

Assessment Timing and Instructional Orchestration Model

Author: Dr. Abraham N. Aldaco-Gastelum

Affiliation: Department of Computer Science / Software Engineering

University of Wisconsin Platteville

Contact: aldacoa@uwplatt.edu

Purpose of This Document

This document presents a reproducible assessment timing and orchestration model for an undergraduate full-stack web development course. It abstracts detailed scheduling artifacts into a semester-agnostic framework suitable for adoption across institutions.

Design Principles

The assessment model follows principles of concept readiness, incremental integration, early cognitive engagement through proposals, intentional overlap between proposal and implementation, workload balance, and asynchronous reinforcement.

Assessment Components Overview

The course includes short-term assignments, midterm and final project proposals, implementation phases, and asynchronous learning activities.

Relative Timing of Assessments

The table below summarizes typical placement of assessments by instructional week.

Proposal–Implementation Overlap

Proposal and implementation phases intentionally overlap to promote iterative refinement and mirror professional development practices.

Integration of Asynchronous Learning Activities

Asynchronous learning activities are released prior to assessments to support preparation and varied learning paces.

Final Project Demonstration and Deliverables

Final deliverables include software, documentation, a demo video, and a live demonstration.

Adaptation Guidelines

Instructors are encouraged to adapt pacing, number of assignments, technologies, and mentoring support while preserving core principles.

Reproducibility Statement

This assessment orchestration model is shared to support replication and adaptation of the course design.

Assessment Timing Summary

Assessment Component	Typical Timing (Weeks)	Pedagogical Purpose
Assignment 1	2–3	Apply foundational frontend concepts
Assignment 2	7–8	Reinforce advanced JavaScript and data handling
Assignment 3	9–10	Apply frontend framework concepts
Midterm Project Proposal	5	Initiate early project ideation
Midterm Project Implementation	6–8	Integrate early full-stack concepts
Final Project Proposal	11	Refine design and architecture
Final Project Implementation	12–15	Full-stack integration and demonstration