

# Media Authentication & Invisible Image Watermarking Platform

## Project Directory Structure



```

├── services/                                # Business logic
│   ├── auth_service.go
│   ├── user_service.go
│   ├── org_service.go
│   └── watermark_service.go
├── watermark/                              # Core watermarking engine
│   ├── engine/
│   │   ├── embedder.go                    # Main embedding orchestrator
│   │   ├── extractor.go                  # Main extraction orchestrator
│   │   └── processor.go                  # Shared processing utilities
│   ├── transform/
│   │   ├── dwt.go                        # Discrete Wavelet Transform
│   │   ├── dct.go                        # Discrete Cosine Transform
│   │   ├── colorspace.go                 # RGB <=> YCbCr conversion
│   │   └── inverse.go                   # IDWT, IDCT implementations
│   ├── qim/
│   │   ├── encoder.go                   # QIM embedding logic
│   │   └── decoder.go                   # QIM extraction logic
│   ├── fingerprint/
│   │   ├── generator.go                 # Perceptual fingerprint generation
│   │   └── comparator.go                # Fingerprint comparison
│   └── payload/
│       ├── builder.go                   # Payload assembly
│       ├── parser.go                    # Payload extraction
│       └── crypto.go                    # Encryption/decryption
├── storage/                                # Storage interfaces
│   ├── s3.go                            # S3 bucket operations
│   ├── redis.go                          # Redis cache operations
│   └── interface.go                      # Storage interface definition
├── database/
│   ├── postgres.go                      # PostgreSQL connection
│   └── migrations/                      # Database migrations
│       ├── 001_create_users.sql
│       ├── 002_create_orgs.sql
│       ├── 003_create_apikeys.sql
│       ├── 004_create_images.sql
│       └── 005_create_watermark_logs.sql
├── config/                                # Configuration management
└── config.go

```

- └─ utils/ # Utility functions
  - └─ jwt.go
  - └─ hash.go
  - └─ validation.go
  - └─ logger.go

- └─ pkg/ # Public shared libraries
  - └─ errors/
    - └─ errors.go
  - └─ response/
    - └─ response.go

- └─ scripts/
  - └─ seed.go # Database seeding
  - └─ migrate.sh # Migration runner

- └─ tests/ # Test files
  - └─ integration/
    - └─ auth\_test.go
    - └─ embed\_test.go
    - └─ extract\_test.go
  - └─ unit/
    - └─ dwt\_test.go
    - └─ dct\_test.go
    - └─ qim\_test.go

- └─ docker/
  - └─ Dockerfile
  - └─ docker-compose.yml

- └─ k8s/ # Kubernetes manifests
  - └─ deployment.yaml
  - └─ service.yaml
  - └─ ingress.yaml
  - └─ configmap.yaml

- └─ .env.example
- └─ .gitignore
- └─ go.mod
- └─ go.sum
- └─ Makefile
- └─ README.md

- └─ frontend/ # React.js Frontend



- └─ ExtractWatermark.jsx
- └─ ImageHistory.jsx
- └─ AdminPanel.jsx
- └─ NotFound.jsx

- └─ services/ # API service layer
  - └─ api.js # Axios instance configuration
  - └─ authService.js
  - └─ userService.js
  - └─ watermarkService.js
  - └─ adminService.js

- └─ hooks/ # Custom React hooks
  - └─ useAuth.js
  - └─ useFileUpload.js
  - └─ usePolling.js
  - └─ useLocalStorage.js

- └─ context/ # React Context providers
  - └─ AuthContext.jsx
  - └─ ThemeContext.jsx

- └─ utils/ # Utility functions
  - └─ validators.js
  - └─ formatters.js
  - └─ constants.js
  - └─ helpers.js

- └─ styles/ # Global styles
  - └─ global.css
  - └─ variables.css
  - └─ themes.css

- └─ assets/ # Static assets
  - └─ images/
  - └─ icons/

- └─ App.jsx # Root component
- └─ index.jsx # Entry point
- └─ routes.jsx # Route definitions

- └─ .env.example
- └─ .gitignore
- └─ package.json
- └─ package-lock.json
- └─ vite.config.js # or webpack.config.js
- └─ tailwind.config.js # If using Tailwind

```
| | — postcss.config.js
| | — README.md
|
| — docs/                                # Documentation
| | — api/
| | | — authentication.md
| | | — endpoints.md
| | | — examples.md
| |
| | — architecture/
| | | — system-design.md
| | | — database-schema.md
| | | — watermarking-pipeline.md
| |
| | — deployment/
| | | — docker-setup.md
| | | — kubernetes-setup.md
|
| — .gitignore
| — docker-compose.yml                  # Full stack orchestration
| — README.md
| — LICENSE
```

## Key Design Principles

### Backend (Go/Fiber)

- **Separation of Concerns:** Clear distinction between handlers, services, and repositories
- **Modular Watermarking Engine:** Isolated watermark processing logic for maintainability
- **Testability:** Unit and integration tests separated by functionality
- **Scalability:** Stateless design for horizontal scaling

### Frontend (React)

- **Component-Based Architecture:** Reusable UI components
- **Service Layer:** Clean API abstraction for backend communication
- **Custom Hooks:** Encapsulated business logic for state management
- **Context Providers:** Global state management for auth and theme

### Infrastructure

- **Containerization:** Docker support for both services
- **Orchestration:** Kubernetes manifests for production deployment

- **Documentation:** Comprehensive API and architecture docs