CGT 141 Project

Alejandro Barroso

2024-10-27

CGT 141 Project Development Report

Project Overview

For this project, we will create a fully functional, responsive website and a native Android mobile app for a fictitious client. The project involves hosting on a VPS and establishing a domain, and following Purdue's requirements for content, structure, and functionality.

Development Phases

1. **Topic Selection**: Identify a suitable subject for the client project.
2. **Team Setup**: Define roles, communication expectations, and create an online workspace.
3. **Initial Mock-up**: Develop and upload a 1-page draft on Brightspace for feedback.
4. **Scope and Functional Specification**: Define project scope and functional requirements.
5. **Mock-up Revision**: Update the mock-up with feedback incorporated.
6. **Site and App Creation**: Implement the final website and mobile app using the specifications.
7. **Packaging for Android**: Use Cordova or WebIntoApp to package the project as a .apk file.
8. **Debugging and Finalization**: Address any issues and prepare final deliverables.

Folder Structure

The project folder structure is essential for efficient organization, debugging, and delivery. Below is the directory layout:

- [**Project.zip**]
  - **Documents**
    - `PreProduction.pdf` — Pre-production documentation, mock-ups, etc.
  - **WWW** — Web documents for the website
    - `index.html` — Homepage (starting point of the app)
    - **other pages** — Additional pages (7 minimum)
    - **CSS**
      - `style.css` — CSS for layout, design, and responsiveness
    - **JS**
      - `main.js` — JavaScript files for interactivity
    - **Images** — Image assets (PNG, JPG, GIF)

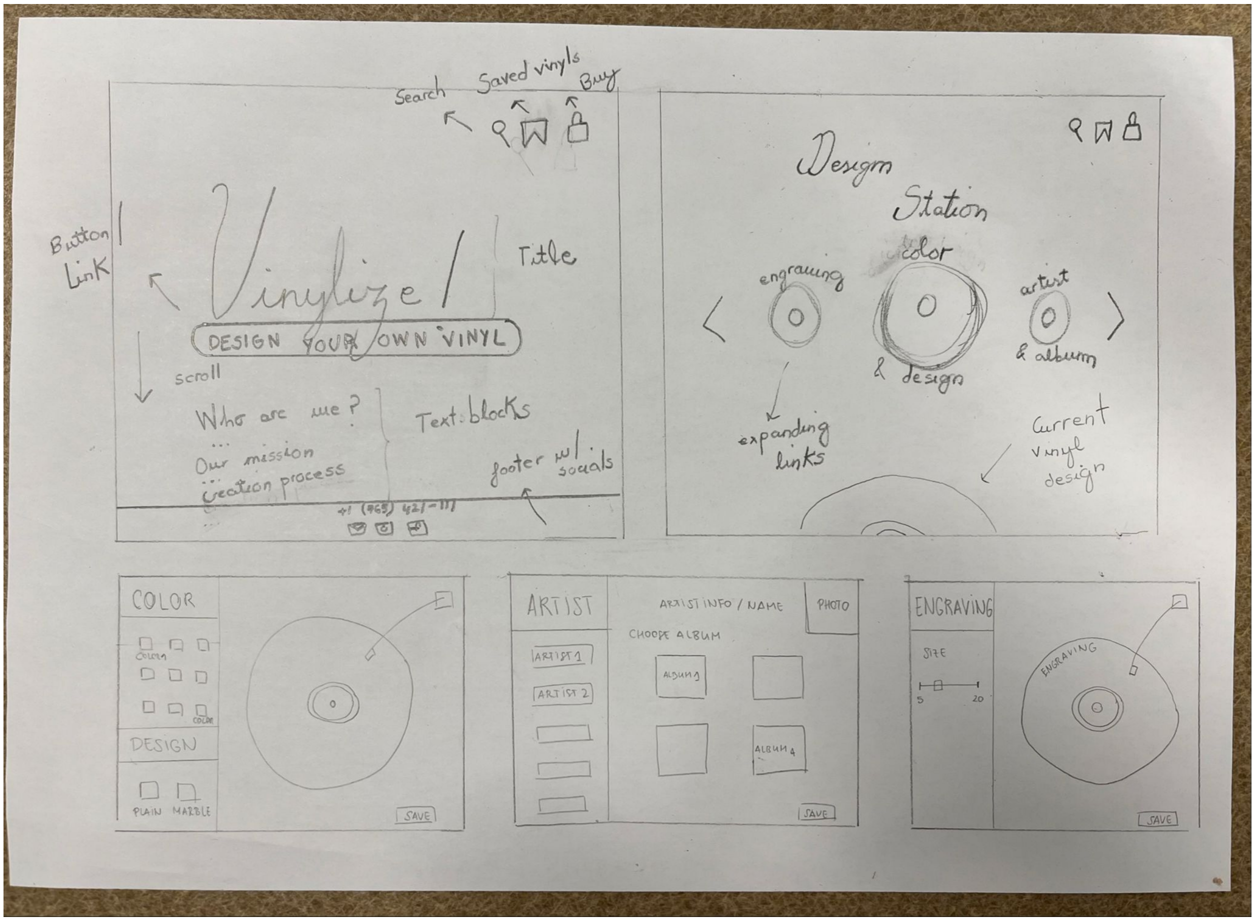
    - **assets** — Reusable page components (e.g., nav, footer, database connection files)
  - **Mobile App**
    - `Project.apk` — Compiled Android app file

Detailed Development Plan

1. Topic Selection and Scope Definition

Vinylize is an interactive platform where users can create custom vinyl records from start to finish. After registering and logging in, users can design their vinyl with a personalized cover art and select a custom image for the vinyl's center label. With a variety of vinyl colors to choose from, Vinylize offers endless possibilities for creating a truly unique record. Once the design is complete, users can save their creation in their account and proceed to purchase, making their personalized vinyl a one-of-a-kind collectible or gift.

2. Mock-up



3. Project Development Checklist

Technical Implementation (200 points)

☒ **7 Content Pages**: The website includes a minimum of 7 unique content pages.
☒ **DTD Validation**: The site validates against XHTML 1.1 STRICT or HTML5.
☒ **Homepage**: Includes a home page with the filename `index.html`.
☒ **Responsive Design**: The website is responsive for all devices (use Bootstrap, media queries, etc.).
☒ **Intrasite and Extrasite Links**: Has both internal links (within the site) and external links (outside the site).
☒ **Document Links**: At least one page has an anchor link to a section within the same document.
☒ **Semantic HTML**: HTML is structured semantically for accessibility and readability.
☒ **Form with Validation**: Contains at least one form with five elements, validated using JavaScript or HTML5/CSS3.
☒ **Aesthetics and Navigation**: The site is visually appealing with an effective navigation scheme.
☒ **Graphics**: Includes at least one PNG, one JPG, and one GIF image.
☒ **Efficient Design**: The site is optimized for download speed and organized backend structure.
☐ **Mobile App Packaging**: The website is packaged as an Android app using Cordova or WebIntoApp.
☒ **Device-Sized Layout**: Properly optimized for mobile device viewing.

CSS Requirements

☒ **Responsive CSS**: Uses media queries or JavaScript for layout adjustments on different screen sizes.
☒ **CSS Formatting**: All formatting is done using CSS.
☒ **CSS Layout**: All layout is structured using CSS.
☒ **Styled Links**: Links are presented as a styled unordered or definition list.
☒ **Floated DIV**: At least one `DIV` element is floated.
☒ **Clear Property**: Uses the `clear` property at least once.
☒ **Box Model**: Uses box model properties (border, margin, padding).
☒ **Multiple CSS Selectors**: Utilizes multiple CSS selectors (IDs, classes, etc.).
☒ **CSS Properties**: Includes background, image, table, and text CSS properties.

JavaScript or HTML5/CSS3 Requirements

☒ **Rollover Buttons**: At least one button set with JavaScript or CSS3 hover effects for rollover functionality.
☒ **Form Validation**: JavaScript or HTML5/CSS3 checks to ensure form fields aren't left blank.

Hosting Requirements

☐ **LAMP/LEMP Stack**: Hosted on a VPS with a LAMP or LEMP environment.
☐ **HTTPS**: The site uses HTTPS for secure connections.
☐ **Domain & DNS**: Configured with a properly set up domain name and DNS.

Additional Grading Criteria (100 points)

☐ **Just-In-Time Assignments (25 points)**: Completion of assignments like server setup, documentation, and pitch video creation.
☐ **Quality Testing and Validation (25 points)**: Tested across multiple devices and browsers with issues resolved.
☐ **Production Documentation (25 points)**: Includes functional specification, user acceptance tests, mock-ups, flowcharts, and preproduction documentation.
☐ **Client Management (25 points)**: Incorporates feedback from TA and classmates, showing improvements in response to guidance

4. Development

HTML Pages, some of them are php files since we need to manage server side features like saving the user designs.

1. **Index.html(floated DIV, clear to manage text-img structure)**
2. **design-station.php**
3. **select-artist.php**
4. **select-color.php**
5. **select-engraving.php**
6. **buy.php**
7. **saved-vynils.php**

CSS Structure

- **Bootstrap:** We'll use Bootstrap for most of the CSS design. Bootstrap provides a set of ready-made styles and layouts that make it easy to create a responsive and visually appealing design. It also has predefined styled links.
- **style.css**
- styles for each page, as needed

JavaScript Requirements

- **Form Validation**: JavaScript or HTML5/CSS3 will validate form inputs.
- **Interactive Elements**: Rollover buttons using JavaScript and/or CSS3 hover effects.

Packaging for Mobile

- Use **Cordova** or **WebIntoApp** to build an `.apk` file, ensuring all media and HTML resources are packaged in a ZIP file for ease of compilation.

Testing and Debugging

1. **Quality Testing**: Ensure functionality across multiple devices and browsers.
2. **Validation**: Validate the site against XHTML 1.1 or HTML5 standards.
3. **Debugging**: Use debugging tools (e.g., Chrome DevTools) to solve any issues and improve load times.

Deliverables

ZIP File Submission

The final project will be a ZIP file named `[LastName]_Project.zip`, containing: - **Documents Folder**: Pre-production PDF with mock-ups and flowcharts. - **WWW Folder**: All web documents, organized into CSS, JS, Images, and individual pages. - **Mobile App Folder**: Final `.apk` file.

Server Deployment

Deploy the website on the selected VPS, configuring HTTPS and DNS settings for the chosen domain.

Client Management

We will treat feedback from the class and TA as client input, iterating on mock-ups, functionality, and design. All modifications will align with Purdue's guidelines and be justified in technical terms if needed.

This approach ensures a structured and methodical path to project completion, with clear objectives, milestones, and deliverables.