# Daily Report – Reveal.js Enhancements

This report summarizes the work completed today on the Web Technologies assignment, focusing on enhancing the Reveal.js presentation framework with custom interactive features.

## 1. Contribution Workflow

We reviewed the Git workflow outlined in the WT updates document. Contributors must work on the 'New-Contribution' branch, pulling updates before making changes, and pushing commits after staging and committing changes (git add, git commit, git push).

## 2. Project Overview

Reveal.js is an open-source, web-native presentation framework built with HTML, CSS, and JavaScript. Unlike PowerPoint, it is highly customizable and integrates well with web technologies. Today's focus was on extending its interactivity and adding new features beyond the built-in demo.

## 3. Enhancements Implemented

The following features were added or improved in index.html:

• Progress bar and slide numbers for better audience orientation.

• Draggable whiteboard overlay for live annotations during presentations.

• Draggable slide minimap to quickly navigate between slides.

• Media effects panel (Glitch, VHS, BlurPulse) to apply cinematic styles to slide media.

• Per-slide overlays (dark, grid, dots) with global toggle via keyboard shortcut.

• Live code playground allowing editing and execution of HTML/JS directly inside a slide.

• Terminal/CLI simulation with auto-typing scripted outputs.

• Dynamic media filter sliders (blur, grayscale, hue, brightness, contrast).

• Drag-and-drop interactive blocks for flowcharts/mind maps.

• Local live poll system with bar chart visualization.

• Added comments in the code for clarity and maintainability.

## 4. Advantages of Enhancements

These improvements provide several advantages:

• Stronger audience engagement via interactivity (polls, playground, whiteboard).

• More professional and cinematic presentations with overlays and effects.

• Easier navigation and orientation with minimap and slide numbers.

• Live teaching and demonstration capabilities (terminal sim, playground, whiteboard).

• Extensible design: new effects, overlays, and interactive blocks can be added easily.

## 5. Next Steps

Future tasks may include:  
• Enabling remote audience polls via WebSocket or Supabase backend.  
• Adding AI-assisted theme styling for faster design.  
• Export options to PDF, video, or GIFs for portability.  
• Testing and optimizing performance on different devices.