Project Schedule (Timeline & Milestones):

Phase	Key Tasks	Duration	Dependencies & Notes
Phase 1: Planning, Requirements & UI/UX Design	- Requirements Engineering: • Finalize functional and non-functional requirements, SRS, user stories (min. 20), and use case diagrams. - UI/UX Design: • Create wireframes, prototypes, and high-fidelity mockups.	3 weeks	These tasks run in parallel.
Phase 2: Prototype Development & Integration	- Develop a minimum viable product (MVP) featuring: • Python-based OCR module. • Basic translation API integration with Rust. • Preliminary UI integration.	3 weeks	Depends on initial documentation from Phase 1. Early integration feedback can help refine both requirements and design.
Phase 3: Testing & Quality Assurance	- Manual Testing: • Develop and execute at least 30 detailed test cases. • Document defects during smoke and full testing. - Automated Testing: • Build a suite of at least 25 automated	2 weeks	Testing may start incrementally as modules become available; complete integration is targeted by the end of this phase.

Phase	Key Tasks	Duration	Dependencies & Notes
	tests for units, integration, UI, and API testing. • Integrate the tests into the CI/CD pipeline.		
Phase 4: Continuous Integration/Deployment (CI/CD) Pipeline	- Set up a build pipeline that: • Automatically compiles code and runs tests on pushes. • Implements code linters, unit tests, and collects code coverage. • Integrates deployment scripts with strategies (e.g., blue-green or canary releases).	2 weeks	Follows successful testing. Tools such as GitHub Actions or Azure DevOps will be used for CI/CD; configuration files (YAML/JSON) will define pipeline steps.
Phase 5: Final Integration, Documentation & Release	- Integrate feedback from testing and deployment. - Finalize user and technical documentation. - Polish UI/UX and prepare the final release.	2 weeks	Depends on a stable CI/CD pipeline. Final review and iteration to ensure the application meets quality and performance targets across both Windows and Linux.

Total Estimated Duration: 12 Weeks.

Resource Allocation

- **Human Resources:**
  - **Solo Developer: Managing all roles as defined (Project Manager, Business Analyst, Software Developer, UX/UI Designer, QA Specialist).**
- **Technological Resources:**
  - **Development & Integration:**
    - **Languages & Frameworks: Python (for OCR and computer vision tasks), C++/Rust/Go (for translation API integration and UI development).**
    - **Tools: IDE, Git for version control.**
  - **Design & Prototyping:**
    - **Tools like Figma, InVision, or Balsamiq for wireframes and prototypes.**
  - **Testing & CI/CD:**
    - **Automated Testing: Utilize frameworks and libraries for unit, integration, and UI testing.**
    - **CI/CD Tools: GitHub Actions or Azure DevOps Pipelines, code coverage tools, and build report generators.**
  - **Deployment:**
    - **Containerization (Docker) for consistent deployment environments on both Windows and Linux.**
- **Financial Resources & Budget Considerations:**
  - **API Costs: Translation API usage fees.**
  - **Software & Tools: Open-source/free.**
  - **Hardware & Testing: Use existing equipment; potential allocation for additional testing devices if necessary.**