05-T2 Website Transformation

Final Report Document

4850- all sections

Spring 2025

April 28, 2025

Professor Perry

Team 2

Website: https://05t2-gso.github.io/pages/

GitHub: https://github.com/05T2-GSO/

Lines of Code: 854

Components/Tools Used: WordPress, Elementor, Docker, GitHub

Total Man Hours: 306

Estimated Hours: 368

NDA: No

Contents

1. Introduction

1.1 Project Background

The Georgia Symphony Orchestra (GSO) is a regional cultural institution that includes a range of music programs such as the Georgia Youth Symphony Orchestra (GYSO), Jazz, and Chorus. Each of these programs previously had their own semi-independent web presence, resulting in fragmented branding, inconsistent user experiences, and outdated designs. The organization sought to unify these digital identities into a single, modern, and user-friendly website.

1.2 Objective

The primary goal of our project was to create a **proof-of-concept (POC)** website that consolidates the GSO, GYSO, Jazz, and Chorus programs into a single cohesive web platform. Initially, the sponsor envisioned a complete rebuild of the site from scratch. However, the scope shifted mid-semester to instead focus on improving and integrating portions of the existing site using the WordPress CMS. Our team was tasked with prototyping enhancements to the user experience, navigation, and event management components of the GSO website. This also included preparing a guide for future updates that could be performed by non-technical staff.

1.3 Scope

Originally, the scope involved a full site rebuild but was adjusted after the first milestone to prioritize improving the existing site structure. Our final deliverables focused on:

- Restructuring the sitemap for better user flow
- Redesigning the Events and Calendar system
- Ensuring mobile responsiveness
- Incorporating accessibility features
- Embedding ticketing tools and donation functionality
- Providing training documentation for GSO's staff

Excluded from our scope were:

- Full rebranding of all pages
- Custom backend development beyond plugin configuration
- Overhauling ticketing/payment systems (limited to integration only)

1.4 Stakeholders and End Users

The primary stakeholder was the Georgia Symphony Orchestra sponsor team. End users include a wide audience:

- Older event attendees
- Parents of student musicians
- Donors and Patrons
- Students exploring youth programs
- Internal GSO staff maintaining content

This diverse user base required thoughtful design to accommodate different accessibility, device usage, and information needs.

1.5 Technologies Used

To deliver the solution, we used:

- **WordPress CMS** with the **Elementor** page builder for front-end customization
- Docker containers for local and staging deployment
- A staging environment hosted at: https://georgiasymphonyatl.staging.tempurl.host/
- Plugins for ticketing (Ovatheme), events calendar, SEO, accessibility, backups, and analytics

This stack allowed us to build a highly modular, responsive, and accessible proof of concept while aligning with the sponsor's familiarity with WordPress.

2. Requirements

2.1 Functional Requirements

The Georgia Symphony Orchestra website must provide the following user-facing functionality:

Homepage

- Integrate branding for GSO, GYSO, Jazz, and Chorus programs
- Include a mission statement and value proposition ("elevator pitch")
- Display upcoming events prominently with "Buy Tickets" buttons
- Contain persistent donation and navigation elements

Event Management

- Events must be listed in both **calendar view** and **list view**
- Users must be able to filter events by:
    - Program (GYSO, Jazz, Chorus)
    - Location
    - Date
- Each event page must include:
    - Title, date, time, venue
    - Performer bios and multimedia content
    - "Buy Tickets" button
    - Social sharing buttons
    - Parking and accessibility info

Donation System

- Display donation call-to-action on every page
- Support external links or embedded tools for various donation types
- Preserve GSO's current donation structure (Youth vs Adult program)

Program Pages

- Individual pages for GSO, GYSO, Jazz, and Chorus
- Each page includes:
    - Program overview
    - Media galleries (photos, videos)
    - Join/volunteer information
    - Testimonials or past highlights

Blog & News

- Unified blog for news updates, past performance recaps, and interviews
- Easily updated by non-technical staff

Administrative Tools

- Admins must be able to:
    - Add/edit events
    - Upload media and posts
    - Modify homepage banners or content areas

2.2 Non-Functional Requirements

- **Accessibility**: Must comply with WCAG standards. Support keyboard navigation, screen readers, alt text, etc.

- **Responsiveness**: Mobile-first layout. Design must function on phones, tablets, and desktops.
- **Performance**: Site should load in under 3 seconds on standard connections
- **Security**: Role-based access (RBAC) for admins vs. general users. Use HTTPS for all communications.
- **Scalability**: Architecture should support future API integration and user analytics dashboards
- **Maintainability**: Easy for non-technical GSO staff to update content via WordPress dashboard
- **SEO**: Pages should follow basic search engine optimization best practices (titles, meta, structured layout)

2.3 Business Rules & Sponsor Constraints

- Project must use **WordPress CMS**
- All features should be configurable using plug-ins or no-code tools
- No major changes to ticketing platform (OvationTix); only embed or link it Design choices must align with GSO's branding and support consistency with their existing promotional materials
- Final deliverables must be hosted on a public site, not Google Docs or Word files

2.4 Assumptions

- WordPress plugins used will be compatible with current CMS version
- Sponsor will provide branding assets (logos, bios, event data)
- GSO staff will maintain the site post-handoff, so training docs are included
- External APIs (e.g., ticketing or Google Calendar) are stable and allow embedding

3. Analysis / Design

3.1 Site Architecture & Navigation

The original GSO website suffered from inconsistent layouts and navigation across its various branches (GYSO, Jazz, Chorus). Our analysis revealed redundant pages, inconsistent link structures, and a lack of clear content hierarchy.

- Discrete examples of these weaknesses that we addressed are:
- Fixing The Favicon
- Enabling certain subdirectories, i.e. branding pages, to support alternative favicons.
- Converting Sponsors Block to a Carousel
- Leveraging built-in Elementor widgets to create a rotating display
- Improves page real estate usage and visual engagement, example on staging site

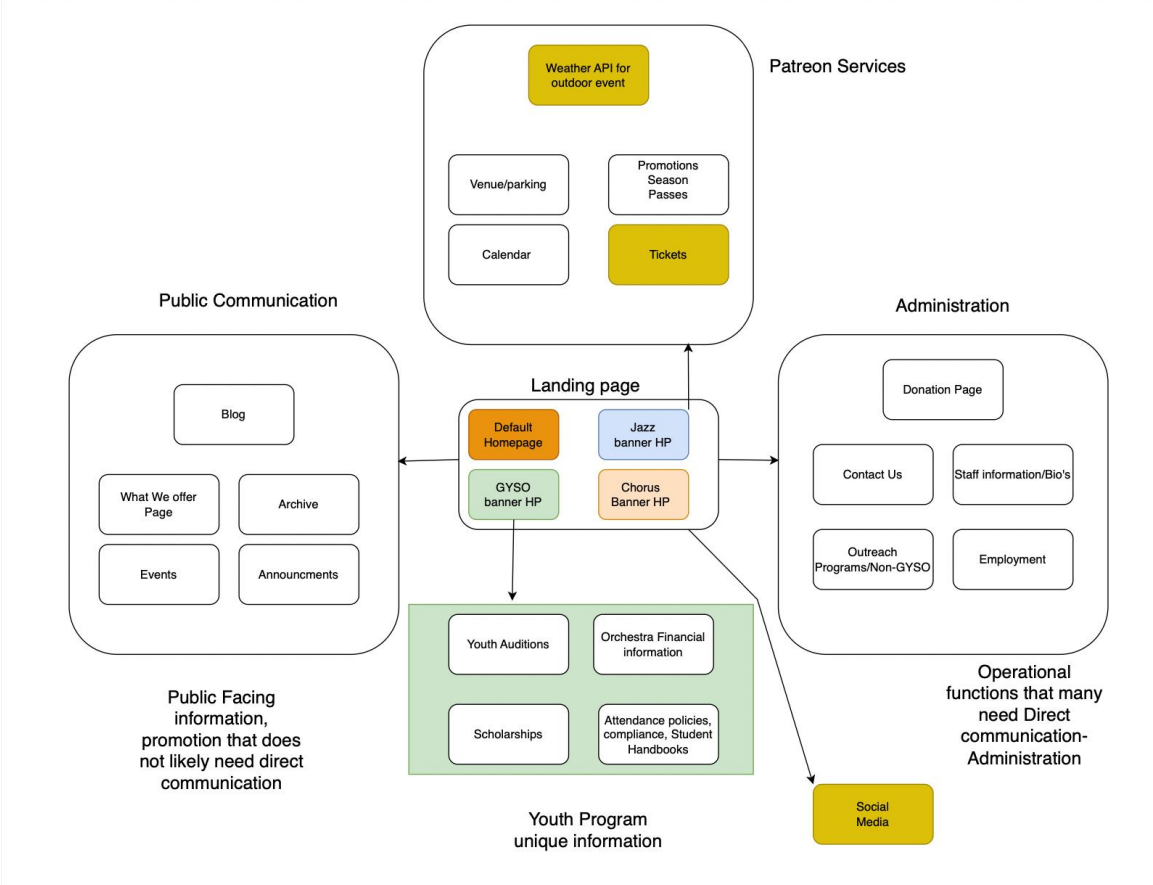
- Combining Headers

Simplifying navigation and increase focus on primary content

We reduced the number of clicks needed to access tickets, events, and contact information. A responsive navigation menu was implemented to improve user flow on both desktop and mobile devices.

Because of inconsistent requirements and request expressed by the sponsors throught the development of the project our scope and approach to design has has to be very agile and open to change.

Initially we planned to combine the site partitions in a structure like the diagram below. Part way into the project the Sponsor decided they wanted us to focus on the ticketing system and this design was set aside

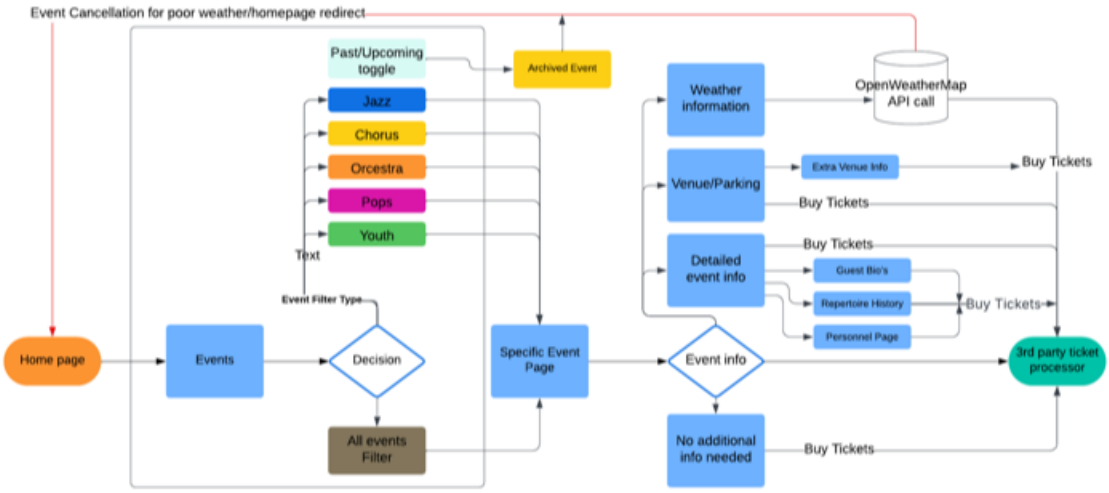


3.2 UI/UX Approach

In the early stages of development, Wireframes and user stories were created Figma to visualize ideal user flows. The team identified four key user types:

- Older Event Attendee (primary)
- Ambitious Music Student (primary)
- Parent of a Student (secondary)
- Donor (secondary)

Our design principles focused on accessibility (large text, contrast), mobile-first layout, and clean information flow. The Elementor's drag-and-drop builder was used for quick iteration and visual consistency. In later stages after sponsor rescopes, we created user flow diagrams like this to help model the site and look for ways to reduce clicks as much as possible



3.3 Technology & Plugin Strategy

Because the sponsor was adamant is maintain usage of the wordpress ecosystem, much of of planning and development had to revolve around the usage of 3rd party plug-ins and researching existing solutions.

WordPress is a content management system and platform that is highly customizable and is mostly based in the PHP language. The benefit to using the CMS rather than a framework from scratch is WordPress already has built in security, UI, dashboard and can be greatly extended through plug-ins.

Most of the core function programs are contained the the WordPress “Core”. These files should rarely ever be touched and all modification around them is typically done through plug-ins or queries.

Beyond the PHP code/plug-ins, the rest of the site code is stored in an open-source SQL fork called MariaDB. The formal SQL is never directly altered in wordpress so our documentation on that is understandably sparce.

Key plugins included:

- Elementor Pro (page building)
- Ovatheme Events (ticket integration)
- WP Mail SMTP (email handling)
- Hummingbird Pro (performance optimization)
- Loco Translate (accessibility/localization)
- Snapshot (backups)

3.4 Design Constraints

- The system must be scalable to accommodate increasing users and events.
- Compliance for data protection and payments via WordPress CDN and outsourcing
- payments.
- Must support mobile and desktop web clients.
- Limited latency in API responses to ensure a seamless user experience.
- The system should be able to handle a high volume of traffic, especially during peak ticket sales.
- Data consistency must be maintained across all components, requiring proper database transactions and rollback mechanisms.
- System backups must be regularly scheduled and stored securely in order to ensure recovery in case of data loss.
- It must be designed to support future integrations with other external systems.
- All content must be editable by non-technical staff
- Ticketing system (OvationTix) could not be replaced, only integrated
- Website should emulate the GSO brand identity

3.5 Wireframes and Mockups

Wireframes were created for key pages (Homepage, Events, Program Pages, Blog). These were user-tested informally to refine layout and labeling. Final mockups were used to guide Elementor layout implementation.

4. Development

4.1 Workflow & Tools

Weekly meetings, research notes, team collaboration, and peer to peer learning and development (Agile methodology).

The weekly meeting is intended to complement the detailed Gant chart from the project planning documents. Each person is responsible for approximately 5-7 hours of work towards development goals that align with the team members role.

In the first weeks, more emphasis was be placed on accountability and learning the new technologies that on concrete deliverables.

Versioning using GitHub/GitHub Repository.

Regular code reviews to maintain best coding practices and standards

Testing and deployment were a major emphasis in the 2nd milestone phase of the project.. Site testing will included having a list of all tags be given to the end user and the user will Deadlinks will be fixed promptly, issues that arise from GUI will also be resolved before stakeholder demonstrations.

All code and site progress were version-controlled using GitHub in the 05T2-GSO organization. We used Docker for creating a portable WordPress + MariaDB staging environment, allowing each member to test features locally.

Toolset:

- WordPress + Elementor
- MariaDB (backend database)
- Docker Compose
- GitHub (feature/dev/main branches)
- Figma (mockups and wireframes)

4.2 Front and Backend Development

Jonathan Turner & Noah Minch developed and enhanced the front and backend, were responsible for development the site calendar, researched plug-ins and creating deliverable outlines for each stage of development:

- Prototyped and tested custom WP plugins and themes.
- Created Docker Compose environment scripts for clean environments.
- Prototyped desired features (e.g., Google Docs to Native HTML.
- Contributed to Figma (mockups and wireframes) Designs.

Overview

- Technology Stack

- WordPress (PHP & JavaScript)
- MariaDB for database storage
- Plugins
- Technology stack
- Front end

Alterations and changes to the front end are what we are primarily concerned with in this project. The front end uses the standard suite of HTML, CSS and JavaScript. Most of our front-end alterations will however be done using plugins and tools that are located in the back end of the stack. Much of the page content is stored in the SQL table in the back end of the database on wp_posts tables. While HTML, CSS & JS are used to run the page, most of what we will use is rendered from assets and code generated by the site builder Elementor.

Backend

Server— The server runs on a Linux architecture. The environment we will be utilizing will be an Ubuntu Server 24.04.2 LTS utilizing Apache2 (httpd) as the reverse proxy with docker containers for the WordPress site with a separate MariaDB database for additional functionality.

Programing languages— Aside from the usual HTML, CSS, JS the WordPress server uses PHP (Hypertext Preprocessor). The core function, indexes, administrative pages, data checks, passwords, site themes and plugins are almost entirely written in PHP. If information is not stored in the database, it will most likely be PHP that is essential the running of the site or a type of asset that in referenced to but not stored in the database. Examples of this are images, pdfs and borrowed font assets from other locations.

Databases— the server runs MariaDB, which is a fork of the MySQL relational database management system. The content, metadata, display of every page is an entry in this database and the individual columns and functions of the database will be outlined in the database section of the design document.

Framework— the framework for our purposes will be considered the Content management system WordPress. Abstractly, the framework is the PHP for the core files contained on the server, but the data administrator very rarely interacts directly with the PHP and much of the functionality expected from a framework is fulfilled by the dashboards from the WordPress PHP.

In addition to the technology stack, the functionality of WordPress is extended and augmented via the usage of open source or service-based 3rd party or custom plugins. Like most of WordPress the plugins are written in PHP and can be responsible for a large array of functions such as site data organization, making API calls, collecting data, adding extra layers of encryption and safety, and anything else a developer can think of.. GSO uses 31 unique plug-ins but conceptually some of the plug-ins can be clustered together. Below are the most important plug-ins:

Plug-ins

Elementor/Elementor pro— Elementor is a drag and drop page builder for WordPress, allowing users to design custom pages and posts without coding. It extends the functionality by adding advanced widgets, theme building capabilities, WooCommerce integrations, and more.

Embed Plus for YouTube Gallery— This plugin allows you to embed and customize YouTube videos, galleries, live streams and shorts on your WordPress site. It offers features like lazy loading with facades, deferment of JavaScript.

Wp File Manager— This plugin provides an easier interface to file management directly in the WordPress dashboard. Allowing for quick uploads, edits, deletion, and organization of files without FTP/cPanel use.

Wp Mail Logging & Wp Mail SMTP— Provides and enables the sending and configuration of the SMTP emails, allows for more granular control of email lists, monitoring and troubleshooting newsletters and other various automated email systems.

The Suite of WooCommerce dependencies— This is a set of plug-ins that are paid services to support ecommerce. The names of the plugin are WooCommerce, WooCommerce Breadcrumbs, WooCommerce Legacy REST API, WooCommerce Shipping & Tax, and WooPayments. All the plugin-ins augment the functionality of the base WooCommerce, the Shipping & Tax plug-in supports international shopping and breadcrumbs is designed to collect information on customers. The legacy REST API plug-in helps WooCommerce run on the Elementor pages.

The Suite of Ovatheme Dependencies— The GSO uses A 3rd Party ticketing service called AudienceView which bought up the ticketing system Ovationtix- hence why the plugin name does not reflect the company's current name. The Ovatheme dependencies plugin-in allows calendar entries created in audience view to be displayed on WordPress and link to the ticketing site.

Hummingbird Pro— It boosts site speed through caching, image optimization, and performance scanning. It also includes tools to defer CSS/JS and improve Google PageSpeed insight scores.

Loco Translate— Provides in-browser editing of WordPress translation files. It simplifies localization by offering direct access to .PO and .MO files from the WordPress dashboard.

Snapshot Manager— It enables scheduled backups of your WordPress site, making it easier to restore in case of site crashes or data loss. It offers both manual and automated backup options.

Widget importer Exporter— It allows you to export widgets from one WordPress site and import them into another. It streamlines site migration or duplication processes.

Redirection— Allows for simple and easy redirects of common webpage errors, such as 301, 404, 403, and many others to specific pages to help SEO ranking.

PDF Generator addon for Elementor page builder— Enables the ability of generating a true- to-scale pdf of an Elementor page via the browser print page option, or other prompts.

Contact form 7— This plugin allows for easy and customizable contact forms to be created directly through the site panel without need of utilizing cPanel or other more complicated means.

The website used the traditional Windows, Icons, Mouse & Pointer (WIMP) paradigm commonly used by desktops. If we get to a mobile development phase this will be extended to touchscreen as well

4.3 Documentation & Content

Rashmi Wagde managed: User stories, documentation

Kevin Syhavong managed: Documentation and scheduling

4.4 Development Challenges

-Sponsor halfway through SDLC rescoped the project. For the sponsor this makes sense because they will derive more value from 3 groups focusing on improving different components rather than rushing to make a vaguely complete prototype of their current sites reskinned.

-Our group was not given the rescope information till slightly before break, presentation will feature SDLC documents from before and after the rescope.

-New restricted Scope is to improve the current site-user flow and improve the current events/calendar system to support attaching biographies, additional event information

- **Scope Change**: Midway through, the sponsor shifted to a consulting-style model rather than a full rebuild.
- **Limited Access**: The sponsor site was hosted offsite and required extra care for staging.
- **Plugin Overhead**: Compatibility issues arose from the sponsor's existing plugin set; we resolved this by containerizing the environment and limiting plugin sprawl.

5. Test (Plan and Report)

5.1 Test Methodology

We conducted **manual testing** across multiple environments including:

- Desktop: Chrome, Firefox, Safari, Edge
- Mobile: iOS Safari, Android Chrome
- Devices: Windows, Mac, Linux laptops, iPhones, Androids

Testing was based on a requirements matrix created from our SRS and design documents. Each team member tested components aligned with their area of development, logging results in a shared spreadsheet.

5.2 Testing Areas

- **Functional Testing**: Ensured events, navigation, ticketing, and donation elements worked as intended
- **Accessibility Testing**: Screen reader support, tab navigation

- **SEO Testing**: Checked sitemap generation, metadata, structured headings
- **Performance Testing**: Evaluated load times with and without plugins
- **404 & Redirects**: Ensured broken links are redirected to a custom 404 page

5.3 Results Summary

Test Area	Result	Severity
Homepage loads properly	Pass	High
Event filtering	Pass	High
Ticket system embedded	Pass	Medium
Screen reader compatibility	Partial	Medium
404 redirect in place	Pass	Medium
Mobile nav menu collapses correctly	Pass	High
Sponsor carousel display	Pass	Low

5.4 Recommendations

- Tune layout spacing on some mobile viewports
- Maintain plugin updates and revalidate monthly
- Provide training documentation that includes backup procedures

6. Version Control

GitHub org: 05T2-GSO. Feature/dev/main branches used.

7. Summary

The Georgia Symphony Orchestra Website Transformation project successfully achieved its core objective: to design and deliver a modern, accessible, and user-friendly website prototype that unifies GSO's various programs under a single digital presence.

Our team built a locally hosted, WordPress-based proof of concept that includes:

- A unified homepage for GSO, GYSO, Jazz, and Chorus
- Redesigned event and ticketing interfaces with calendar and filtering
- Accessibility features aligned with WCAG standards
- Responsive design for mobile and desktop
- Integrated donation and blog systems
- A full backend system supporting admin content updates and plugin management

Despite mid-semester changes in project scope, we adapted our development strategy, redefined deliverables, and ensured alignment with the sponsor's evolving expectations.

This project not only shows our technical skills—such as working with WordPress, Elementor, and APIs—but also demonstrates our adaptability, communication, and commitment to user-centered design.

We believe this project serves as a solid foundation for the Georgia Symphony Orchestra's long-term digital transformation.

8. Appendix

Included documents: Gantt Chart, Final Research Report (Per Sponsors Request), Time Sheets, Source Code.

8.1 Project Selection excerpt

Abstract:

The Georgia Symphony Orchestra (GSO) seeks to integrate its diverse offerings (GSO, GYSO, Jazz, Chorus) into a unified website to streamline branding, convey their mission statement and improve user experience. Your capstone project involves designing and developing a proof of concept (POC) for this unified website. The goal is not to deploy a production-ready solution but to demonstrate innovative ideas, technical feasibility, and advanced computing skills that align with the project's objectives.

Platform:

The POC website will be hosted locally on a private server and tentatively designed via the content management system (CMS) WordPress. The project will deploy a technology stack abstractly divided into a front-end, backend, database and API endpoints for a ticketing system.

Tentatively, the front-end system will use the standard web development technologies HTML, CSS, JavaScript, and PHP.

The backend may use node.js and/or python libraries.

The database Tentatively will be either done in mySql or SqlLite.

To create the UX/UI, we will primarily use Figma.

Project Deliverables:

Team/Project Selection document (Individual Assignment)

- Website Design and Development:
  - Home page integrating GSO, GYSO, Jazz, and Chorus into a unified structure
    - 1. Showcase Program Diversity
      - Highlight Programs in a Unified Way: Develop a "What We Offer" page showcasing all programs (youth, jazz, chorus, and orchestral performances) under a cohesive GSO umbrella.
      - Integrate Multimedia Content: Feature video highlights, artist interviews, and behind-the-scenes stories from all programs to engage and inspire audiences.
    - 2. Enhance Accessibility and Inclusivity
      - Enhance Accessibility Features: consider including screen-reader compatibility, large-text options, and sensory-friendly concert labels to ensure inclusivity for all users.
    - Strengthen Communication and Engagement
      - Leverage a Centralized News and Blog Section: Dedicate a place to publish unified stories, achievements, and updates about all GSO programs to emphasize their shared mission and increase audience connection.
      - Implement Targeted Landing Pages for Campaigns: Design specific pages for campaigns like fundraising, volunteer recruitment, and corporate sponsorships to attract diverse audiences and stakeholders.
    - 4.Optimize Digital Experience
      - Optimize for Mobile Users: Ensure the website is responsive, offering an exceptional user experience across devices, particularly mobile.
      - Integrate Advanced Analytics: consider simulating features to track user interactions and provide insights for optimizing site performance and engagement strategies.
  - Sections for each program with sub-navigation for their specific offerings.
  - Interactive elements like sliders, tabs, and multimedia galleries.
  - Mock booking system for events, demonstrating different categories of tickets.

- Functional Features - chose one or more after discussion with your capstone professor:
    - User Analytics: Implement sample dashboards showing metrics such as page visits, user demographics, and engagement levels.
    - Dynamic Content: Simulate features like upcoming events, blogs, and volunteer opportunities.

- Backend & Technology Demonstrations
    - Sample CMS implementation.
    - Database simulation for storing events, user feedback, and analytics data.
    - Mock API integration for external services like Google Maps or email newsletters.

- Documentation
    - User manual outlining website navigation and key features.
    - Technical documentation for the developed POC, including the system architecture, data flows, and code explanations.
- Prototype Presentation for Peer Review (Group Assignment)
    - A final demonstration of the POC to stakeholders, highlighting key features and innovative aspects.
    - Include a slide deck explaining the thought process, challenges faced, and lessons learned.
- Peer Reviews (Individual Assignment)
- Weekly Activity Reports (WARs – Individual Assignment)
- Team Status Report (TSR – Group Assignment)
- Project Plan (Group Assignment)
- SRS, SDD, STP & Dev Doc (Group Assignment)
- Final Report Package (Group Assignment)
    - Final Report (Group Assignment)
    - Source Code (Group Assignment)
    - Website (Group Assignment)
    - Video Demo (Group Assignment)

8.2 Project Plan Except

Project Overview

The Georgia Symphony Orchestra (GSO) website transformation project aims to create a unified proof-of-concept (POC) platform combining the current separate sites for GSO, GYSO, Jazz, and Chorus into a cohesive, user-friendly system. This locally hosted prototype, developed using WordPress, will include features like centralized event management, role-based access simulation, dynamic content, accessibility tools, and multimedia integration. The project focuses on enhancing user experience, ensuring mobile responsiveness, and showcasing advanced computing techniques such as mock API integrations and analytics dashboards. Deliverables include a functional prototype, technical documentation, and a final demonstration highlighting innovative solutions and technical feasibility.

Project specific Milestones & Important dates

- January 26- Project plan & Gantt charts due
- February 2- SRS, SDD

- February 12- Milestone 1 due
    - Milestone 1- Initial Requirement Analysis
        - Deliverables: Wireframes, architectural layout, and design approval.
- March 9- Development document due
- March 12- Milestone 2 due
    - Milestone 2- Development of Core Features
        - Deliverables: Functional prototype with unified navigation, branding and event calendar.
- March 27- Prototype Presentation
- March 30- Test plan and report
- April 9- Milestone 3 due
    - Milestone 3- Accessibility and Multimedia Integration
        - Deliverables: Accessibility features, multimedia content integration, and inclusivity tests.
- April 20- Final report draft
- April 28- Final report package
    - Final Report, website, source code
        - Website: link to report, GitHub, and final presentation video

Deliverables & project plan structure

The deliverables will be broken up into 6 phases that meet milestone deadlines. The 6 phases are outlined below and correspond to our Gantt chart.

○ Requirements phase (Start-Jan 29nd) ○ Meet with stakeholder(s) SH ○ Project plan Document ○ Software research ○ Team Logistics, General planning ○ UX/UI Philosophy research ○ Scope concerns research ○ Current website analysis ○ Github Configuration ○ SRS ○ SDS	○ Project design phase (Jan 29th- Feb 12th) ○ Define tech required * ○ Database design ○ ERD design concepts ○ learning MariaDB/SQL server ○ Creating Database ○ Testing/review database ○ Front End design ○ Create site wireframes ○ WordPress tutorial(s) ○ Figma Tutorial(s) ○ Learning GSO brand guide ○ Learning Ovix API ○ Figma mock-up ○ Backend research ○ Backend implementation

○ prototype dev. phase (Feb 12th-Feb 26th) ○ Develop working prototype ○ Adapting Figma to WordPress ○ Implementing database ○ hosting Prototype ○ Test prototype	○ Development Phase 2(Feb 26th-Mar 12th) ○ Review prototype design ○ notes/Scripts for class presentation ○ Class presentation practice ○ Update functional req. ○ Update nonfunctional req. ○ Redefine Scope ○ User analytics (start) ○ Hotjar/other analytics research ○ platform compatibility (start) ○ Site navigation ○ Calendar ○ Development Documents ○ E-Commerce (Start) ○ Phase 2 testing ○ Phase 2 revisions
○ Development Phase 3(Mar 12th-Apr 9th) ○ User analytics (implement) ○ Platform compatibility (implement) ○ E-Commerce (Implement) ○ Multimedia integration ○ Security ○ CIA Triad analysis ○ SSL/TLS ○ Security plug-ins	○ Final report Phase (Apr 9th-Apr 28th) ○ Document updated design ○ Stakeholder user manual ○ Presentation script/notes ○ Presentation compilation ○ Presentation preparation ○ Video Preparation ○ Documentation revisions ○ Final packed ○ Review final package ○ Final report submission to D2L and project owner

Requirements

- Home page integrating GSO, GYSO, Jazz, and Chorus into a unified structure
  - Showcase Program Diversity
    - Highlight Programs in a Unified Way: Develop a “What We Offer” page showcasing all programs (youth, jazz, chorus, and orchestral performances) under a cohesive GSO umbrella.
    - Integrate Multimedia Content: Feature video highlights, artist interviews, and behind-the-scenes stories from all programs to engage and inspire audiences.
  - Enhance Accessibility and Inclusivity
    - Enhance Accessibility Features: consider including screen-reader compatibility, large-text options, and sensory-friendly concert labels to ensure inclusivity for all users.
    - Strengthen Communication and Engagement
    - Leverage a Centralized News and Blog Section: Dedicate a place to publish unified stories, achievements, and updates about all GSO programs to emphasize their shared mission and increase audience connection.
    - Implement Targeted Landing Pages for Campaigns: Design specific pages for campaigns like fundraising, volunteer recruitment, and corporate sponsorships to attract diverse audiences and stakeholders.

- Optimize Digital Experience
    - Optimize for Mobile Users: Ensure the website is responsive, offering an exceptional user experience across devices, particularly mobile.
    - Integrate Advanced Analytics: consider simulating features to track user interactions and provide insights for optimizing site performance and engagement strategies.
    - Sections for each program with sub-navigation for their specific offerings.
    - Interactive elements like sliders, tabs, and multimedia galleries.
    - Mock booking system for events, demonstrating different categories of tickets.
- A document for the stakeholder describing different means to migrate and/or maintain the project.

- Functional Features - chose one or more after discussion with your capstone professor:
    - User Analytics: Implement sample dashboards showing metrics such as page visits, user demographics, and engagement levels.
    - Dynamic Content: Simulate features like upcoming events, blogs, and volunteer opportunities.

- Backend & Technology Demonstrations
    - Sample CMS implementation.
    - Database simulation for storing events, user feedback, and analytics data.
    - Mock API integration for external services like Google Maps or email newsletters.

- Documentation
    - User manual outlining website navigation and key features.
    - Technical documentation for the developed POC, including the system architecture, data flows, and code explanations.
- Prototype Presentation for Peer Review (Group Assignment)
    - A final demonstration of the POC to stakeholders, highlighting key features and innovative aspects.
    - Include a slide deck explaining the thought process, challenges faced, and lessons learned.

Group Meeting Schedule Date/Time

The group will attempt to meet via teams on a weekly basis on Mondays at 5pm. As a contingency we have established that Wednesdays at 5pm or Thursday at 10am are potential meeting times.

Collaboration and Communication Plan

Meetings entirely in teams. Figma will be used to facilitate mock-ups of the website in the planning and early development phases, and GitHub is where code will be collaborated on.

Communication will revolve around status updates on deliverable and questions about joining the project components together.

Version Control Plan

Our project involves analyzing, designing, and developing a website. Due to the wide range of tasks and requirements, we will use a variety of services. We will utilize Github's version control for our collaboration in regard to research scripting and backend development. Our GitHub will be under an organization named **05T2-GSO** and will have repositories configured with feature, development, and main (live) branches. All members of the team will be owners of this organization. This will allow us to have a well-organized SRP (Single Responsibility Protocol) method of maintaining the code within this project's wide scope.
Outside of the codebase and scripts, design documents will maintain versioning via Figma's Version Control History, Fig Jam and Wireframe systems. After each major design, we will retain a major version copy of the wireframe and components in a separate file.
Lastly, when it comes to database and deployment versioning, we will utilize a variety of setup scripts for controlling and performing rollbacks on the databases (MariaDB) via a docker container snapshot. We will also maintain WordPress through a variety of methods depending on the status of the project. WordPress Rollback will be utilized for themes and plugins, while GitHooks will be used for database integration. Lastly, complete versions of the website will be saved again via docker container snapshots and images to allow for easy spin-up and evaluation.

8.3 Project SRS Excerpt

Introduction

1.1 Overview

This document defines the software requirements for the Georgia Symphony Orchestra (GSO) website transformation project, which aims to consolidate the individual websites for GSO, GYSO, Jazz, and Chorus into a unified, locally hosted proof-of-concept (POC) platform. It details the system's intended functionality, design constraints, and operational environment to provide developers and stakeholders with a clear, technical foundation for implementation. This SRS is complemented by separate project **management** and quality assurance plans covering scheduling, cost estimation, development methodology, and testing procedures. Additional sources such as design prototypes and user interface storyboards from the kickoff meeting are referenced to ensure comprehensive coverage of requirements.

1.2 Project Goals

The primary goals of this project are:

- Integrate separate websites into a single cohesive interface.
- Develop an intuitive navigation system and responsive design that adapts seamlessly across desktop and mobile devices.
- Ensure robust accessibility features to accommodate users with varying technical proficiencies.

1.3 Definitions and Acronyms

GSO: Georgia Symphony Orchestra
GYSO: Georgia Youth Symphony Orchestra
POC: Proof of Concept; a demonstration system to validate feasibility and design.
CMS: Content Management System

PHP: Hypertext Processor
RBAC: Role-based access control
API: Application Programming Interface

1.4 Assumptions

The project presumes the availability of third-party APIs for simulating dynamic event data and weather updates, and that all integrated plugins are compatible with the chosen CMS version. Additionally, it is assumed that all team members have access to standard development tools such as Git for version control and that external dependencies, such as web hosting for demonstrations, will meet the project's performance and security requirements.

2.0 Design Constraints

2.1 Environment

The system will be developed as a locally hosted POC using WordPress as the CMS. Custom plugins and integrations will be implemented using JavaScript and PHP. The system will interface with simulated external APIs to demonstrate dynamic functionality.

The system will operate independently; however, it will reference standards and practices from the existing GSO websites for consistency.

2.2 User Characteristics

The website is designed for a diverse audience, including administrators, editors, and general users, with a strong emphasis on accessibility for older adults. Administrators will have the ability to manage events, upload content, and modify site settings, requiring a moderate level of technical proficiency. Editors will focus on content updates such as adding blog posts, images, and event details, necessitating a user-friendly content management interface.

General users, including concert attendees, donors, community members, students, potential students, parents, educators, and employees who may have varying levels of technical expertise. To ensure ease of use, the interface will feature intuitive navigation, clear labels, and large-font readability. Additionally, accessibility features such as keyboard shortcuts, screen-reader compatibility, and simplified navigation options will be prioritized to accommodate users with disabilities or limited digital literacy.

3.0 Functional Requirements

The website will feature a modern, unified design with high-quality graphics optimized for both desktop and mobile devices. It will run on major operating systems via standard web browsers and adhere to responsive design principles. The system is developed using WordPress as the

CMS, with PHP and JavaScript for custom functionalities. Key constraints include maintaining compatibility across devices and ensuring optimized performance within these environments

3.1 Homepage

- Display unified branding for GSO, GYSO, Jazz, and Chorus.
- Feature upcoming events with prominent "Buy Tickets" buttons.
- Include a brief mission statement and a global search bar with filters.

3.2 Event Management

- Present event listings in calendar and list views.
- Enable filtering by program, location, and date.
- Provide an event details page with descriptions, multimedia, ticket purchase options, and social sharing buttons.
- Parking & location information

3.3 Program Page

- Create dedicated pages for each program with multimedia content.
- Include sections for joining and volunteering.

4.0 Non-Functional Requirements (use if applicable)

4.1 Security

- The system shall implement simulated role-based access control (RBAC) to ensure that administrators, editors, and general users have appropriate permissions.
- All data input shall be sanitized to mitigate potential security threats such as SQL injection or cross-site scripting.
- Secure communication protocols (e.g., HTTPS) are required for data transmission, even in the POC environment.

4.2 Usability

- The user interface must adhere to modern usability standards with a clean, intuitive design that minimizes the learning curve.
- The system shall be compliant with accessibility standards (e.g., WCAG) to accommodate users with disabilities, including support for screen readers, text resizing, and full keyboard navigation.
- Consistent navigation and clear labeling are required to ensure a seamless user experience.

4.3 Scalability

- Although a proof-of-concept, the design should accommodate future expansion, such as live API integrations and a robust database for real-time event management.

5.0 External Interface Requirements (use if applicable)

5.1 User Interface Requirements

The website's user interface shall be intuitive and accessible, employing a clean design that supports responsive layouts for both desktop and mobile devices. It will utilize standard web elements such as navigation menus, buttons, and forms to ensure ease of use. Additionally, the

interface must adhere to accessibility guidelines, providing features like high contrast modes, text resizing, and compatibility with screen readers to accommodate users with disabilities.

5.2 Hardware Interface Requirements

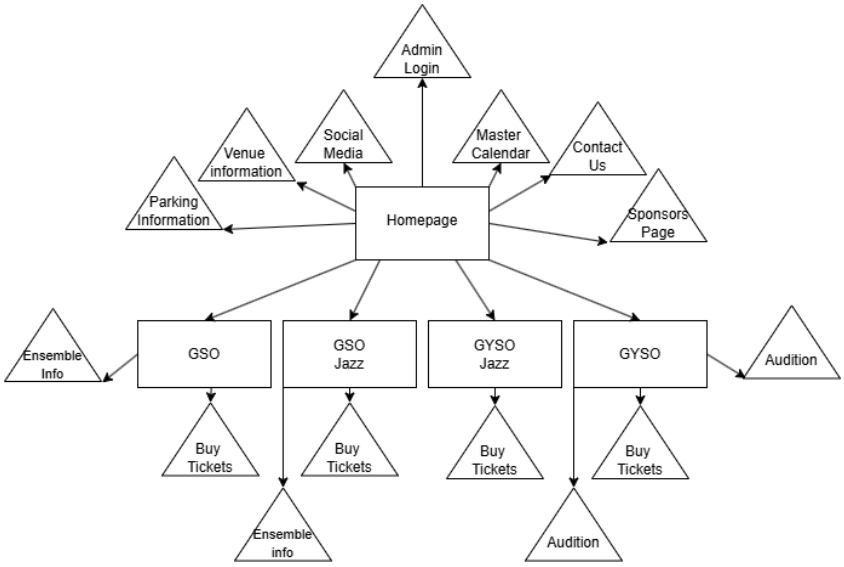
As a web-based application, the system is designed to operate on standard computing hardware, including desktops, laptops, tablets, and smartphones. It is expected to function correctly across devices that support modern web browsers. No specialized hardware interfaces are required; however, the system must be optimized for various screen sizes and input methods, including touch and keyboard inputs.

5.3 Software Interface Requirements

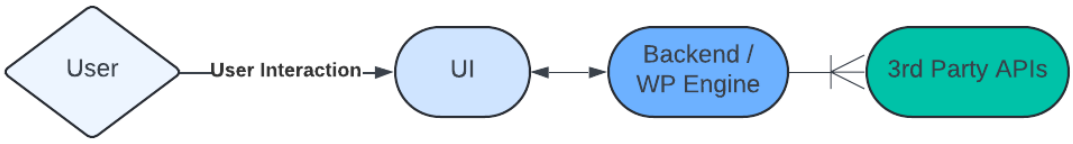
The website will be built on the WordPress platform, with custom functionalities developed in PHP and JavaScript. It will interface with third-party APIs to simulate dynamic content, such as event data and weather updates. The system is also required to integrate with standard web services and plugins, and it should provide compatibility with MySQL for storing event and user data.

6.0 External Interface Requirements (use if applicable)

6.1 Initial flowchart of user interaction with site



6.2 Abstract interaction of user with system as a whole



8.4 Gantt chart predicted and actual

See next page-

Project Name: 05-T2 Website Transformation
Report Date: 1/26/2025

Projected Hours

Phase	Tasks	Complete%	Current Status Memo	Assigned To	Milestone #1				Milestone #2				Milestone #3				Final Demo	
					01/22	01/29	02/05	02/12	####	02/26	####	03/12	####	03/26	04/02	04/09	04/16	04/23
Requirements	Meet with stakeholder(s) SH	100%	Weekly meetings	All	1	1	1	1	1	1	1	1	1	1	1	1		
	Project plan Document	100%	Complete	NR	2													
	Software research	100%	Database ? Backend ?	KJ	1	1												
	Team Logistics, General planning	100%	Comperhensive cal. Set	N	1	1												
	UX/UI Philosophy research	100%	select articles being rea	J	1	1												
	Scope concerns research	100%	Scope set	All	1	1												
	Current website analysis	100%	XML mapping	J	1	1												
	Github Configuration	100%	Set up, all joined	J		1												
	SRS	100%	Software research	KR		2												
	SDS	100%	Software research	KR		2												
Project design	Define tech required *	100%		All	3	1												
	Database design	100%		N		1	2											
	ERD design concepts	100%		N		2	1											
	learing MariaDB/SQL server	100%		N		1	3											
	Creating Database	100%		N		1	3											
	Testing/review database	100%		N		1	1											
	Front End design	100%		J		1	1											
	Create site wireframes	100%		All		2	2	2										
	Wordpress tutorial(s)	100%		All		2	2	2										
	Figma Tutorial(s)	100%		All		2	2											
	Learning GSO brand guide	100%		R		1	1											
	Learning Ovix API	100%		JKN		2	2											
	Figma mock-up	100%		All		3	2	2										
	Backend research	100%		K		3	3											
	Backend implementation	100%		KJ		3	3	2										
Development	**Develop working prototype**	100%		All		3	2	10										
Protoype Phase	Adapting figma to wordpress	100%		KJR			3	4										
	Implementing database	100%		N			2	7										
	hosting Prototype	100%		J				7	3									
	Test prototype	100%		R				3	2									
Development	**Review prototype design**	100%		All					1	1								
Phase 2	notes/Scripts for class presentatior	100%		R					1	2								
	Class presentation practice	100%		All					1	1								
	Update funtional req.	100%		R					3	3	2							
	Update nonfunctional req.	100%		R					3	2	2							
	Redefine Scope	100%		All					1	1	1							
	User analytics (start)	100%		JK					5	2	2	3						
	Hotjar/other analytics research	100%		JKN					3	2	2	3						
	platform compatibility (start)	100%		J					1	2	2	1						
	Site navigation	100%		JKN						4	4	4						
	Calendar	100%		NR						4	4	4						
	Deveopment Documents	100%		R						2	3	1						
	E-Commerce (Start)	100%		KJ						4	4	2						
	Phase 1 testing	100%		KJ							2	2						
	Phase 1 revisions	100%		KJ							6	6	1					
Development	User analytics (implement)	100%		NJ								1	3	3	3			
Phase 3	Platform compatibility (implement	100%		J									3	3	3			
	E-Commerce (Implement)	100%		K									3	3	3			
	Multimedia integration	100%		JK									4	3	3			
	Security	100%		NR									3	1	1			
	CIA Triad analysis	100%		NR									1	2	1			
	SSL/TLS	100%		NR										2	2			
	Security plug-ins	100%		NR										2	2	2		
Final report	Document updated design	100%		RJ								1	3	3	4	2	2	
	Stakeholder user manual	100%		All								1	1	1	1	3	1	
	Presentation script/notes	100%		R											1	2	3	
	Presentation compilation	100%		R											1	3	3	
	Presentation preparation	100%		R											3	3	2	
	Video Preperation	100%		R											1	4	1	
	Documentation revisions	100%		R											2	2	2	
	Final packed	100%		All												1	1	
	Review final package	100%		All												1	1	
	Final report submission to D2L and project owner	100%		All														1
			Total work hours	**368**	**11**	**40**	**36**	**40**	**25**	**31**	**35**	**30**	**23**	**24**	**32**	**24**	**16**	**1**

Actual Hours

Phase	Tasks	Complete%	Current Status Memo	Assigned To	Milestone #1				Milestone #2				Milestone #3				Final Demo	
					01/22	01/29	02/05	02/12	02/19	02/26	03/05	03/12	03/19	03/26	04/02	04/09	04/16	04/23
Requirements	Meet with stakeholder(s) SH	100%	Weekly meetings	All	1	1	1	1	1	1	1	1	1	1	1	1		
	Project plan Document	100%	Complete	NR	3													
	Software research	100%	Database ? Backend ?	KJ	2	1												
	Team Logistics, General planning	100%	Comperhensive cal. Set	N	1	1												
	UX/UI Philosophy research	100%	select articles being rea	J	1	1												
	Scope concerns research	100%	Scope set	All	2	1				2								
	Current website analysis	100%	XML mapping	J	1	1												
	Github Configuration	100%	Set up, all joined	J		1												
	SRS	100%	Software research	KR	2	2												
	SDS	100%	Software research	KR	2	2												
Project design	Define tech required *	100%		All	3	4												
	Database design	100%		N		1	2											
	ERD design concepts	100%		N		2	1											
	learing MariaDB/SQL server	100%		N		1	3											
	Creating Database	100%		N		1	3											
	Testing/review database	100%		N		1	1											
	Front End design	100%		J		1	1											
	Create site wireframes	100%		All		2	2	2										
	Wordpress tutorial(s)	100%		All		3	3	2										
	Figma Tutorial(s)	100%		All		2	3											
	Learning GSO brand guide	100%		R		1	1											
	Learning Ovix API	100%		JKN		2	2	1										
	Figma mock-up	100%		All		3	2	2										
	Backend research	100%		K		3	2											
	Backend implementation	100%		KJ		1	1	2										
Development	**Develop working prototype**	100%		All			2	5	5									
Protoype Phase	Adapting figma to wordpress	100%		KJR			1	1										
	Implementing database	100%		N			2	7										
	hosting Prototype	100%		J				7	3									
	Test prototype	100%		R					2				1			1		
Development	**Review prototype design**	100%		All					1	1								
Phase 2	notes/Scripts for class presentatior	100%		R					1	2								
	Class presentation practice	100%		All					1	1								
	Update funtional req.	100%		R					1	1	2							
	Update nonfunctional req.	100%		R					3	2	2							
	Redefine Scope	100%		All					1	1	1	1						
	User analytics (start)	100%		JK					1									
	Hotjar/other analytics research	100%		JKN					3									
	platform compatibility (start)	100%		J					1	2	2	1						
	Site navigation	100%		JKN								4	4	4				
	Calendar	100%		NR								4	4	5				
	Deveopment Documents	100%		R						2	3	1						
	E-Commerce (Start)	100%		KJ								2						
	Phase 1 testing	100%		KJ							2	2						
	Phase 1 revisions	100%		KJ								6	1	5				
Development	User analytics (implement)	100%		NJ								1						
Phase 3	Platform compatibility (implement	100%		J									3	3	3			
	E-Commerce (Implement)	100%		K									1					
	Multimedia integration	100%		JK									1	1	1			
	Security	100%		NR									3	1	1			
	CIA Triad analysis	100%		NR									1					
	SSL/TLS	100%		NR										2	2			
	Security plug-ins	100%		NR										2				
Final report	Document updated design	100%		RJ								1	3	3	4	2	2	
	Stakeholder user manual	100%		All												3	1	
	Presentation script/notes	100%		R											1	2	3	
	Presentation compilation	100%		R											1	3	3	
	Presentation preparation	100%		R											3	3	2	
	Video Preperation	100%		R											1	4	1	
	Documentation revisions	100%		R											2	2	2	
	Final packed	100%		All												1	1	
	Review final package	100%		All												1	1	
	Final report submission to D2L and project owner	100%		All														1
			Total work hours	**306**														

* formally define how you will develop this project including source code management

Legend		*Names*
Planned		Noah- N
Delayed		Rashmi-R
Number	Work: man hours	Johnathan- J
		Kevin-K

Class Due dates	
26-Jan	Plan, Gantt, RSRS, SDS
9-Mar	Development Docments
27-Mar	Protoype presentation
30-Mar	STP, STR
20-Apr	Draft report due
28-Apr	Final Package

Stakeholder Due Dates	
12-Feb	Mock-up & Architechure
19-Mar	functional Prototype
16-Apr	Accessible, and multimedia demonstrations