

Project Approach

Overall Strategy: Our approach follows an agile development methodology, ensuring iterative progress, continuous feedback, and adaptability. The project will be executed in three primary phases: Planning & UI Development, Backend & Database Development, and Testing & Deployment. Each phase will have well-defined deliverables and acceptance criteria.

Rule: Project Quality is most important and developer must provide the perfect project quality in each step to go to the next plan.

Payment Cycle: Each milestone amount is paid as you pass it.

Milestone Structure

Milestone 1: Planning & UI Development

- **Duration:** 1 Week
- **Tasks:**
 - Set up project repository
 - Design UI components
 - Implement authentication system
 - Develop dashboards
- **Acceptance Criteria:**
 - Fully functional UI components with responsive design
 - Secure authentication system working with test data
 - User and admin dashboards operational
- **Deliverables:**
 - Project plan
 - UI codebase with comments
 - Initial authentication module

Milestone 2: Backend & Database Development

- **Duration:** 1 Week
- **Tasks:**
 - Develop backend APIs
 - Implement income, expenses, and banking logic
 - Set up document management & compliance tracking
 - Automate invoice duplication detection
 - Implement role-based access control
- **Acceptance Criteria:**
 - API endpoints functional and tested
 - Secure role-based access control in place
 - Financial transactions properly recorded

- **Deliverables:**
 - Backend source code with API documentation
 - Database schema design and ER diagrams

Milestone 3: Testing & Deployment

- **Duration:** 1 Week
- **Tasks:**
 - Finalize automation workflows
 - Integrate government API & payment gateways
 - Optimize database queries for performance
 - Conduct user testing and security checks
 - Deploy platform on AWS/Digital Ocean
- **Acceptance Criteria:**
 - No critical bugs during testing
 - Payment and API integrations working seamlessly
 - Performance benchmarks met
- **Deliverables:**
 - Test reports
 - Deployment documentation
 - Final project report

Deliverable Documentation

- **Project Plan:** Scope, objectives, and timeline.
- **Flow Diagrams:** System architecture and user workflows.
- **Wireframes and Prototypes:** UI design and interactions.
- **Source Code:** Well-documented with API references.
- **User & Admin Manuals:** Operational guides.
- **Final Report:** Summary of the project, key learnings, and recommendations.

Code Ownership & Access

- **Ownership:** The client will have full ownership of the source code without any licensing restrictions.
- **Repository Access:** The code will be delivered and maintained in the client's private GitHub repository for security and intellectual property protection.

Support & Bug Fixes

- **Post-Delivery Support Duration:** 30 days of bug fixes and minor updates(in special case, can delay the updated days).
 - **After deployment support price:** Hourly \$5(1/3 of main price)
 - **Response Time:** Critical issues will be addressed within 24-48 hours, while minor issues will be resolved within a mutually agreed timeframe.
-

Hosting & Deployment Responsibility

- **Server Management:** The developer will manage hosting-related issues during the 30-day support period.
 - **Documentation:** Comprehensive deployment and maintenance documentation will be provided to facilitate future server operations and maintenance.
-

Risk Management

- **Risk Identification:** Regular risk assessments during sprints.
 - **Mitigation Strategies:**
 - Backup and recovery strategies for data loss
 - Continuous security audits for vulnerabilities
 - Scalable architecture to prevent performance bottlenecks
 - **Contingency Plans:**
 - Alternative third-party service providers for dependencies
 - Reserve sprint time for critical bug fixes
-

Post-Delivery Support

- **Warranty Period:** 30 days of bug fixes and minor updates
- **Ongoing Maintenance:** Optional monthly support plans
- **Technical Assistance:** Email and ticket-based support system
- **Future Enhancements:** It define as discussing with admin about adding item price at that time(about 1/2 of main price)
- **Future Enhancements:** Custom feature additions based on user feedback
- **Sharing Github:**<https://github.com/Andrew-Costure>