# Emtelaak Development Timeline and Resources

## Project Timeline

The project will follow an Agile development methodology with two-week sprints. The overall timeline spans 7 months (30 weeks), divided into the following phases:

### Phase 1: Planning and Setup (Weeks 1-4)

**April 1, 2025 - April 28, 2025**

| Week | Activities | Deliverables |
| --- | --- | --- |
| Week 1 | Project kickoff; Requirements gathering; Architecture planning | Project charter; Initial requirements document |
| Week 2 | Technical architecture design; Database schema design | Architecture document; Database schema |
| Week 3 | Development environment setup; CI/CD pipeline configuration | Development environment; CI/CD pipelines |
| Week 4 | Core infrastructure setup; Base project scaffolding | Core infrastructure; Project skeleton |

### Phase 2: Core Systems Development (Weeks 5-14)

**April 29, 2025 - July 7, 2025**

| Week | Activities | Deliverables |
| --- | --- | --- |
| Week 5-6 | User identity & authorization system | User authentication system; Role-based access control |
| Week 7-8 | User registration flow; KYC integration | User registration API; KYC verification flow |
| Week 9-10 | Property listing system; Virtual tours; Document management | Property API; Virtual tour integration; Document storage system |
| Week 11-12 | Investment processing; Property share tokenization; Wallet management | Investment API; Tokenization engine; Wallet system |
| Week 13-14 | Transaction processing; Payment integration | Transaction API; Payment gateway integration |

### Phase 3: Mobile Development & Secondary Market (Weeks 15-22)

**July 8, 2025 - September 1, 2025**

| Week | Activities | Deliverables |
| --- | --- | --- |
| Week 15-16 | Flutter app architecture; UI/UX design for mobile | Mobile app designs; Architecture document |
| Week 17-18 | Core Flutter functionality; Authentication; Property browsing | Mobile app MVP; Authentication; Property browsing features |
| Week 19-20 | Investment features; Map integration; Push notifications | Investment features on mobile; Map integration; Notification system |
| Week 21-22 | Secondary market on mobile; Offline capability; Final app integration | Complete mobile application; App store submission package |

### Phase 4: Secondary Market & Advanced Features (Weeks 15-22)

**July 8, 2025 - September 1, 2025** (Parallel track)

| Week | Activities | Deliverables |
| --- | --- | --- |
| Week 15-16 | Secondary market listing; Property share trading; Bid/ask order book | Secondary market API; Trading engine |
| Week 17-18 | Property share transfer system; Ownership records; Price history tracking | Transfer system; Ownership tracking; Price history API |
| Week 19-20 | Admin dashboard; Platform management tools | Admin dashboard; Management APIs |
| Week 21-22 | Reporting system; Analytics features | Reporting API; Analytics dashboard |

### Phase 5: Integration & Testing (Weeks 23-26)

**September 2, 2025 - September 29, 2025**

| Week | Activities | Deliverables |
| --- | --- | --- |
| Week 23 | System integration testing; Performance testing; Mobile app testing | Integration test results; Performance metrics; Mobile test results |
| Week 24 | Security audit; Penetration testing; Mobile security testing | Security audit report; Remediation plan; Mobile security approval |
| Week 25 | User acceptance testing; Bug fixing; Mobile UAT | UAT results; Updated system; Mobile app refinements |
| Week 26 | Final system testing; Documentation completion; App store submission preparation | Test certification; Complete documentation; App store submission package |

### Phase 6: Deployment & Launch (Weeks 27-30)

**September 30, 2025 - October 31, 2025**

| Week | Activities | Deliverables |
| --- | --- | --- |
| Week 27 | Staging environment deployment; Pre-production testing; Mobile TestFlight/Beta distribution | Staging environment; Test results; Beta testing feedback |
| Week 28 | Production environment setup; Data migration planning; Mobile app store approvals | Production environment; Migration plan; App store approvals |
| Week 29 | Final pre-launch testing; Operational readiness; Mobile launch preparation | Launch readiness report; Mobile launch plan |
| Week 30 | Production deployment; Mobile app release; Go-live support | Launched web platform; Published mobile apps; Support system |

## Resource Requirements

### Human Resources

| Role | Quantity | Skills Required | Allocation | Responsibilities |
| --- | --- | --- | --- | --- |
| Project Manager | 1 | Project management; Agile methodologies; Financial domain knowledge | 100% | Overall project coordination; Stakeholder management; Risk management |
| Product Owner | 1 | Financial domain expertise; Product management; User story creation | 100% | Requirements definition; Feature prioritization; Acceptance criteria |
| Technical Architect | 1 | .NET architecture; Cloud services; Security architecture; Database design | 100% | System architecture; Technical leadership; Quality assurance |
| Backend Developers | 3 | C#; ASP.NET Core; Entity Framework; SQL | 100% | Backend API development; Business logic implementation; Database integration |
| Frontend Developers | 2 | TypeScript; React/Angular; HTML/CSS; UI/UX implementation | 100% | Frontend development; UI implementation; User experience |
| Flutter Developers | 2 | Flutter; Dart; Mobile app architecture; RESTful API integration | 100% | Cross-platform mobile app development; Map integrations; Mobile UX |
| DevOps Engineer | 1 | Azure/AWS; CI/CD; Infrastructure as Code; Monitoring | 50% | Deployment pipelines; Infrastructure automation; Monitoring setup |
| QA Engineers | 2 | Test automation; Manual testing; Security testing | 100% | Test planning; Test execution; Defect management |
| Database Administrator | 1 | SQL Server; Database optimization; Data migration | 50% | Database design; Performance tuning; Data integrity |
| Security Specialist | 1 | Application security; Compliance; Penetration testing | 50% | Security review; Compliance verification; Vulnerability assessment |
| UX Designer | 1 | User experience design; Wireframing; Prototyping | 100% | User interface design; User flows; Usability testing for web and mobile |

### Technical Resources

| Resource | Description | Purpose |
| --- | --- | --- |
| Development Workstations | High-performance developer machines | Software development |
| Mac Development Machines | MacBook Pros for iOS development | iOS app development |
| Azure DevOps | Project management and CI/CD platform | Project tracking and automated deployment |
| Azure Cloud Services | Cloud infrastructure platform | Hosting and infrastructure services |
| Visual Studio Enterprise | Development IDE | .NET development environment |
| Xcode | iOS development environment | iOS app development |
| Android Studio | Android development environment | Android app development |
| Firebase | Mobile backend services | Mobile app analytics, notifications, testing |
| App Center | Mobile app distribution platform | Testing distribution, crash analytics |
| Mobile Testing Devices | Various iOS and Android devices | Mobile app testing across different form factors |
| Mobile Device Management | Device management solution | Managing test devices and profiles |

## Backend Architecture Vision

Instead of providing detailed code implementation at this stage, here’s a high-level architectural vision for the .NET backend:

### Architecture Principles

1. **Clean Architecture**
   * Domain-centric approach with clear separation of concerns
   * Independence from frameworks and external dependencies
   * Highly testable structure
2. **Microservices Approach**
   * Modular services aligned with business domains
   * Independent deployment capabilities
   * Service isolation for improved resilience
3. **API-First Design**
   * RESTful API design with OpenAPI (Swagger) documentation
   * Consistent API patterns and conventions
   * Versioned APIs for backward compatibility
4. **Security by Design**
   * Identity and access management using Identity Server 4
   * Role-based and claim-based authorization
   * Data encryption at rest and in transit
   * OWASP security principles implementation

### Key Components

1. **API Gateway**
   * Request routing and aggregation
   * Authentication and authorization
   * Rate limiting and throttling
   * Logging and monitoring
2. **Microservices**
   * User Service (registration, profiles, KYC)
   * Offering Service (creation, management, documents)
   * Investment Service (processing, management)
   * Wallet Service (balances, transactions)
   * Secondary Market Service (listings, transfers)
   * Notification Service (emails, push notifications)
   * Reporting Service (analytics, reports)
3. **Cross-Cutting Concerns**
   * Logging and monitoring
   * Exception handling
   * Caching strategy
   * Messaging infrastructure
4. **Data Stores**
   * SQL Server for relational data
   * Azure Blob Storage for documents
   * Redis for caching
   * Elasticsearch for logging and search
5. **Mobile Backend Integration**
   * Mobile-optimized API endpoints
   * Push notification service
   * Offline synchronization support
   * Mobile-specific security considerations