

YieldMax Complete Project Structure

Root Directory Structure

```
yieldmax/  
├── frontend/           # Next.js/React frontend application  
├── contracts/          # Smart contracts and blockchain code  
├── backend/            # API backend (if needed)  
├── scripts/            # Deployment and utility scripts  
├── docs/               # Documentation  
├── .github/            # GitHub workflows  
├── docker-compose.yml  # Docker setup for development  
├── docker-compose.test.yml # Docker setup for testing  
├── .gitignore          # Git ignore file  
├── README.md           # Project documentation  
└── package.json        # Root package.json for workspaces
```

FRONTEND FOLDER STRUCTURE

```

frontend/
├─ src/
│   ├─ app/                                # Next.js 13+ app directory
│   │   ├─ layout.tsx                      # Root layout with providers
│   │   ├─ page.tsx                       # Home page (imports YieldMaxDashboard)
│   │   ├─ portfolio/
│   │   │   └─ page.tsx                   # Portfolio page
│   │   ├─ markets/
│   │   │   └─ page.tsx                   # Markets page
│   │   ├─ history/
│   │   │   └─ page.tsx                   # History page
│   │   └─ api/                           # API routes
│   │       ├─ yields/
│   │       │   └─ route.ts               # GET /api/yields
│   │       ├─ portfolio/
│   │       │   └─ route.ts               # GET /api/portfolio
│   │       └─ transactions/
│   │           └─ route.ts               # GET /api/transactions
│   └─ components/                         # React components
│       ├─ Dashboard/
│       │   ├─ YieldMaxDashboard.tsx      # Main dashboard (from artifact)
│       │   ├─ StatCard.tsx               # Stat card component
│       │   ├─ YieldComparison.tsx        # Yield comparison component
│       │   ├─ PortfolioOverview.tsx      # Portfolio overview
│       │   ├─ TransactionHistory.tsx     # Transaction history
│       │   └─ OptimizationSuggestions.tsx
│       └─ UI/
│           ├─ Button.tsx                 # Button component
│           ├─ Card.tsx                   # Card component
│           ├─ Input.tsx                  # Input component
│           ├─ Modal.tsx                  # Modal component
│           └─ Badge.tsx                   # Badge component

```

```

| | | └─ Spinner.tsx          # Loading spinner
| | |   └─ Tooltip.tsx        # Tooltip component
| | └─ Layout/
| | | └─ Sidebar.tsx          # Sidebar navigation
| | | └─ Header.tsx           # Header component
| | |   └─ MobileMenu.tsx     # Mobile menu
| | └─ Wallet/
| | | └─ WalletButton.tsx     # Wallet connection button
| | | └─ NetworkSelector.tsx  # Network switcher
| | |   └─ GasPriceIndicator.tsx # Gas price display
| | └─ StyledComponents.ts    # All styled components
|
| └─ hooks/                   # Custom React hooks
| | └─ useWeb3.ts             # Web3 connection hook
| | └─ useRealTimeData.ts     # Real-time data hook
| | └─ usePortfolio.ts        # Portfolio management
| | └─ useTransactions.ts     # Transaction handling
| |   └─ useOptimization.ts   # Yield optimization
|
| └─ utils/                   # Utility functions
| | └─ format.ts              # Number/currency formatting
| | └─ constants.ts           # App constants
| | └─ chains.ts              # Chain configurations
| | └─ contracts.ts           # Contract addresses/ABIs
| |   └─ helpers.ts           # Helper functions
|
| └─ styles/                  # Global styles
| | └─ globals.css            # Global CSS
| |   └─ theme.ts             # Theme configuration
|
| └─ types/                   # TypeScript types
| | └─ index.ts               # Main types export
| | └─ portfolio.ts           # Portfolio types

```

```
| | | └─ transactions.ts      # Transaction types
| | |   └─ web3.ts           # Web3 types
| | |
| | └─ design-system/        # Design system
| |   └─ index.ts            # Main export (yieldmax-ui-system.ts content)
| |   └─ tokens.ts           # Design tokens
| |   └─ components.ts       # Component exports
| |
| └─ public/                  # Static assets
|   └─ favicon.ico
|   └─ logo.svg
|   └─ aave-logo.svg
|   └─ compound-logo.svg
|   └─ morpho-logo.svg
|   └─ spark-logo.svg
|
| └─ tests/                    # Test files
|   └─ e2e/                    # E2E tests
|     └─ wallet-connection.spec.ts
|     └─ deposit-flow.spec.ts
|     └─ cross-chain.spec.ts
|     └─ performance.spec.ts
|     └─ mobile.spec.ts
|     └─ error-handling.spec.ts
|     └─ helpers/
|       └─ setup.ts           # E2E test helpers
|   └─ unit/                  # Unit tests
|     └─ components/
|     └─ hooks/
|     └─ utils/
|
| └─ package.json              # Frontend dependencies
| └─ tsconfig.json            # TypeScript config
```

— next.config.js	# Next.js config
— jest.config.js	# Jest config
— jest.setup.js	# Jest setup
— playwright.config.ts	# Playwright config
└ .env.example	# Environment variables template

CONTRACTS FOLDER STRUCTURE

```

contracts/
├─ contracts/                # Solidity contracts
│  ├─ core/
│  │  ├─ YieldMaxVault.sol   # Main vault contract
│  │  ├─ StrategyEngine.sol  # Strategy engine
│  │  └─ CrossChainRouter.sol # CCIP router
│  ├─ interfaces/
│  │  ├─ IYieldMaxVault.sol
│  │  ├─ IStrategyEngine.sol
│  │  ├─ ICrossChainRouter.sol
│  │  └─ IChainlinkOracle.sol
│  ├─ libraries/
│  │  ├─ SafeMath.sol
│  │  ├─ RebalanceLib.sol
│  │  └─ CCIPHelpers.sol
│  ├─ mocks/                 # Mock contracts for testing
│  │  ├─ MockERC20.sol
│  │  ├─ MockCCIPRouter.sol
│  │  ├─ MockChainlinkOracle.sol
│  │  └─ MockProtocol.sol
│  └─ attacks/               # Security test contracts
│      ├─ ReentrancyAttacker.sol
│      └─ MaliciousReceiver.sol
│
├─ scripts/                  # Deployment scripts
│  ├─ deploy.js               # Main deployment
│  ├─ deploy-testnet.js       # Testnet deployment
│  ├─ verify.js               # Contract verification
│  └─ setup-ccip.js           # CCIP setup
│
├─ test/                     # Contract tests
│  ├─ YieldMaxVault.test.ts   # Vault tests
│  └─ CrossChain.test.ts      # Cross-chain tests

```

```
| | └─ GasOptimization.test.ts # Gas optimization tests
| | └─ Security.test.ts      # Security tests
| | └─ EdgeCases.test.ts    # Edge case tests
| | └─ helpers/
| |   └─ index.js           # Test helpers
| |   └─ utils/
| |     └─ chainlink-client.js # Chainlink mock
| |     └─ ccip-simulator.js  # CCIP simulator
|
└─ deployments/              # Deployment artifacts
  └─ mainnet/
  └─ arbitrum/
  └─ polygon/
  └─ optimism/

└─ package.json              # Contract dependencies
└─ hardhat.config.js         # Hardhat configuration
└─ .env.example              # Environment template
└─ README.md                 # Contract documentation
```

BACKEND FOLDER STRUCTURE (Optional API)

```
backend/
├─ src/
│   ├── controllers/           # API controllers
│   │   ├── yields.controller.ts
│   │   ├── portfolio.controller.ts
│   │   └── optimization.controller.ts
│   ├── services/             # Business logic
│   │   ├── chainlink.service.ts
│   │   ├── optimization.service.ts
│   │   └── blockchain.service.ts
│   ├── models/              # Data models
│   │   ├── User.ts
│   │   ├── Position.ts
│   │   └── Transaction.ts
│   ├── routes/              # API routes
│   │   ├── index.ts
│   │   └── v1/
│   │       ├── yields.ts
│   │       └── portfolio.ts
│   ├── middleware/          # Express middleware
│   │   ├── auth.ts
│   │   ├── cors.ts
│   │   └── rateLimit.ts
│   ├── config/              # Configuration
│   │   ├── database.ts
│   │   └── chains.ts
│   └── index.ts              # Entry point
├─ tests/                    # API tests
├─ package.json
├─ tsconfig.json
└─ .env.example
```


SCRIPTS FOLDER STRUCTURE

```
scripts/
├─ demo/                # Demo scripts
│   ├── setup-demo-data.js # Setup demo data
│   ├── run-demo.js       # Run demo flow
│   └─ reset-demo.js      # Reset demo state
├─ deployment/          # Deployment automation
│   ├── deploy-all-chains.js # Multi-chain deployment
│   ├── verify-all.js      # Verify all contracts
│   └─ setup-infrastructure.js
└─ monitoring/          # Monitoring scripts
    ├── check-yields.js     # Monitor yield changes
    └─ alert-opportunities.js # Alert on opportunities
```

DOCUMENTATION FOLDER

```
docs/
├─ technical/           # Technical documentation
│   ├── architecture.md  # System architecture
│   ├── smart-contracts.md # Contract documentation
│   └─ api-reference.md   # API documentation
├─ user-guides/         # User documentation
│   ├── getting-started.md
│   ├── wallet-setup.md
│   └─ optimization-guide.md
├─ hackathon/           # Hackathon materials
│   ├── pitch-deck.pdf    # From artifacts
│   ├── demo-script.md    # From artifacts
│   └─ judge-qa.md        # From artifacts
└─ README.md            # Documentation index
```

GITHUB WORKFLOWS

```
.github/  
└─ workflows/  
    │─ test.yml          # Run tests on PR  
    │─ deploy.yml        # Deploy on merge  
    │─ security.yml      # Security scanning  
    └─ lint.yml           # Code quality checks
```

ROOT CONFIGURATION FILES

```
yieldmax/  
└─ .gitignore            # Git ignore patterns  
└─ .env.example          # Environment template  
└─ docker-compose.yml    # Development setup  
└─ docker-compose.test.yml # Test environment  
└─ package.json          # Workspace configuration  
└─ README.md             # Project overview  
└─ LICENSE               # License file  
└─ .prettierrc           # Code formatting
```

FILE CREATION COMMANDS

bash

Create all directories first

```
mkdir -p yieldmax/{frontend,contracts,backend,scripts,docs}
```

```
mkdir -p yieldmax/frontend/{src/{app/{portfolio,markets,history,api/{yields,portfolio,tra
```

```
mkdir -p yieldmax/contracts/{contracts/{core,interfaces,libraries,mocks,attacks},scripts,
```

```
mkdir -p yieldmax/backend/{src/{controllers,services,models,routes/v1,middleware,config},
```

```
mkdir -p yieldmax/scripts/{demo,deployment,monitoring}
```

```
mkdir -p yieldmax/docs/{technical,user-guides,hackathon}
```

```
mkdir -p yieldmax/.github/workflows
```

Now you know EXACTLY where every file goes!

