📄 CallIntegrationTest.kt         # Call integration tests
        └── 📄 WebSocketIntegrationTest.kt    # WebSocket integration
```

📊 Monitoring & Logs Structure

```
monitoring/
├── 📁 prometheus/
│   ├── 📄 prometheus.yml                     # Prometheus config
│   ├── 📄 alert-rules.yml                    # Alert rules
│   └── 📄 recording-rules.yml                # Recording rules
├── 📁 grafana/
│   ├── 📄 dashboards/
│   │   ├── 📄 backend-metrics.json           # Backend dashboard
│   │   ├── 📄 android-metrics.json           # Android app dashboard
│   │   └── 📄 system-overview.json           # System overview
│   └── 📄 datasources/
│       └── 📄 prometheus.yaml                # Prometheus datasource
├── 📁 logs/
│   ├── 📄 backend.log                        # Backend application logs
│   ├── 📄 websocket.log                      # WebSocket connection logs
│   ├── 📄 sms.log                            # SMS activity logs
│   ├── 📄 calls.log                          # Call activity logs
│   └── 📄 security.log                       # Security event logs
└── 📁 alerts/
    ├── 📄 slack-webhook.yaml                 # Slack notifications
```

```
├── email-alerts.yaml          # Email notifications
└── pagerduty.yaml             # PagerDuty integration
```

🔒 Security & Secrets Structure

```
security/
├── certificates/
│   ├── server.crt             # SSL certificate
│   ├── server.key             # SSL private key
│   ├── ca.crt                 # Certificate authority
│   └── client.crt             # Client certificate
├── keys/
│   ├── jwt-private.key        # JWT signing key
│   ├── jwt-public.key         # JWT verification key
│   ├── encryption.key         # Data encryption key
│   └── api-keys.enc           # Encrypted API keys
└── policies/
    ├── rate-limits.yaml       # Rate limiting policies
    ├── firewall-rules.yaml    # Firewall configuration
    └── access-control.yaml    # Access control policies
```

📝 Additional Project Files

Root Level Files

```
remote-sim-gateway/
├── .github/
│   ├── workflows/
│   │   ├── ci.yml             # Continuous Integration
│   │   ├── cd.yml             # Continuous Deployment
│   │   └── security-scan.yml  # Security scanning
│   ├── ISSUE_TEMPLATE/
│   │   ├── bug_report.md      # Bug report template
│   │   └── feature_request.md # Feature request template
│   └── pull_request_template.md # PR template
├── CHANGELOG.md               # Version changelog
├── SECURITY.md                # Security policy
├── CODE_OF_CONDUCT.md         # Code of conduct
├── CONTRIBUTING.md            # Contribution guidelines
├── LICENSE                    # MIT License
└── .editorconfig              # Editor configuration
```

🛠️ Development Tools Configuration

Backend Tools

```
backend/
├── 📄 .air.toml                # Hot reload config
├── 📄 .golangci.yml            # Go linter config
├── 📄 Makefile                 # Build automation
└── 📁 tools/
    ├── 📄 generate-docs.go     # API docs generator
    ├── 📄 migrate.go           # Migration tool
    └── 📄 seed.go              # Database seeder
```

Frontend Tools

```
frontend/
├── 📄 .eslintrc.json           # ESLint configuration
├── 📄 .prettierrc              # Prettier configuration
├── 📄 tailwind.config.js       # Tailwind CSS config
├── 📄 next.config.js           # Next.js configuration
├── 📄 jest.config.js           # Jest testing config
├── 📄 playwright.config.ts     # E2E testing config
└── 📄 tsconfig.json            # TypeScript config
```

Android Tools

```
android-app/
├── 📄 .editorconfig            # Editor configuration
├── 📄 detekt.yml               # Kotlin linter config
├── 📄 ktlint.gradle            # Code formatting
└── 📁 gradle/
    └── 📄 libs.versions.toml   # Version catalog
```

🚀 Getting Started Commands

Quick Setup (Recommended)

```bash
```

```
# 1. Clone repository
git clone https://github.com/yourusername/remote-sim-gateway.git
cd remote-sim-gateway

# 2. Start everything with Docker
make dev-start

# 3. Open in browser
open http://localhost:3000
```

Manual Setup

bash

```
# 1. Backend
cd backend
cp .env.example .env
go mod download
make migrate-up
go run cmd/server/main.go

# 2. Frontend (new terminal)
cd frontend
npm install
cp .env.local.example .env.local
npm run dev

# 3. Android (Android Studio)
cd android-app
# Open in Android Studio
# Build and install on device
```

📋 File Creation Checklist

✅ Must Create First (Core Files)

- [ ] backend/cmd/server/main.go
- [ ] backend/internal/config/config.go
- [ ] android-app/app/src/main/java/com/simgateway/MainActivity.kt
- [ ] frontend/pages/index.js
- [ ] docker-compose.yml

✅ Essential Configuration Files

- [ ] `backend/.env.example`
- [ ] `frontend/.env.local.example`
- [ ] `android-app/app/src/main/AndroidManifest.xml`
- [ ] `.gitignore`
- [ ] `README.md`

✅ Database Files

- [ ] `backend/migrations/000001_initial_schema.up.sql`
- [ ] Database connection configuration
- [ ] Seed data files

✅ Security Files

- [ ] JWT key generation
- [ ] SSL certificates
- [ ] Environment variable templates

📱 Mobile App File Details

Key Android Files to Create

```
android-app/app/src/main/java/com/simgateway/
├── 📄 SimGatewayApplication.kt          # Application class
├── 📄 MainActivity.kt                   # Main activity
├── 📄 SettingsActivity.kt               # Settings screen
├── 📄 PermissionActivity.kt             # Permission requests
├── 📄 services/
│   ├── 📄 ForegroundService.kt          # Keep-alive service
│   ├── 📄 SmsHandlerService.kt          # SMS operations
│   ├── 📄 CallHandlerService.kt         # Call operations
│   └── 📄 WebSocketService.kt           # WebSocket management
├── 📄 network/
│   ├── 📄 WebSocketManager.kt           # WebSocket connection
│   ├── 📄 RetryPolicy.kt                # Connection retry logic
│   ├── 📄 HeartbeatManager.kt           # Keep connection alive
│   └── 📄 MessageQueue.kt               # Offline message queue
├── 📄 database/
│   ├── 📄 AppDatabase.kt                # Room database
│   ├── 📄 LogDao.kt                     # Local logs DAO
│   └── 📄 entities/
│       ├── 📄 LocalSmsLog.kt            # Local SMS log entity
│       └── 📄 LocalCallLog.kt           # Local call log entity
```

```
└── 📄 utils/
    ├── PhoneNumberValidator.kt          # Phone validation
    ├── EncryptionHelper.kt              # Data encryption
    ├── DeviceIdGenerator.kt             # Unique device ID
    └── BatteryOptimizationHelper.kt     # Battery optimization
```

🌐 Frontend Component Hierarchy

Detailed Component Structure

```
frontend/src/components/
├── 📁 layout/
│   ├── AppLayout.jsx                    # Main app layout
│   ├── Header.jsx                       # Top navigation
│   ├── Sidebar.jsx                      # Side navigation
│   ├── Footer.jsx                       # Footer component
│   └── Breadcrumb.jsx                   # Navigation breadcrumb
├── 📁 auth/
│   ├── LoginForm.jsx                    # Login form
│   ├── RegisterForm.jsx                 # Registration form
│   ├── ForgotPassword.jsx               # Password reset
│   ├── ProtectedRoute.jsx               # Route protection
│   └── AuthGuard.jsx                    # Authentication guard
├── 📁 dashboard/
│   ├── DashboardOverview.jsx            # Main dashboard
│   ├── StatsCards.jsx                   # Statistics cards
│   ├── RecentActivity.jsx               # Recent SMS/calls
│   ├── DeviceStatus.jsx                 # Device connection status
│   ├── QuickActions.jsx                 # Quick action buttons
│   └── AlertsBanner.jsx                 # System alerts
├── 📁 sms/
│   ├── SmsComposer.jsx                  # SMS composition form
│   ├── SmsHistory.jsx                   # SMS history table
│   ├── SmsTemplates.jsx                 # Message templates
│   ├── BulkSmsForm.jsx                  # Bulk SMS sender
│   ├── SmsStatus.jsx                    # SMS delivery status
│   └── ContactPicker.jsx                # Contact selection
├── 📁 calls/
│   ├── CallDialer.jsx                   # Call dialer interface
│   ├── CallHistory.jsx                  # Call history table
│   ├── ActiveCallStatus.jsx             # Live call status
│   ├── CallDuration.jsx                 # Call timer
│   └── CallControls.jsx                 # Call control buttons
├── 📁 devices/
│   ├── DeviceList.jsx                   # List of devices
```

```
│   ├── DeviceCard.jsx              # Individual device card
│   ├── AddDeviceModal.jsx          # Add device modal
│   ├── DeviceSettings.jsx          # Device configuration
│   └── ConnectionIndicator.jsx     # Connection status indicator
├── settings/
│   ├── UserProfile.jsx             # User profile settings
│   ├── SecuritySettings.jsx        # Security preferences
│   ├── NotificationSettings.jsx    # Notification preferences
│   ├── ApiKeyManagement.jsx        # API key management
│   └── ExportData.jsx              # Data export options
├── history/
│   ├── CombinedHistory.jsx         # Combined SMS/call history
│   ├── HistoryFilter.jsx           # Filter controls
│   ├── HistoryTable.jsx            # History data table
│   ├── ExportHistory.jsx           # Export functionality
│   └── SearchHistory.jsx           # Search functionality
└── ui/
    ├── Button.jsx                  # Reusable button
    ├── Input.jsx                   # Form input
    ├── Table.jsx                   # Data table
    ├── Modal.jsx                   # Modal dialog
    ├── Toast.jsx                   # Notification toast
    ├── Spinner.jsx                 # Loading spinner
    ├── Badge.jsx                   # Status badge
    └── Card.jsx                    # Card container
```

🔧 Configuration Files Structure

Environment Files

```
├── .env.example                    # Template for all environments
├── .env.development                # Development environment
├── .env.staging                    # Staging environment
├── .env.production                 # Production environment
└── config/
    ├── development.yaml            # Development config
    ├── staging.yaml                # Staging config
    ├── production.yaml             # Production config
    └── default.yaml                # Default values
```

Docker Configuration

```
deployments/docker/
├── Dockerfile.backend              # Backend container
```

```
├── 📄 Dockerfile.frontend          # Frontend container
├── 📄 Dockerfile.nginx             # Nginx reverse proxy
├── 📄 docker-compose.yml           # Development environment
├── 📄 docker-compose.prod.yml      # Production environment
├── 📄 docker-compose.test.yml      # Testing environment
└── 📁 configs/
    ├── 📄 nginx.conf               # Nginx configuration
    ├── 📄 postgres.conf            # PostgreSQL configuration
    └── 📄 redis.conf               # Redis configuration
```

📊 Sample File Sizes (Estimated)

Component	Files	Lines of Code	Size
Backend	~45 files	~8,000 LOC	~2.5 MB
Frontend	~60 files	~12,000 LOC	~15 MB (with node_modules)
Android	~35 files	~6,000 LOC	~5 MB
Documentation	~15 files	~3,000 lines	~500 KB
Configuration	~25 files	~1,000 lines	~200 KB
Total	**~180 files**	**~30,000 LOC**	**~25 MB**

🎯 Priority File Creation Order

Phase 1: Core Backend (Days 1-2)

bash

1. backend/cmd/server/main.go
2. backend/internal/config/config.go
3. backend/internal/models/user.go
4. backend/internal/handlers/auth.go
5. backend/internal/websocket/hub.go

Phase 2: Android App (Days 3-4)

bash

1. android-app/app/src/main/AndroidManifest.xml
2. android-app/app/src/main/java/com/simgateway/MainActivity.kt
3. android-app/app/src/main/java/com/simgateway/services/SmsService.kt
4. android-app/app/src/main/java/com/simgateway/network/WebSocketClient.kt

Phase 3: Frontend (Days 5-6)

bash

1. frontend/pages/_app.js
2. frontend/pages/index.js
3. frontend/components/auth/LoginForm.jsx
4. frontend/components/sms/SmsForm.jsx
5. frontend/utils/api.js

Phase 4: Integration (Days 7-8)

bash

1. docker-compose.yml
2. Database migrations
3. Testing files
4. Documentation updates

🚀 Quick Start File Generation

Want to generate the entire folder structure quickly?

bash

```
# Create all directories
mkdir -p remote-sim-gateway/{android-app,backend,frontend,docs,deployments,scripts}
mkdir -p remote-sim-gateway/backend/{cmd/server,internal/{api/handlers,middleware,models,services,repository,we
mkdir -p remote-sim-gateway/frontend/{src/{components/{auth,dashboard,sms,calls,devices,settings,ui},pages,hook
mkdir -p remote-sim-gateway/android-app/app/src/main/{java/com/simgateway/{services,network,models,utils,recei

# Create basic files
touch remote-sim-gateway/{README.md,LICENSE,.gitignore,docker-compose.yml}
touch remote-sim-gateway/backend/{go.mod,.env.example,Makefile}
touch remote-sim-gateway/frontend/{package.json,.env.local.example,next.config.js}
```

This structure gives you:

- ✅ **Professional organization**
- ✅ **Clear separation** of concerns
- ✅ **Scalable architecture**
- ✅ **Easy navigation** for developers
- ✅ **Deployment ready** structure

- ✅ **Testing support** built-in

Ready to start building? Begin with the backend server structure and work your way up!