Client Precondition Report Template

Each project team will consist of 8–10 senior Computer Science students. Each student contributes approximately 120 hours over the semester, resulting in about 960–1,200 total team hours. Students have around 13 weeks of active development time. Students possess strong general programming skills and Git/GitHub experience but should not be expected to have prior industry-level experience with specialized frameworks or technologies. Advanced tools should be listed as skills to be learned, not required prerequisites.

1. Company Information

- **Company Name:** Farmer Community
- **Primary Contact Person:** Laura MacNeil
- **Email Address:** laura@lauramacneil.com
- **Website:** https://www.farmer.community
- **Location / Time Zone:**

2. Team Structure & Required Skills

Describe your expectations for how the student team will operate.

Each student will work on a full-stack product development project using agile methodologies. There will be tasks for backend development, frontend and database design while effectively performing GitHub pull requests. That person will be required to understand the full system even if they focus on other area. The student is expected to have general development experience and the willingness to learn new technologies and frameworks.

a. Team Composition

Please outline the structure of the team as you envision it (roles, rotations, responsibilities, collaboration expectations, etc.).

Role	**Responsibility**
Product Lead	Coordinates scope, backlog, and stakeholder communication
Backend Developer	Laravel API, business logic, authentication, permissions
Frontend Developer	Filament admin UI, user dashboards, UX workflows
Database Engineer	Data modeling, migrations, query optimization
QA & DevOps	Testing, CI, deployment, bug tracking

b. Required Baseline Skills

List the *foundational* skills students must already have to participate successfully.
CS 472 students are **senior CS undergraduates** with general programming experience. They are **not industry experts**, but they are capable of learning new tools and technologies used in your project.

Baseline skills may include:

- General programming experience
- Basic understanding of software development practices
- Familiarity with Git/GitHub
- Willingness and ability to learn new frameworks or tools

Please avoid listing advanced or specialized skills as mandatory prerequisites.

Students are expected to have:

- General programming experience (any language)
- Understanding of basic software development concepts
- Familiarity with Git and GitHub
- Willingness to learn new technologies and frameworks
- Basic knowledge of databases and web applications

c. Skills to Be Developed During the Project

List skills the students will gain while working with you (frameworks, tools, domain knowledge, engineering practices, etc.).

Students will gain hands-on experience with:

- Laravel (PHP web framework)
- Filament (admin dashboard & UI framework)
- REST APIs and MVC architecture
- Database design (MySQL)
- Cloud storage (AWS S3 or equivalent)
- Authentication and role-based access control
- UI/UX for data-driven platforms
- Agile project management and sprint planning
- Automated testing and CI workflows

3. Project Title & Acronym

- **Project Title:** Farmer Community website
- **Acronym (for GitHub repository):** CARIBFARM

4. Project Overview & Problem Statement

Provide a clear description including:

- The problem your project addresses
- Background/context
- Target users
- Goals and expected impact

Please describe the project in clear, student-friendly language, emphasizing what students will learn rather than assuming prior expertise with advanced technologies. Avoid overly technical or domain-heavy jargon so that students can easily understand the scope and feel confident selecting the project.

Small and medium-scale farmers often lack access to simple and affordable digital tools for managing documents, sharing information, and working with buyers. Many existing systems are either too complex or not designed for farming communities. Caribfarm aims to solve this problem by providing a user-friendly web platform where farmers can store important files, manage their profiles, communicate with their community, and share information with cooperatives and buyers. The goal is to improve digital inclusion, organization, and efficiency while giving students experience building a real-world web application.

5. Solution Overview & Core System Components

Describe the main system components, modules, or functionalities required.

Please identify the following:

- **Core MVP Deliverables:** The minimum set of features that must be achievable within the 13-week development window.
- **Stretch Goals (Optional):** Additional features that students may pursue if time allows.

Core MVP Deliverables

Within 13 weeks, the team must deliver:

- *Secure login and user roles*
- *Farmer profile management*
- *File upload and storage system*
- *Admin panel*
- *Database for users, files, and categories*
- *Basic UI for farmers to view data*
- *Deployed working web app*

Stretch Goals

- *SMS or email notifications*
- *File preview and tagging*
- *Mobile-friendly UX improvements*

6. Technical Considerations

Specify:

- Required or preferred tech stack
- APIs, SDKs, datasets, or tools
- Constraints or limitations
- Integration requirements

- Quality/testing/CI considerations
- Security/privacy needs

Tech Stack

- Backend: Laravel (PHP)
- Frontend: Blade + Filament
- Database: MySQL
- Cloud Storage: AWS S3 or equivalent
- Hosting: DigitalOcean / AWS / Render

Other Considerations

- GitHub for version control
- Role-based access control
- Basic security (password hashing, permissions)
- CI/CD for automated testing and deployment
- Privacy: farmer data must be protected

7. Innovation & Competitive Advantage

Describe what makes your project innovative or distinct.

Farmer Community is unique because it combines farmer identity, document management, and community interaction into a single platform designed specifically for small agricultural producers rather than large agribusinesses.

8. Proposed Implementation Timeline (13 Weeks)

Provide your **own** realistic 13-week timeline and expected deliverables.

The project will begin with requirements gathering and system design, followed by setting up the Laravel environment and database. Core features such as authentication, profiles, and file management will be built in the middle of the term, with testing, refinement, and deployment occurring in the final weeks.

9. Deployment & Support Expectations

Specify any requirements related to:

- Hosting or infrastructure
- Access credentials or resources you will provide
- Documentation or training expectations

The client will provide:

- *Cloud hosting credentials*
- *AWS or storage access*

- *API keys if needed*
- *Sample data*

Students are expected to provide:

- *Setup documentation*
- *Deployment instructions*
- *System overview*

10. Confidentiality & Intellectual Property

Provide your IP, NDA, portfolio usage, and disclosure policies.

Please describe any confidentiality or IP expectations in clear, student-friendly language so that students understand what they can and cannot share in portfolios and presentations. Avoid overly legalistic or restrictive wording unless absolutely necessary.

Students may work with non-public information related to the Carib.Farm platform. Any such information should be kept confidential and not shared publicly.

Students may describe their work at a high level in portfolios, resumes, and presentations (e.g., features built, technologies used), but should not publish proprietary code, internal documentation, credentials, or sensitive system details.

Any existing code or materials provided by Carib.Farm remain the property of Carib.Farm. An NDA may be required and will be limited to protecting confidential information, without restricting academic presentations or portfolio use.

11. Proposed Success Metrics

List the metrics (quantitative and qualitative) you will use to evaluate project success.

- *Working deployed system*
- *All MVP features implemented*
- *Stable user login and file uploads*
- *Code quality and documentation*
- *Positive feedback from client*
- *Team collaboration and GitHub activity*

12. Mentorship & Communication Plan

Specify:

- Expected meeting cadence (Professor prefers weekly meetings)

- Communication platforms
- Response time expectations

- Weekly meetings (Zoom)
- Mail for communication
- GitHub for issue tracking
- 24–48 hour response time from client

13. Additional Notes (Optional)