Project Wixs: Technical Manual

COSC 4F00

Prepared By

Group 02

Drayton Williams (5925342) | **Jasdeep Grewal** (5757828) | **Cameron Hammel** (5808746) | **Ian LeMasters** (5877667) | **Curtis Honsberger** (6630362) | **Liam Howes** (5880331) | **Kieran Colaco** (6054654) | **Nathan Hellinga** (6002620)

Installation Manual

Based on the current system setup, no unorthodox installation process is required for access. The system is hosted on Brock's Sandcastle server and available as a publicly accessible web application. All that is needed for access is a web browser. The web application can be found at the following link: http://cosc.brocku.ca/~c4f00g02/projectWixs/#/

To use the system as we intended, please ensure all cookies have been cleared for the domain 'cosc.brocku.ca' on initial launch. We make use of browser alerts to accurately identify server responses, so we ask that users do not disable browser alerts for the site. Also, for the site to function, JavaScript must be enabled.

System Architecture

Front-End

The front-end makes use of React.js, which handles the communication and organization of various front end components that make up the user's interface. It is here that values are transferred and temporarily stored in React constructs called state and props. Each React component can range in size from being an entire viewable page, down to the cards that populate on the user's dashboard. State and props are the main tool of data transmission around the app and are often the values being fed into parameters that are sent along to the back-end via an Axios request. In order to maintain data over page refreshes and possible restarts to the entire application, it is important that the data communicates with some form of database. The front-end is tested using a Node.js development and npm's dependency management. When everything is ready for deployment, the React package is bundled and minified into a production build using Parcel.js and deployed on the team's dedicated server location at 'http://cosc.brocku.ca/~c4f00g02/projectWixs'.

Middle-Ware

Communication between the front-end and back-end is achieved by making HTTP requests and sending data to PHP endpoints acting as a RESTful API. React makes use of a module called Axios which allows for the asynchronous sending and receiving of data to these PHP endpoints. Data is sent to a PHP endpoint, like http://cosc.brocku.ca/~c4f00g02/projectWixs/addUser.php,

and the system will still function as normal as it waits for the response. Usually the response will trigger a data change, page reload, or page redirect, after the user has been alerted of what has just occurred.

Back-End
Each PHP script in need of access to the Postgres database obtains credentials from a hidden file (via file permissions) and uses those credentials to establish a connection with the Postgres database. Depending on the request type, most PHP files will gain access to variables sent via the axios request by pulling them from the GET or POST values. From there, the PHP file can perform any operations necessary to facilitate a safe and efficient submission of data to our database. This can come in the form of checking whether a user with submitted credentials already exists, sanitizing/validating inputs, or prepping the response object to be sent back containing a success flag and possible tuples obtained from the database tables.

Maintenance Manual

Issue Resolution Procedures | Change Management Process

[As reflected in SRS]

When running into a system issue, the Project Wixs team follows a structured process in order to ensure any alterations are handled in as efficient of a manner as possible. When a change is implemented, the team will be sure to maintain an updated source code, repository, message board, and SRS. Below the tools used to aid in the management of system changes are described:

Discord
Within the Project Wixs Discord, the team is able to handle most of the detailed communication necessary to initiate, process, and finalize a system change. On an ongoing basis, new ideas are internally formulated and vetted for feasibility then delegated to members based on experience, schedule, and necessity. Ultimately, Discord is the main message board used to maintain a cohesive flow of communication within the team.

GitHub
GitHub acts as the team's main repository for the source code, issue management, and task delegation. Internally, when it is decided a change is needed to be made, the issue will be created and described in as thorough detail as possible to ensure no information can be misinterpreted. The issue is then delegated to member(s) based on previous communications within the aforementioned message board. When an issue is being worked on, it will be tested and updated within a separate 'development' branch before being reviewed by members involved in the change, and merged with the 'master' branch upon approval. Also a project roadmap is used to better visualize the current progress of issues within a project. External users of the system (non-team members) who wish to see a change may visit the repository to fork, branch from, or produce a new issue for the Project Wixs team to review.

Google Drive
The Google Drive allows the team to store any files necessary for recurring usage by Project Wixs team members. Most importantly, here is where the SRS is stored and regularly updated within a Google Doc. The benefit of documenting changes here is that members can concurrently edit, view, and comment on changes being made by other team members. After a change has been discussed within the message board then completed within the GitHub repository, the change will be reflected within the SRS as a final statement reflecting the details of the change that has been made.

pgAdmin
In order to gain control of any back-end processes, members of the team will access the Postgres database using pgAdmin to have a well-visualized idea of what issues are occurring or what changes need to be made. pgAmin allows the team a detailed view of when transactions occur, allows for editing of table properties, and viewing of table data.

User Manual

Usage Walkthrough

Accessing the system

- Live website can be found at http://cosc.brocku.ca/~c4f00g02/projectWixs/
    - We recommend clearing cookies, allowing browser alerts, and enabling JavaScript for the domain 'cosc.brocku.ca' in order to operate the website as intended.
- Login and register button can be found at the top right of the header of the bottom of the landing page to bring you to the login or register page, respectively.
- After creating an account, you will be brought to your dashboard page.

Your dashboard

- If logging in for the first time you will be met with a dashboard empty of any templates.
    - From here you can create up to 4 new templates.
- Once a template Is created, you have the option to set it as active.
    - Once active, others can see your site using the “shareable live template link” found at the top of the page.
- You also have the option to edit or delete any template.
    - Clicking the edit button brings you to the editor.

The editor
The editor allows users to easily create their own website. This tool supports adding a range of basic components as well as layout controls and custom styling features. The buttons to access the various editor features are displayed on the right side of the screen. Components are arranged in a draggable grid layout, most can also be resized via a handle at the bottom right of their box.

- Clicking on the yellow plus sign on the right side of the screen opens the **Component Panel**. The Component Panel provides the following functionality:
    - Add new elements to the page

- The Components Panel has the following elements:
    - Plain Text Area - a standard textbox
    - Rich Text Area - a textbox that allows text styling such as bold, underline, lists etc.
    - Text With Header - a Rich textbox with a header section
    - Button - a standard button (functionally does nothing when clicked)
    - HTML Box - a section where the user can add their own HTML code. This component is intended for expert users of the system that need a highly specific element. Improper use of this component can cause unintended effects on the users template.
    - Table - allows multiple rows and columns as a typical table would. The table is limited to 5 rows and 5 columns to demonstrate that functionality but it could just as easily be unlimited.
    - Register Form - a form with “Email”, “Password” and “Repeat Password” fields and a Register button (again the button does nothing)
    - Login Form - a form with “Email”, “Password” fields and a Login button (button does nothing)
    - Image - allows users to upload an image along with text options for the image. Useful for providing image context or descriptions if required.
    - Standard Header - a standard Header
    - Standard Footer - a standard Footer (similar to the header but contains copyright symbol)
    - Flexible Card - A Card that contains a Header, Body, and Footer section, along with the option to add an image

- Clicking on a component in the Component Panel places it onto the site.
    - Once on the site, said component can be dragged around, edited and styled.
    - Components get added underneath each other, use the scroll bar to scroll down and see the elements you added if they are not visible
- When a component is selected, the **Data Panel** (orange pencil) will become visible. This panel gives access to modifying the contents of components; clicking apply will automatically update the selected component.
    - Components can also be deleted from this tab by clicking on the ‘trash can’ icon at the top right of the panel.
- The **Style Panel** (blue canvas) is also only visible with a component selected and allows users to make additional stylistic changes to the component. This currently includes the ability to change font size, font colour and background color.

- The editor superimposes some graphics onto the site to make it easier while editing. These graphics will **NOT** remain on a published version of the site. The graphic overlays are:
    - Dashed outer border of components
    - Light blue dashed outline of selected component
    - Resize handles in the bottom right of components
    - All drag specific graphics including the red placement indicator

- All changes are automatically saved to the users account in the background. Autosaves are triggered when adding new components, updating the layout of existing components or when applying changes through the data/style panels.

Editor Limitations

The editor is restricted to a fixed width of 1200px. What this means is that on small form factor devices such as mobile phones or small tablets, the page will be displayed with a horizontal scroll bar instead of dynamically resizing content to fit the device. On devices *larger* than 1200px, the content will remain at the fixed width and center to the middle of the screen.

The Editor only supports creation of *single page* applications. This means that there is only a single page to edit and publish, with a single URL.

Additional editor information can be found here:
https://github.com/DrayWilliams1/ProjectWixs/tree/master/wixs-tester/src/Components/Editor

Content upload

- User images may also be inserted onto a template.
- Firstly, upload your image under the “Upload” tab on Project Wixs’ Nav-bar.
    - Image sizes are restricted to 10MB or smaller.
- After uploading, they can be inserted by means of the “image component”.

Any additional help can be found by visiting the Help/FAQ page available within the website. Here can be found both written and video walkthroughs of system processes with additional Q&A. This can be found within the navigation bar or within the right corner of the footer.