Advanced Server-Side Scripting with PHP Lab Project Plan

Student Name: Holly Hebert

Team Name: Team One

Start Date: January 27, 2026

End Date: February 26, 2026

GitHub: https://github.com/verchew/SDC342L-Project-Group-One

Overview
This document will be used throughout the course and will serve 3 purposes:

- Define the work breakdown (tasks) you need to complete each week
- Track and report on the work you accomplish each week
- Assist in replanning your work to complete in subsequent weeks
- Provide a weekly opportunity for you to reflect on your own performance during the week and provide feedback to your instructor on how your team and teammates performed during the week.

Instructions/Submissions
You will submit an updated version of this document each week. The document will serve as a weekly status update to accompany your software code files. Instructions on what information to include in each section of the document are included with each section.

Submissions will be as follows:

- Week 1: Initial submission – Due Sunday of Week 1
    - Complete the "Initial Plan" section of the Work Breakdown table for ***each*** week of development.
    - Leave all other sections blank
- Week 2: Due Sunday of Week 2
    - Complete the following sections in the Week 2 Development section:
        - I – Additional Work section of Work Breakdown table
        - II – Work Completed
        - III – Work Remaining
        - IV – Barriers/Roadblocks Encountered
        - V – Requested Assistance
    - Add information to the following sections in the Week 3 Development section:
        - I – Additional Work section of Work Breakdown table
- Week 3: Due Sunday of Week 3
    - Complete the following sections in the Week 3 Development section:
        - I – Additional Work section of Work Breakdown table
        - II – Work Completed
        - III – Work Remaining
        - IV – Barriers/Roadblocks Encountered

        - V – Requested Assistance
    - Add information to the following sections in the Week 4 Development section:
        - I – Additional Work section of Work Breakdown table
- Week 4: Due Sunday of Week 4
    - Complete the following sections in the Week 4 Development section:
        - I – Additional Work section of Work Breakdown table
        - II – Work Completed
        - III – Work Remaining
        - IV – Barriers/Roadblocks Encountered
        - V – Requested Assistance
    - Add information to the following sections in the Week 5 Development section:
        - I – Additional Work section of Work Breakdown table
- Week 5: Due Sunday of Week 5
    - Complete the following sections in the Week 5 Development section:
        - I – Additional Work section of Work Breakdown table
        - II – Work Completed
        - III – Work Remaining
        - IV – Barriers/Roadblocks Encountered
        - V – Lessons Learned

Week 1 Planning: Project Inception & Planning

This week, the focus will be on planning and design. Your task will be to use the instructions above to complete the planning for your project and make some decisions about your project design, which go together for this project. You'll need to make design decisions about your database structure – what tables do you need, for example – as well as your User Interface layout – what pages do you need, how will you navigate from page to page, etc. All these design-level decisions will feed into your work breakdown for your project plans. Make sure to refer to the Course Project Overview for the project requirements to ensure your design and your plan meet all the requirements documented there.

Assessing	**Performance Assessment (What went well, what could improve, any actions you plan on taking)**
Self	I shared my database from last term to help with design and code structure.
Team as a Whole	The team worked well together and was able to talk through any ideas/issues without issue.
Mary	Helped document the project plan with Corey's input.
Corey	Assisted in writing the project plan and giving relevant feedback based on prior experience.
Ryan	Provided positive feedback, which boosted morale for the team.

Week 2 Development: Creating the Database and Application Framework

This week, focus on beginning your application development by creating the database your application will use and the application infrastructure. Tasks should define the table(s) to be created, the contents of those tables, and the pages/files you will create as part of the infrastructure. You may want to consider laying out the basics for the utilities you will need, such as generic validation functions. Make sure to refer to the Course Project Overview for the project requirements.

I. Work Breakdown

Add tasks to the table below to break the week's work into smaller, more manageable items. The Task Name should be a short identifier for the task. The Task Description should include enough detail to understand what needs to be done. Replace the examples in the table with your tasks. Planned Completion is the planned completion date for the task.

Task Name	Task Description	Planned Completion
Initial Plan		
Define Tables	Create the contents of the tables	February 5, 2026
Define Pages/Files	Determine the architecture of the site	February 5, 2026
Utilities	Validation functions and other mechanics	February 5, 2026
Additional Work - work not planned, but that needed to be done		
Reformat	Reformat tables via PhpMyAdmin	February 5, 2026
GitHub Updates	Create tags and commit changes to GitHub	February 5, 2026

II. Work Completed

From the table above, list the Task Name and the Actual Completion Date for the task.

Task Name	Actual Completion
Define Tables	February 5, 2026
Define Pages/Files	February 5, 2026
Utilities	February 5, 2026
Role-Based Data Separation	February 5, 2026
Reformat	February 5, 2026
Centralized Support System	February 5, 2026
Allocated Space for Hashed Passwords	February 5, 2026

III. Work Remaining (copy this to Week 3 Additional Work)

From the Initial Plan table above, list the Task Name for any tasks that were planned but not completed.

Task Name
None

IV. Barriers/Roadblocks Encountered

None.

V. Requested Assistance

No assistance was requested.

VI. Week in Review - Assessment

Assessing	Performance Assessment (What went well, what could improve, any actions you plan on taking)
Self	Integrated constraint checks within each preexisting and new table of SQL.
Team as a Whole	Created and reformatted five table structures with effective communication.
Corey	Created functional databases into PhpMyAdmin and updated SQL file with three news table structures.
Ryan	Communicated and provided feedback regarding table formatting, index integration, and design.
Mary	Updated GitHub, creating new phase and adding proper notes upon commit release.

Week 3 Development: Database Support – Representing the Database in PHP Objects

This week, the focus will be on adding database support to your web application by representing the database as PHP objects. This should be done using the MVC (Model, View, Controller) architecture, focusing on the Model and at least part of the Controller classes. Support should include database classes for connection processing and all CRUD (Create, Read, Update, Delete) operations you believe will be needed to support your application. You should also consider creating Controller classes that at least represent the tables in the database, and possibly start on the Controller classes for delivering information to the UI. Make sure to refer to the Course Project Overview for the project requirements.

I. Work Breakdown

Add tasks to the table below to break the week's work into smaller, more manageable items. The Task Name should be a short identifier for the task. The Task Description should include enough detail to understand what needs to be done. Replace the examples in the table with your tasks. Planned Completion is the planned completion date for the task.

Task Name	Task Description	Planned Completion
Initial Plan		
MVC	Apply the framework to the existing architecture.	February 12, 2026
CRUD	Implement CRUD support for database (db) classes.	February 12, 2026
Database	Create a reusable Database class using PDO to handle connection processing.	February 12, 2026
Additional Work - work not completed as part of the previous week & unplanned work that had to be completed		
File Refactoring	Renamed and reorganized files to maintain consistent naming conventions (singular model, plural table).	February 12, 2026
Error Resolution	Resolved file path issues and corrected table naming mismatches between model and database.	February 12, 2026
Password Handling	Implemented password hashing during customer creation for improved security practice.	February 12, 2026

II. Work Completed

From the table above, list the Task Name and the Actual Completion Date for the task.

Task Name	Actual Completion

Database Class	February 12, 2026
Customer Model	February 12, 2026
Customer Controller	February 12, 2026
MVC Routing	February 12, 2026
Basic UI Integration	February 12, 2026

III. Work Remaining (copy this to Week 4 Additional Work)

From the Initial Plan table above, list the Task Name for any tasks that were planned but not completed.

Task Name
Users Table
User Registration

IV. Barriers/Roadblocks Encountered

The primary challenges encountered involved file path errors, naming inconsistencies between singular and plural file/table names, and adjusting the routing structure to move from a standalone entry file to a centralized index.php front controller. Additionally, resolving database table name mismatches caused temporary SQL errors that required debugging.

V. Requested Assistance

No additional assistance is needed at this time. All identified issues were resolved through troubleshooting and testing.

VI. Week in Review - Assessment

Assessing	Performance Assessment (What went well, what could improve, any actions you plan on taking)
Self	Successfully implemented MVC structure, created database abstraction through PDO, and developed fully functional CRUD operations for the customers table. I also refactored the project structure to prepare additional tables in future weeks.
Team as a Whole	The team continues to collaborate effectively. Responsibilities are clearly divided, and communication remains open regarding structure and implementation of decisions. This week's progress establishes a foundation that will make integration of additional tables smoother.
Corey	Successfully implemented MVC structure, created database abstraction through PDO, and developed fully functional CRUD operations for the

	complaints table. Also updated the project's GitHub with current content and releases.
Team Member	Successfully implemented MVC structure, created database abstraction through PDO, and developed fully functional CRUD operations for the employees table.

Week 4 Development: Site Security

This week, the focus will be on site security, specifically adding both Authentication and Authorization. HTTPS is required and only authenticated; authorized users should be able to access specific pages. Make sure to refer to the Course Project Overview for the project requirements.

I. Work Breakdown

Add tasks to the table below that break up the week's work into smaller, more manageable work items. The Task Name should be a short identifier for the task. The Task Description should include enough detail to understand what needs to be done. Replace the examples in the table with your tasks. Planned Completion is the planned completion date for the task.

Task Name	**Task Description**	**Planned Completion**
Initial Plan		
Authentication	Ensure users can log in to the correct account (HTTPS).	February 19, 2026
Authorization	Ensure users can access appropriate databases/pages.	February 19, 2026
Session Management	Implement PHP session handling to maintain an authenticated state.	February 19, 2026
Secure Password Handling	Implement password hashing and verification using secure PHP functions.	February 19, 2026
HTTPS Enforcement	Ensure site is accessed securely and verify Apache SSL configuration.	February 19, 2026
Additional Work - work not completed as part of the previous week & unplanned work that had to be completed		
Database Schema Correction	Resolved inconsistencies between CREATE TABLE and INSERT statements in updated SQL file.	February 22, 2026
Table Name Alignment	Standardized table and column naming to match application queries.	February 22, 2026
Access Error Resolution	Debugged and resolved database connection and missing table errors (e.g., users table not found).	February 22, 2026
Rebuild Database	Dropped and re-imported database to ensure clean schema alignment.	February 22, 2026

II. Work Completed

From the table above, list the Task Name and the Actual Completion Date for the task.

Task Name	**Actual Completion**
Authentication	February 19, 2026

Authorization	February 19, 2026
Session Management	February 19, 2026
Secure Password Handling	February 19, 2026
HTTPS Enforcement	February 19, 2026
Database Schema Correction	February 22, 2026

III. Work Remaining (copy this to Week 5 Additional Work)

From the Initial Plan table above, list the Task Name for any tasks that were planned but not completed.

Task Name
None. All planned Week 4 tasks were completed.

IV. Barriers/Roadblocks Encountered

During implementation, inconsistencies between SQL table definitions and insert statements caused database import failures. Additionally, mismatches in table and column naming resulted in runtime errors when attempting to authenticate users. Coordination challenges related to maintaining a single shared database structure required careful schema alignment and re-importing to ensure application stability.

V. Requested Assistance

No additional assistance is required at this time. All identified security and database integration issues were resolved through troubleshooting and validation testing.

VI. Week in Review - Assessment

Assessing	Performance Assessment (What went well, what could improve, any actions you plan on taking)
Self	Created user account registration page. While database inconsistencies initially slowed progress, systematic debugging and schema alignment allowed successful implementation of secure login, role-based access control, and HTTPS verification
Team as a Whole	The team successfully implemented the required security features and resolved structural database issues. While naming inconsistencies created temporary instability, collaborative debugging ensured the system is now aligned and functioning as expected.
Mary	Authorization and authentication functions and help to rewrite tables for consistency.
Ryan	Helped with testing, especially with faulty https features.

Corey	Delegated tasks around illnesses and personal conflict, wrote coding for page linking, rewrote missing CRUD, and helped debug errors.

Week 5 Development: File Support and Finalizing the Application

This week, the focus will be on adding support for image file uploads and text file manipulation, finalizing your application, and testing it. Make sure to refer to the Course Project Overview for the project requirements.

I. Work Breakdown

Add tasks to the table below to break the week's work into smaller, more manageable items. The Task Name should be a short identifier for the task. The Task Description should include enough detail to understand what needs to be done. Replace the examples in the table with your tasks. Planned Completion is the planned completion date for the task.

Task Name	Task Description	Planned Completion
Initial Plan		
Image files/text	Add support for image files and text file manipulation.	February 26, 2026
Finalization	Review and finalize the changes implemented during the previous weeks.	February 26, 2026
Testing	Debug and eliminate possible errors.	February 26, 2026
Final Integration	Integrate all team components (customers, employees, complaints, products, cart) into one functioning application.	February 26, 2026
UI Refinement	Apply consistent styling across all pages and ensure visual consistency.	February 26, 2026
Bug Resolution	Identify and resolve runtime errors, database mismatches, and path issues.	February 26, 2026
Additional Work - work not completed as part of the previous week & unplanned work that had to be completed		
Full System Testing	Test authentication, authorization, CRUD operations, and database integrity across all roles.	February 26, 2026
Final Validation Review	Verify application meets all Course Project Overview requirements.	February 26, 2026
Schema Reconciliation	Resolved inconsistencies between team SQL updates and existing schema.	February 26, 2026
Path & Routing Fixes	Corrected relative and absolute paths for consistent page rendering.	February 26, 2026
Style Integration	Unified stylesheet across all views to ensure consistent UI.	February 26, 2026
Database Rebuild	Dropped and re-imported database to ensure a clean final state.	February 26, 2026

II. Work Completed

From the table above, list the Task Name and the Actual Completion Date for the task.

Task Name	Actual Completion
Final Integration	February 26, 2026
Full System Testing	February 26, 2026
UI Refinement	February 26, 2026
Bug Resolution	February 26, 2026
Final Validation Review	February 26, 2026

III. Work Not Completed

From the Initial Plan table above, list the Task Name for any tasks that were planned but not completed.

Task Name
None. All planned Week 5 tasks were completed prior to final submission.

IV. Barriers/Roadblocks Encountered

During final testing, several minor bugs and integration inconsistencies were identified. These primarily involved database schema mismatches and styling inconsistencies across pages. While not major blockers, resolving them required careful debugging and coordination to ensure the final version of the project functioned correctly across all features.

V. Week in Review - Assessment

Assessing	Performance Assessment (What went well, what could improve, any actions you plan on taking)
Self	I contributed to testing efforts and provided updates to the overall style sheet to improve the consistency and visual design of the application, ensuring that the final UI presentation was consistent across all views.
Team as a Whole	The team worked collaboratively to finalize the project. Communication during the testing phase helped identify remaining issues quickly, and everyone contributed to polishing their assigned components to meet the final requirements.

Mary	Mary assisted with reporting bugs discovered during testing and updated the README file to reflect final project details.
Ryan	Ryan participated in bug reporting and created the final demonstration video outlining the application's features and functionality.
Corey	Corey performed comprehensive testing of the application, reviewed reported issues from team members, and resolved bugs related to database interactions, authentication, and page styling.

VI. Lessons Learned

Throughout the planning and execution of this project, I learned the importance of maintaining consistent naming conventions, clear database version control, and structured collaboration within a team environment. As the application grew in complexity, even small inconsistencies in database schema, file paths, or naming caused larger integration issues. This reinforced how critical organization and communication are in multi-developer projects.

I also strengthened my debugging process. Rather than reacting emotionally to errors, I learned to isolate variables, verify database structure, confirm file paths, and test systematically. That methodical approach made resolving issues much more efficient.

If starting over, I would implement clearer version control practices from the beginning, including maintaining separate database versions and structured naming standards across all team components. Establishing these standards early would reduce integration friction and prevent unnecessary rework later in the project.

Overall, this project improved my understanding of MVC architecture, database interaction, authentication security, and collaborative development. It highlighted both the technical and organizational skills required to deliver a functioning, polished application.