

Media Authentication & Invisible Image Watermarking Platform

Project Directory Structure

```
media-watermark-platform/
    |
    +-- backend/          # Go (Fiber) Backend
        |
        +-- cmd/
            +-- server/
                +-- main.go      # Application entry point
        |
        +-- internal/       # Private application code
            +-- api/
                +-- handlers/
                    +-- auth.go
                    +-- user.go
                    +-- organization.go
                    +-- embed.go
                    +-- extract.go
                    +-- admin.go
            +-- middleware/   # HTTP middleware
                +-- auth.go
                +-- ratelimit.go
                +-- validation.go
                +-- audit.go
                +-- cors.go
            +-- routes/        # Route definitions
                +-- routes.go
        |
        +-- models/          # Data models
            +-- user.go
            +-- organization.go
            +-- apikey.go
            +-- image.go
            +-- watermark_log.go
        |
        +-- repository/     # Database access layer
            +-- user_repo.go
            +-- org_repo.go
            +-- apikey_repo.go
            +-- image_repo.go
            +-- audit_repo.go
```

```
|- 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
```

```
|- public/
  |- index.html
  |- favicon.ico
  |- manifest.json

|- src/
  |- components/          # Reusable components
    |- common/
      |- Button.jsx
      |- Input.jsx
      |- Modal.jsx
      |- LoadingSpinner.jsx
      |- Alert.jsx

    |- layout/
      |- Header.jsx
      |- Sidebar.jsx
      |- Footer.jsx
      |- Layout.jsx

    |- auth/
      |- LoginForm.jsx
      |- RegisterForm.jsx
      |- ProtectedRoute.jsx

    |- watermark/
      |- ImageUploader.jsx
      |- ProcessingStatus.jsx
      |- ExtractionResult.jsx
      |- TamperScore.jsx

    |- dashboard/
      |- StatsCard.jsx
      |- RecentActivity.jsx
      |- QuickActions.jsx

    |- admin/
      |- UserManagement.jsx
      |- APIKeyManager.jsx
      |- AuditLogViewer.jsx

  |- pages/                # Page components
    |- Home.jsx
    |- Login.jsx
    |- Register.jsx
    |- Dashboard.jsx
    |- EmbedWatermark.jsx
```

```
    |   |   |   └── 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/
    |
    └── routes             # Route definitions
        ├── App.jsx          # Root component
        ├── index.jsx         # Entry point
        └── routes.jsx
    |
    └── .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