Iteration 5

CIS 320-02

Team: BusinessFirst

**Alexander Welsh, Dalton Karl, Anthony Striepe, Jrew Simpson, La’Daysha Wade**

Table of Contents

[Table of Contents 2](#_Toc183384493)

[1. Class Diagram 3](#_Toc183384494)

[1.1 Class Diagram Model 3](#_Toc183384495)

[1.2 Class Diagram Narrative 4](#_Toc183384496)

[2. Database Design 5](#_Toc183384497)

[2.1 Database Diagram 5](#_Toc183384498)

[2.2 Database Narrative 6](#_Toc183384499)

[2.3 Database Definitions 7](#_Toc183384500)

[3. User Interface Diagram 14](#_Toc183384501)

[3.1 User Interface Navigation Diagram 14](#_Toc183384502)

[3.2 User Interface Diagram Narrative 14](#_Toc183384503)

[3.3 Screen Layouts 14](#_Toc183384504)

[3.3A Data Input Screens 14](#_Toc183384505)

[Contact Form 15](#_Toc183384506)

[Newsletter Form 16](#_Toc183384507)

[RSVP Form 17](#_Toc183384508)

[Application Form 18](#_Toc183384509)

[3.3B Data Output Screens 18](#_Toc183384510)

[Contact/Application Submissions 18](#_Toc183384511)

[Newsletter Submissions 19](#_Toc183384512)

[RSVP Submissions 19](#_Toc183384513)

[Application Submissions 20](#_Toc183384514)

[4. Gantt Chart 21](#_Toc183384515)

[4.1 Gantt Chart Diagram 21](#_Toc183384516)

[5. User Interface Prototypes 22](#_Toc183384517)

# 1. Class Diagram

## 1.1 Class Diagram Model

A diagram of a company

Description automatically generated

## 1.2 Class Diagram Narrative

This class diagram outlines a dynamic system designed to handle multiple aspects of organizational operations, including user interactions, donations, events, and administrative tasks. At the center of the design are two key roles: Admin and User. The Admin class is responsible for overseeing essential organizational functions such as managing song catalogs, programs, contracts, donations, applications, and event logistics. On the other hand, the User class represents everyday participants who interact with the system by making donations, RSVP’ing for events, and subscribing to newsletters. The system includes a Donation class, which supports both one-time and recurring contributions, with user-specific records maintained through the User\_Donation class. Events are organized via the Event class, which tracks details like dates, RSVPs, and updates. Administrative efficiency is further enhanced by modular components such as the Sponsors\_Widget and Volunteer\_Hours\_Widget, which simplify managing sponsorships and volunteer hours. Additionally, the Song\_Catalog and Program\_Module classes allow for the management of multimedia content and program-related details, connecting administrators and users for seamless updates. To support communication and engagement, features like the Newsletter, MissionStatement, and Reviews\_Module keep users informed and involved. At the same time, dedicated classes handle specialized tasks such as processing applications and managing contracts. The system is designed with a strong emphasis on modularity, hierarchy, and interconnectedness, creating an effective framework for both user engagement and internal processes.

# 2. Database Design

## A group of text boxes Description automatically generated with medium confidence2.1 Database Diagram

## 2.2 Database Narrative

Our database is designed around the actions of the system admin and website users. Many of our design choices for our website make it so that the data design is fairly simple. The admin has the most relationships and does not require many bridges to map out the system. The admin\_ID attribute is listed with several classes as a way to mark the author of certain changes, such as who changed the mission statement, or who added a sponsor to the carousel. Normalizing the submissions of reviews and donations are what required the most work. In these instances, we were able to separate classes into two distinct ones: one for the admin to review, and the other for the user to create or submit. We accomplished this by having the two classes share the same primary key, with the attributes depending on a combination of the primary key and foreign key from the user or admin. Our diagram branches off into the several options that users are given to go via the ribbon on the landing page. We ensured that classes were covered by following our To-Be model.

## 2.3 Database Definitions

**User**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute | Definition | Type | Size | Key |
| user\_ID | User’s unique identifier | int | 5 bytes | Primary |
| firstName | User’s first name | Str | 256 bytes | N/A |
| lastName | User’s last name | Str | 256 bytes | N/A |
| email | User’s email address | Str | 256 bytes | N/A |
| phone | User’s phone number | Str | 256 bytes | N/A |

**Admin**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute | Definition | Type | Size | Key |
| admin\_ID | Administrator’s unique identifier | Int | 5 bytes | Primary |
| firstName | Administrator’s first name | Str | 256 bytes | N/A |
| lastName | Administrator’s last name | Str | 256 bytes | N/A |
| email | Administrator’s email address | Str | 256 bytes | N/A |
| phone | Administrator’s phone number | Str | 256 bytes | N/A |

**Volunteer**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute | Definition | Type | Size | Key |
| volunteer\_ID | Volunteer unique identifier | Int | 5 bytes | Primary |
| user\_ID | User ID | Int | 5 bytes | Foreign |
| volunteerTitle | Title for volunteer hours | Str | 256 bytes | N/A |
| volunteerHours | Number of volunteer hours | Int | 5 bytes | N/A |

**Newsletter**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute | Definition | Type | Size | Key |
| newsletter\_ID | Newsletter’s unique identifier | Int | 5 bytes | Primary |
| admin\_ID | Administrator’s unique identifier | Int | 5 bytes | Foreign |
| distributionList | Distribution list of users | Str | 5 bytes | N/A |
| content | Contents of the newsletter | Str | 256 bytes | N/A |

**SongCatalog**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute | Definition | Type | Size | Key |
| songCatalog\_ID | Song Catalog’s unique identifier | int | 5 bytes | Primary |
| admin\_ID | Administrator’s unique identifier | int | 5 bytes | Foreign |
| genre | Song genres | str | 256 bytes | N/A |

**Song**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute | Definition | Type | Size | Key |
| songID | Song’s unique identifier | int | 5 bytes | Primary |
| songCatalog\_ID | Song Catalog’s unique identifier | int | 5 bytes | Foreign |
| title | Song's title | str | 256 bytes | N/A |
| artist | Artist of song | str | 256 bytes | N/A |
| duration | Length of song | int | 5 bytes | N/A |

**SponsorModule**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute | Definition | Type | Size | Key |
| sponsor\_ID | Sponsor Module’s unique identifier | int | 5 bytes | Primary |
| admin\_ID | Administrator’s unique identifier | int | 5 bytes | Foreign |
| sponsor\_Name | Name of Sponsor | str | 256 bytes | N/A |
| description | Description of Song | str | 256 bytes | N/A |

**ProgramModule**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute | Definition | Type | Size | Key |
| program\_ID | Program Module’s unique identifier | int | 5 bytes | Primary |
| admin\_ID | Administrator’s unique identifier | int | 5 bytes | Foreign |
| title | Title of Program | str | 256 bytes | N/A |
| description | Description of program | str | 256 bytes | N/A |

**MissionStatement**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute | Definition | Type | Size | Key |
| mission\_ID | Mission’s Statement’s unique identifier | int | 5 bytes | Primary |
| admin\_ID | Administrator’s unique identifier | int | 5 bytes | Foreign |
| description | Description of Worth the Words’s Mission | str | 256 bytes | N/A |

**ApplicationSubmission**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute | Definition | Type | Size | Key |
| application\_ID | Application’s unique identifier | int | 5 bytes | Primary |
| admin\_ID | Administrator’s unique identifier | int | 5 bytes | Foreign |
| user\_ID | User’s unique identifier | int | 5 bytes | Foreign |
| submissionDate | Applicant’s submission date | date | 256 bytes | N/A |
| resume | Applicant’s Resume | str | 256 bytes | N/A |
| message | Applicant’s message | str | 256 bytes | N/A |
| status | Status of application | str | 256 bytes | N/A |

**ContactForm**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute | Definition | Type | Size | Key |
| contactForm\_ID | Contact Form’s unique identifier | int | 5 bytes | Primary |
| userID | User's unique identifier | int | 5 bytes | Foreign |
| subject | Subject of contact message | str | 256 bytes | N/A |
| message | Contact form’s message | str | 256 bytes | N/A |

**ContactSubmissions**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute | Definition | Type | Size | Key |
| contactForm\_ID | Contact Form’s unique identifier | int | 5 bytes | Primary/Foreign |
| admin\_ID | User’s unique identifier | int | 5 bytes | Foreign |

**Donation**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute | Definition | Type | Size | Key |
| donation\_ID | Donation’s unique identifier | int | 5 bytes | Primary |
| user\_ID | User’s unique identifier | int | 5 bytes | N/A |
| method | Method of donation | str | 256 bytes | N/A |
| recurring | Whether the donation recurs | bit | 2 bytes | N/A |
| donationStatus | Status of being received | str | 256 bytes | N/A |

**DonationReceipt**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute | Definition | Type | Size | Key |
| donation\_ID | Donation’s unique identifier | int | 5 bytes | Primary/Foreign |
| admin\_ID | Admin’s unique identifier | int | 5 bytes | N/A |
| donationDate | Date of donation | date | 256 bytes | N/A |
| amount | Amount donated | int | 5 bytes | N/A |

**Review**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute | Definition | Type | Size | Key |
| review\_ID | Review’s Unique identifier | int | 5 bytes | Primary |
| user\_ID | User’s unique identifier | int | 5 bytes | Foreign |
| content | Content of review | str | 256 bytes | N/A |
| author | Author of review | str | 256 bytes | N/A |
| date | Date of review | date | 256 bytes | N/A |
| review\_Score | Company rating | int | 5 bytes | N/A |

**ReviewSubmission**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute | Definition | Type | Size | Key |
| review\_ID | Review unique identifier | int | 5 bytes | Primary/Foreign |
| admin\_ID | Admin’s unique identifier | int | 5 bytes | Foreign |
| display\_Status | Whether review is displayed or not | bit | 2 bytes | N/A |

**Calendar**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute | Definition | Type | Size | Key |
| calendar\_ID | Calendar’s unique identifier | int | 5 bytes | Primary |
| admin\_ID | Admin’s unique identifier | int | 5 bytes | Foreign |
| events | Events on the calendar | str | 256 bytes | N/A |

**Events**

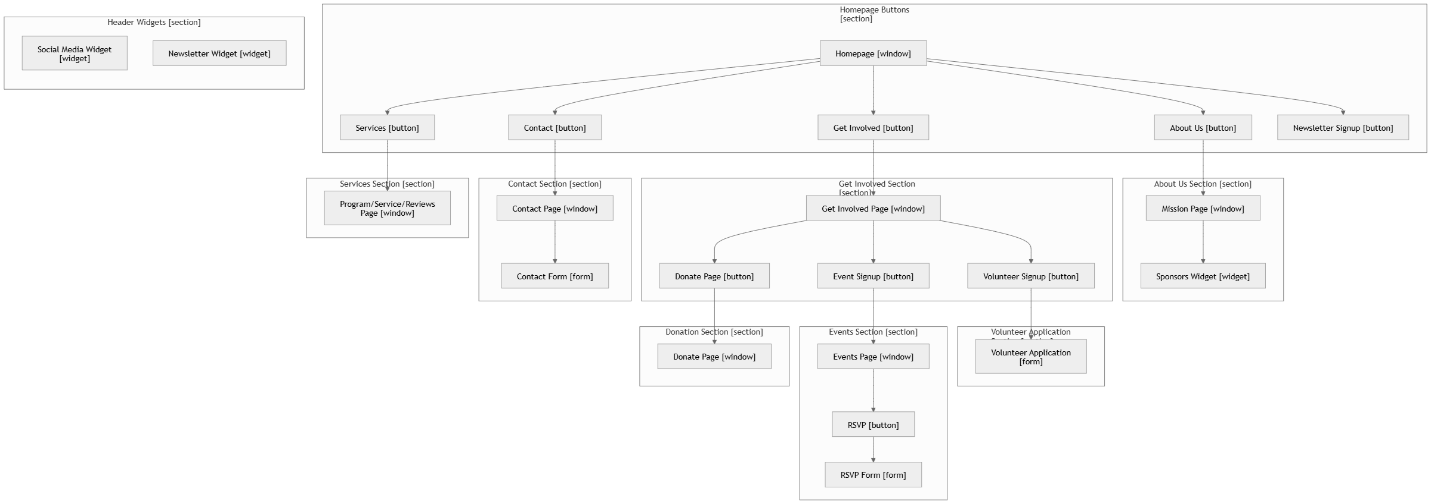
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute | Definition | Type | Size | Key |
| event\_ID | Event unique identifier | int | 5 bytes | Primary |
| calendar\_ID | Calendar’s unique identifier | int | 5 bytes | Foreign |
| rsvp\_count | Number of RSVPs | int | 5 bytes | N/A |
| time | Time of event | time | 5 bytes | N/A |
| date | Date of event | date | 5 bytes | N/A |
| recurring | Recurring status of event | bit | 2 bytes | N/A |

**RSVP**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute | Definition | Type | Size | Key |
| rsvp\_ID | RSVP unique identifier | int | 5 bytes | Primary |
| event\_ID | Event’s unique identifier | int | 5 bytes | Foreign |
| user\_ID | User’s unique identifier | int | 5 bytes | Foreign |
| rsvpDate | Date of RSVP | date | 256 bytes | N/A |
| numberofAttendees | Number of people the RSVP is for | int | 5 bytes | N/A |

# 3. User Interface Diagram

## 3.1 User Interface Navigation Diagram



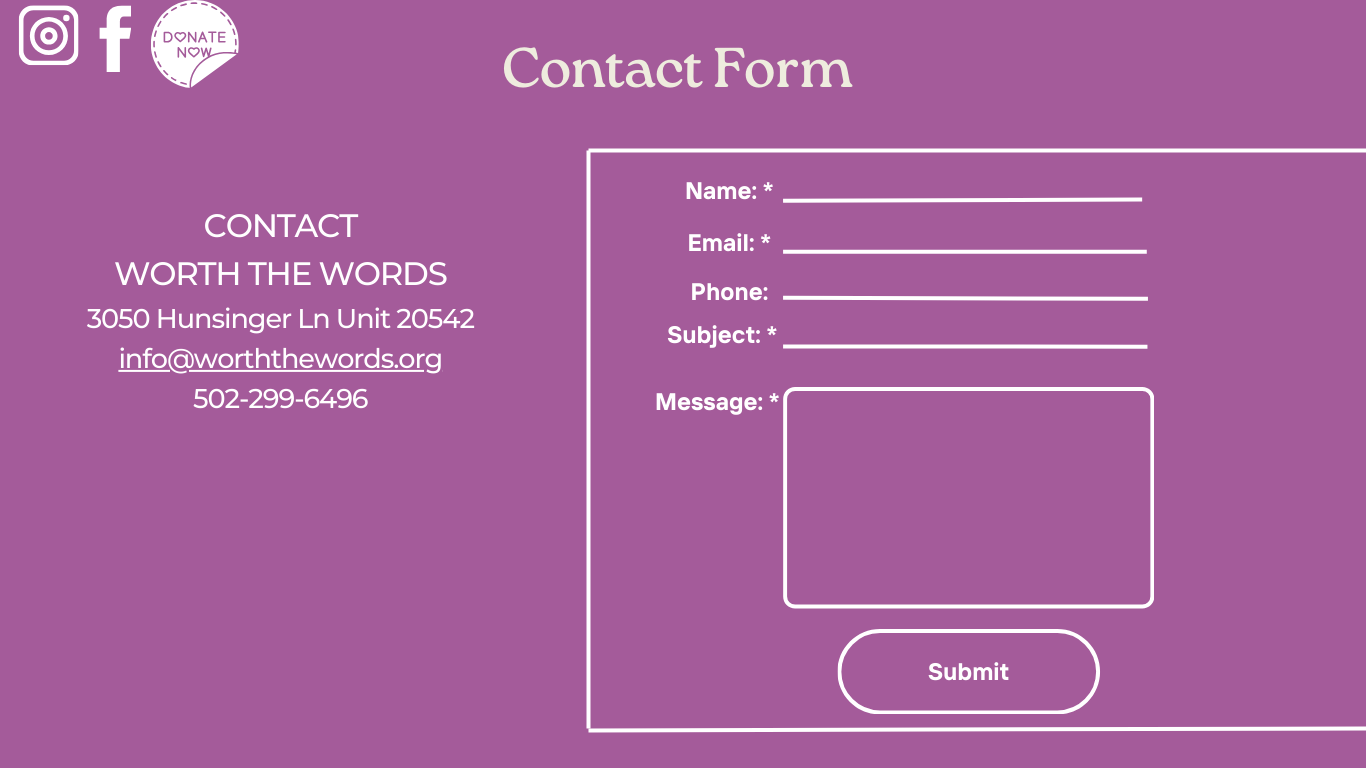
## 3.2 User Interface Diagram Narrative

The "Worth the Words" website is designed with a user-friendly interface, centered around the Homepage, which provides access to the site's core sections through five main buttons: About Us, Get Involved, Services, Contact, and Newsletter Signup. Each page features a consistent header containing a Social Media Widget for engaging with the organization’s platforms and a Newsletter Widget for easy subscription access. The About Us section leads to the Mission Page, which highlights the organization’s purpose and includes a Sponsors Widget to showcase its supporters. The Get Involved section offers users opportunities to engage further by signing up as a volunteer, completing the Volunteer Application, registering for events through the Event Signup button, or contributing via the Donate Page. The Events Page allows users to RSVP for upcoming events with a simple form, while the Contact Page includes a dedicated form for inquiries and feedback. The Services Page provides comprehensive information about the organization’s programs, services, and user reviews. This streamlined design ensures intuitive navigation and encourages user engagement through its consistent structure and accessible features.

## 3.3 Screen Layouts

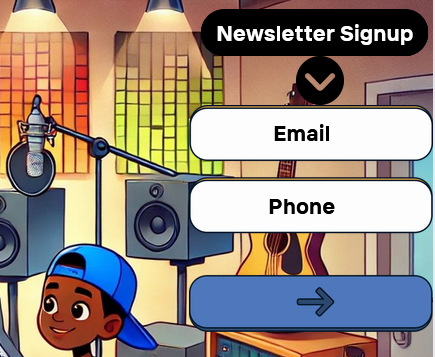
### 3.3A Data Input Screens

#### Contact Form



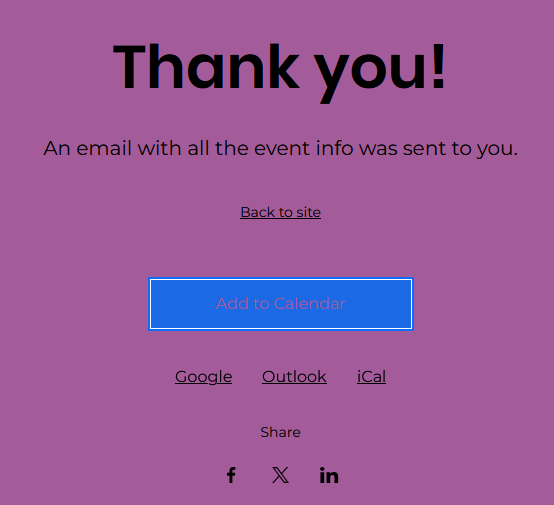
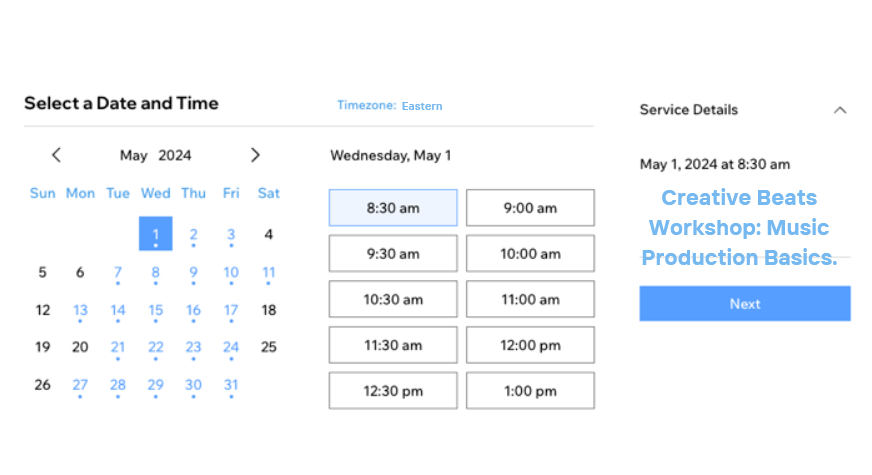
As shown in the prototype, all fields in the contact form will be empty by default. All of the fields will be required for a submission to process. With one exception, the end user may leave the phone number field blank if they’d like. After filling out the fields the submit button sends the information to the backend.

#### Newsletter Form



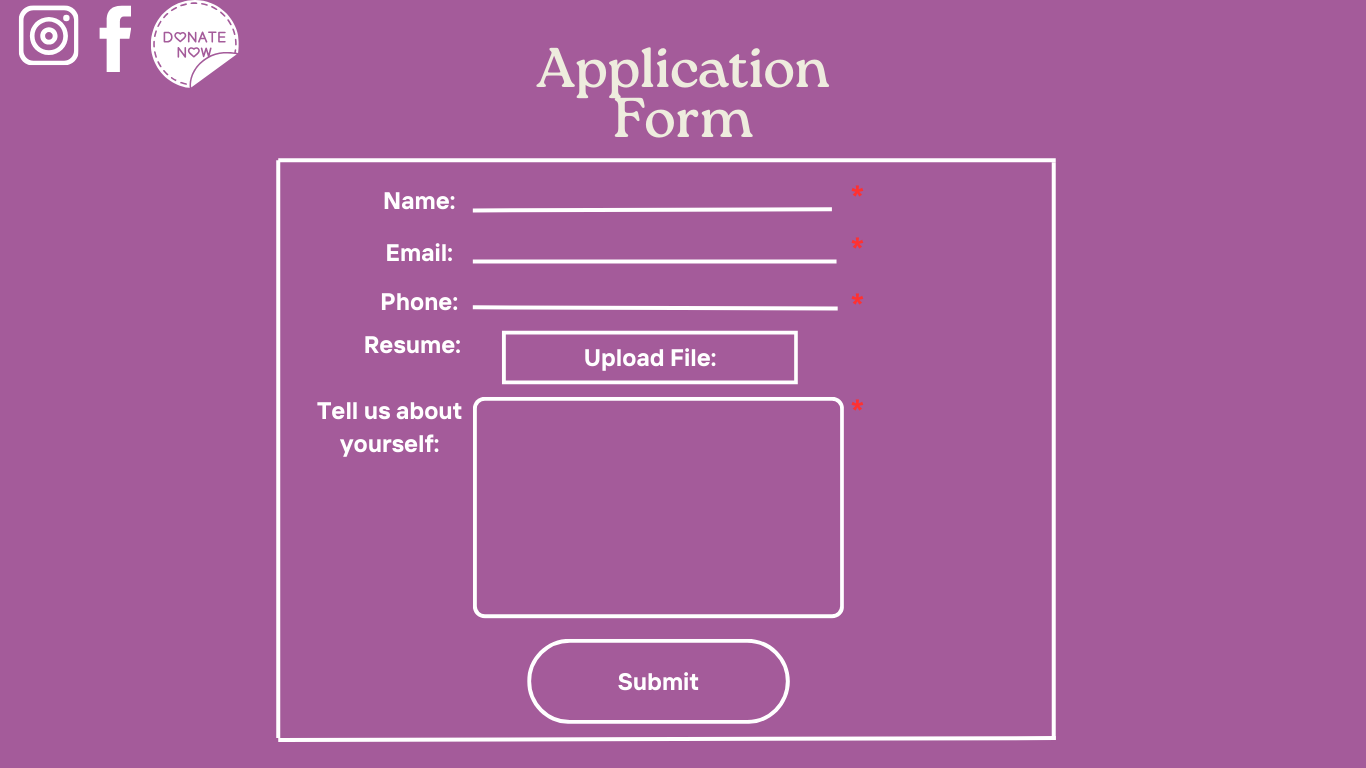
The newsletter form shows up on multiple different pages including the home page and the Stay Connected page. The end user has the option to input either their email or phone number, for the form to submit one of the fields has to be filled. After submission, the data is sent to a Wix dashboard containing all of the Newsletter submissions.

#### RSVP Form



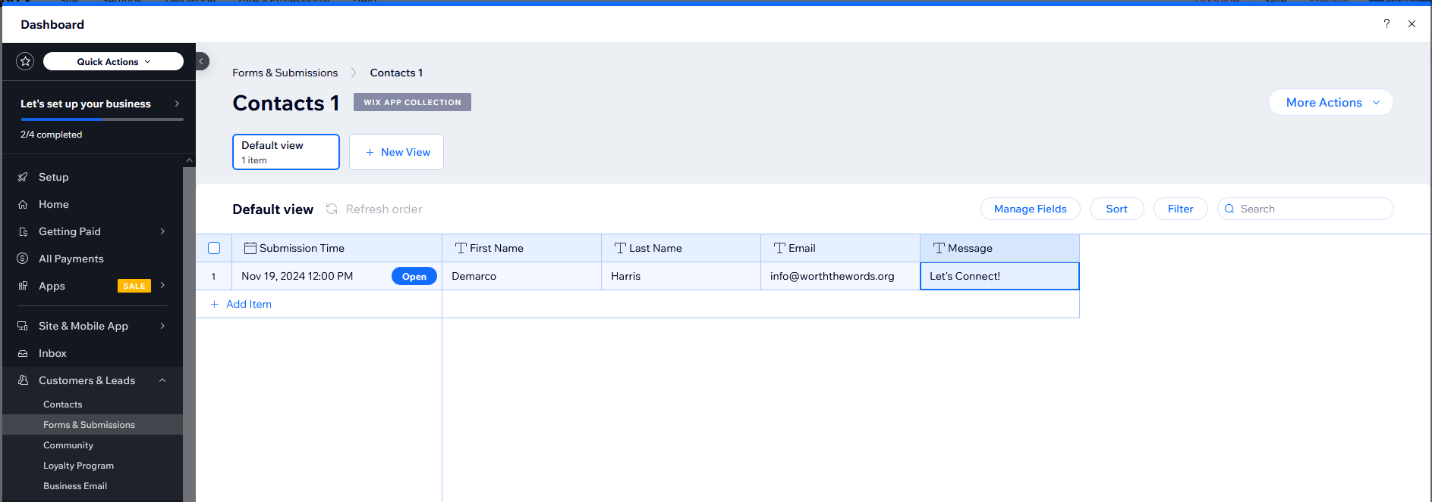
Using the RSVP form, the end user can select the event or service date, the time, view the details of the event/service, and can then click next to confirm their RSVP. After confirming the RSVP, the submission will be sent to a dashboard that contains the full list of guests. After confirming, the end user may also add the event to whichever calendar app of their choice.

#### Application Form

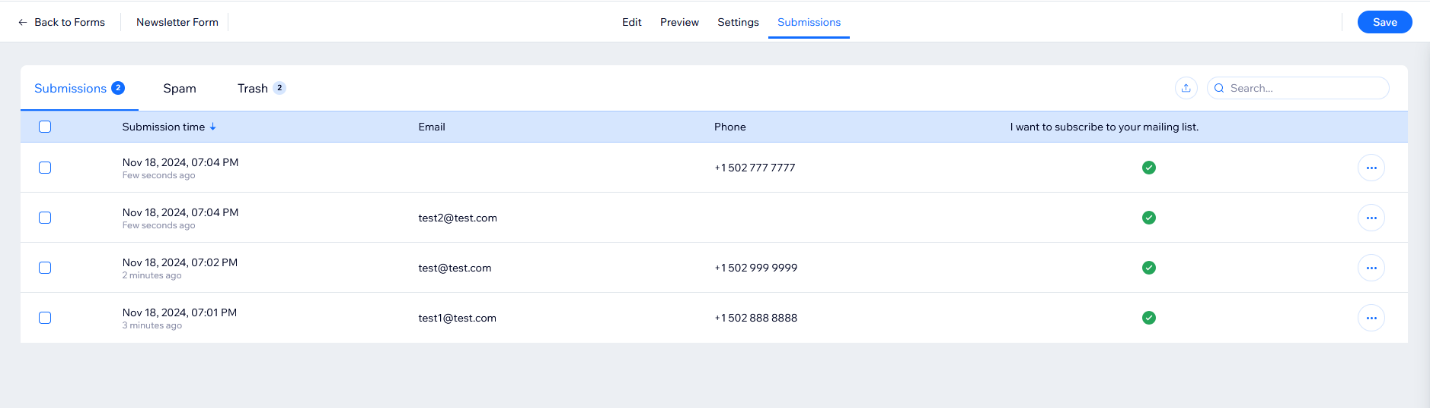
The application form requires the end user to enter their name, email address, phone number, and a custom message about themselves. There’s also a resume field that allows the user to attach a resume to their submission, but that is not a required field. After submission, the user’s information is sent to the Wix backend and is stored in a dashboard containing all of the current application form submissions.

### 3.3B Data Output Screens

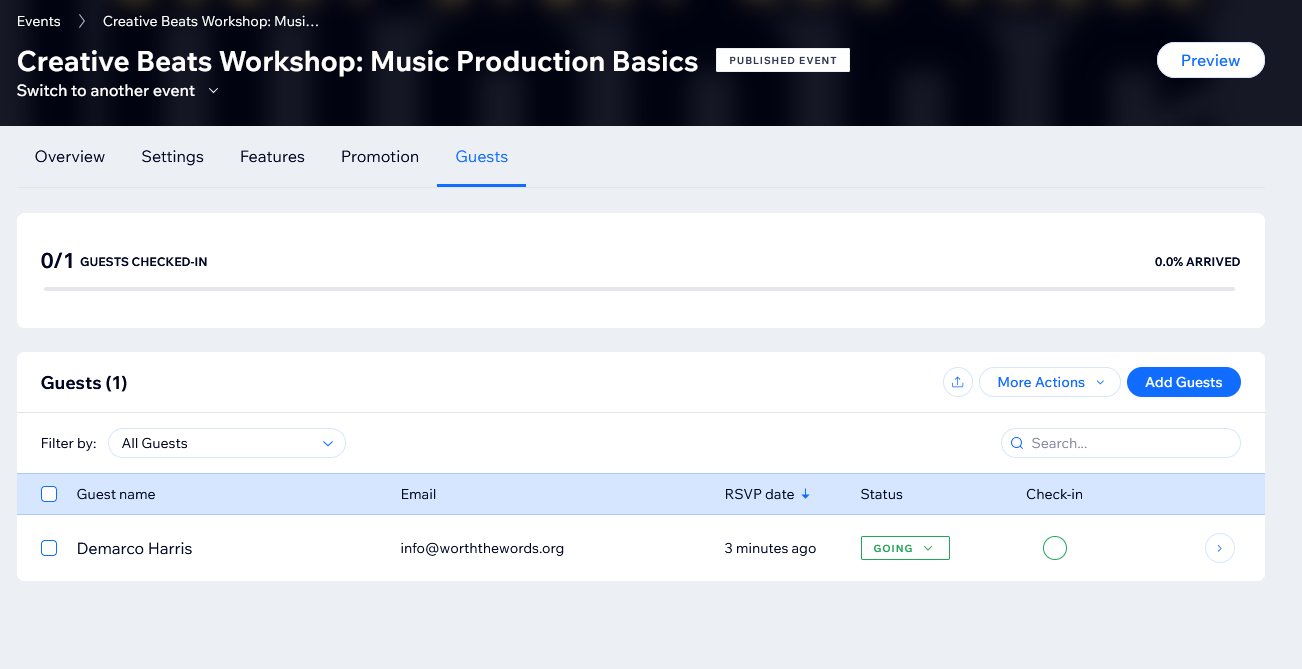
#### Contact/Application Submissions

After an end user submits a contact request, their submitted information is sent to the Wix backend which hosts a dashboard that displays each contact submission. If needed, the administrator can create separate views that showcase different filters or sorted information. The dashboard can also be exported to a CSV if needed.

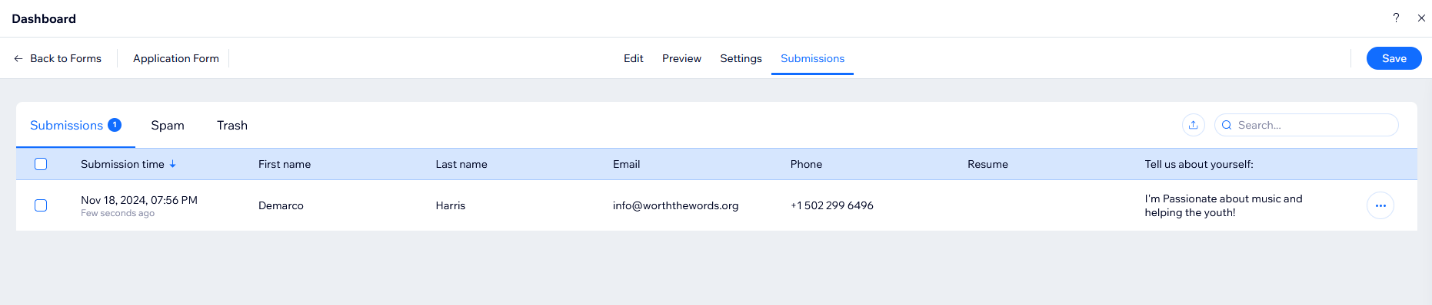
#### Newsletter Submissions

After an end user enters submits either their email address or phone number through the newsletter form, their information is sent to a dashboard in the Wix backend. Each submission’s submission time, email address, and phone number can be viewed from the dashboard. Along with this, there’s a spam filter and trash tab in case the administrator deletes an item and wants to recover it.

#### RSVP Submissions

After the end user submits the RSVP form, their registration is stored in the event dashboard in the guest tab. The registration contains the guests name, email, RSVP date, status, and check-in status. If the number of guests is limited, then the user would not have been able to submit the RSVP form. Each guest can check into the event and the administrator will be notified of the amount of checked in guests through the dashboard. The administrator can also export the guest list or utilize email marketing for the guests.

#### Application Submissions

After an end user submits their volunteer application form, their information is sent to the Wix backend that displays a dashboard. The dashboard contains the application’s submission time, the user’s first name, last name, email, phone number, resume (if submitted), and a custom message from the user. Similar to the newsletter submissions, the dashboard has a spam and trash filter for application submissions.

# 4. Gantt Chart

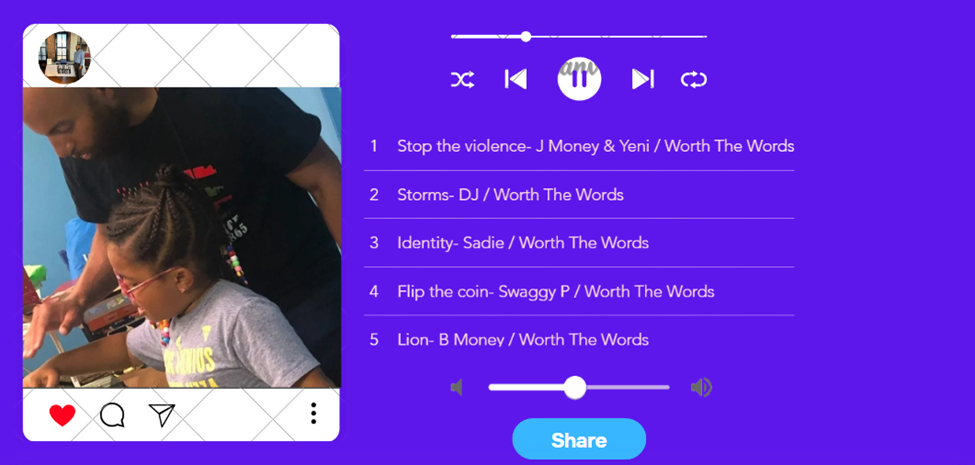
## A screenshot of a graph Description automatically generated4.1 Gantt Chart Diagram

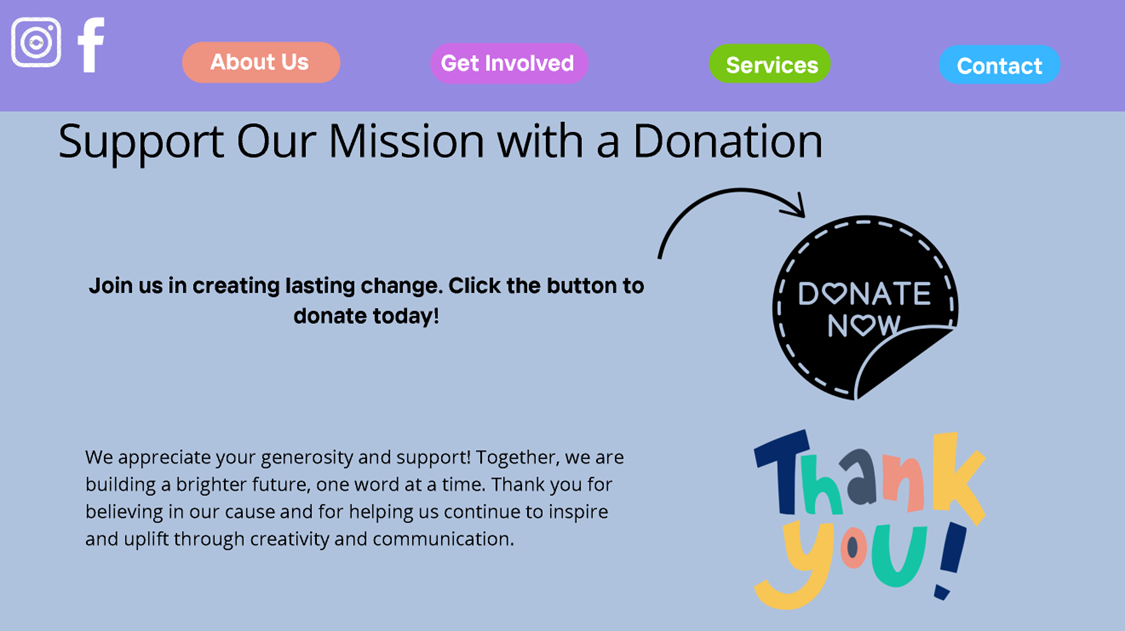
A table with text and numbers

Description automatically generated

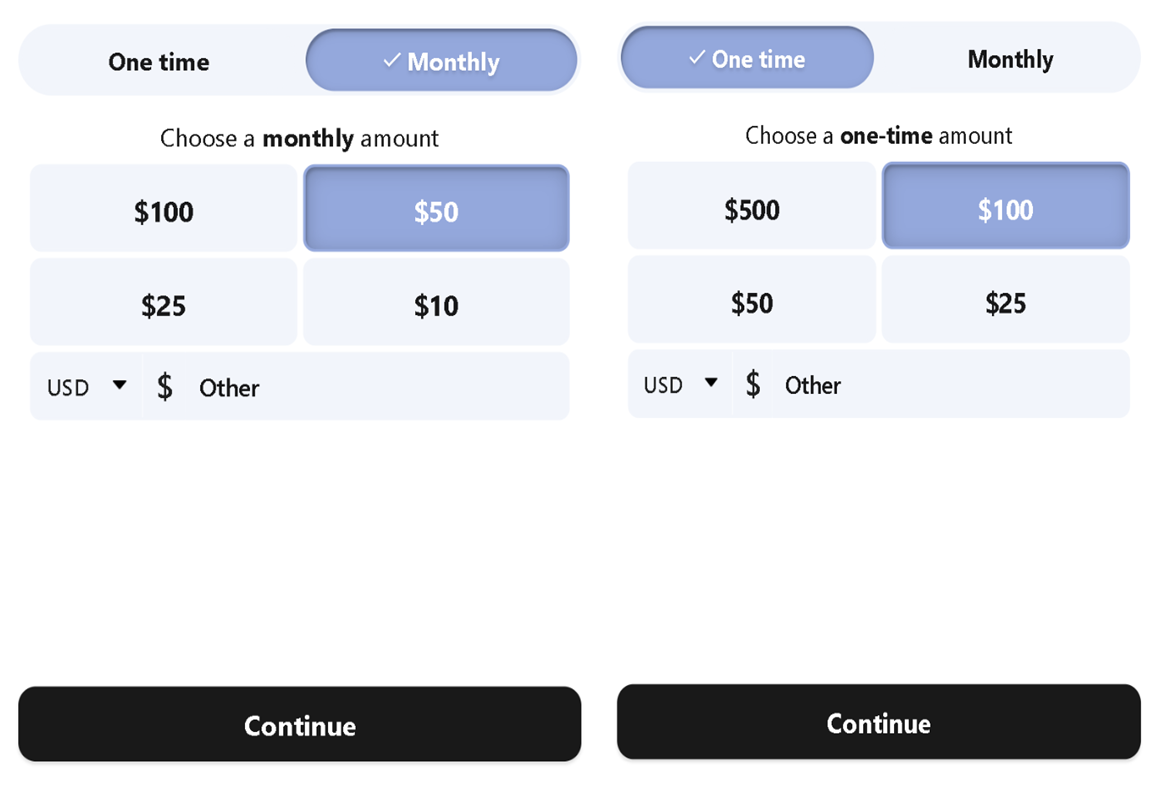
# 5. User Interface Prototypes

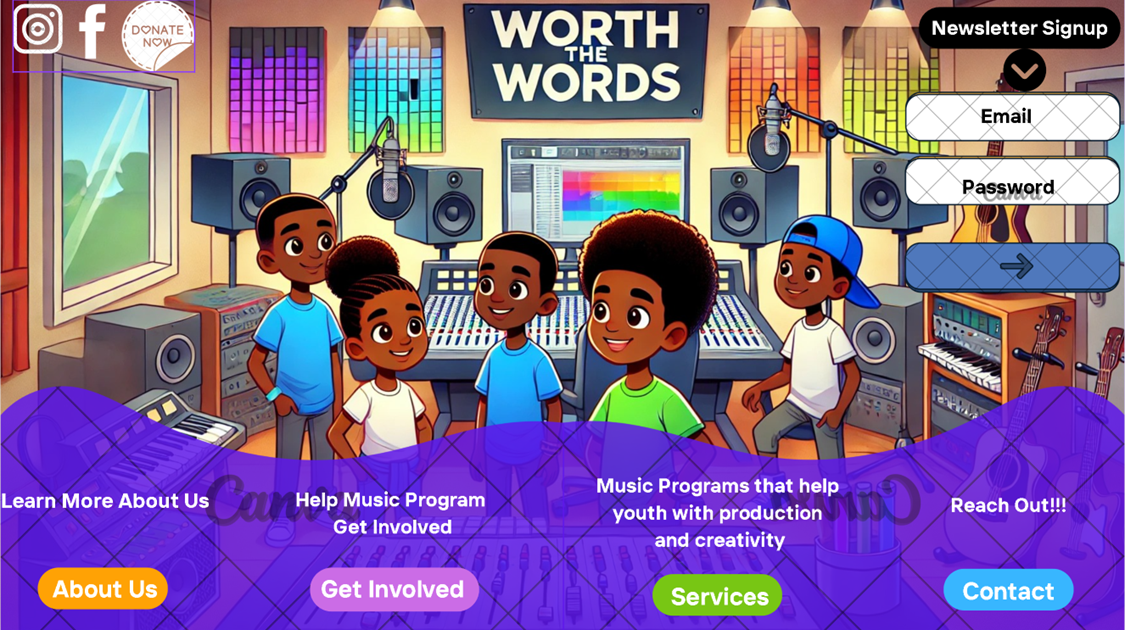
Song Catalog

Donation Screen

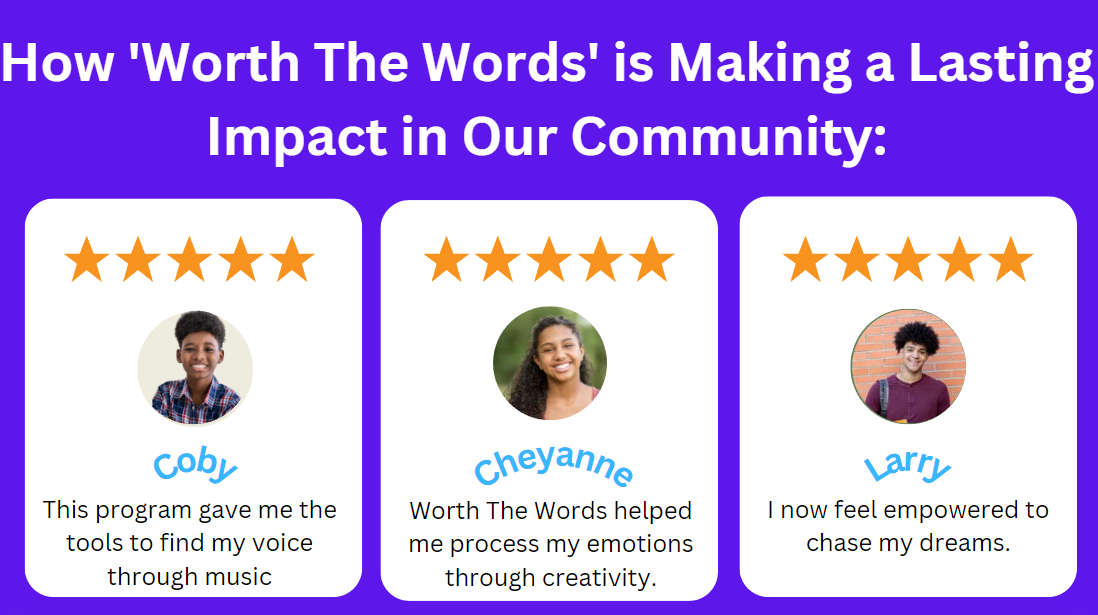


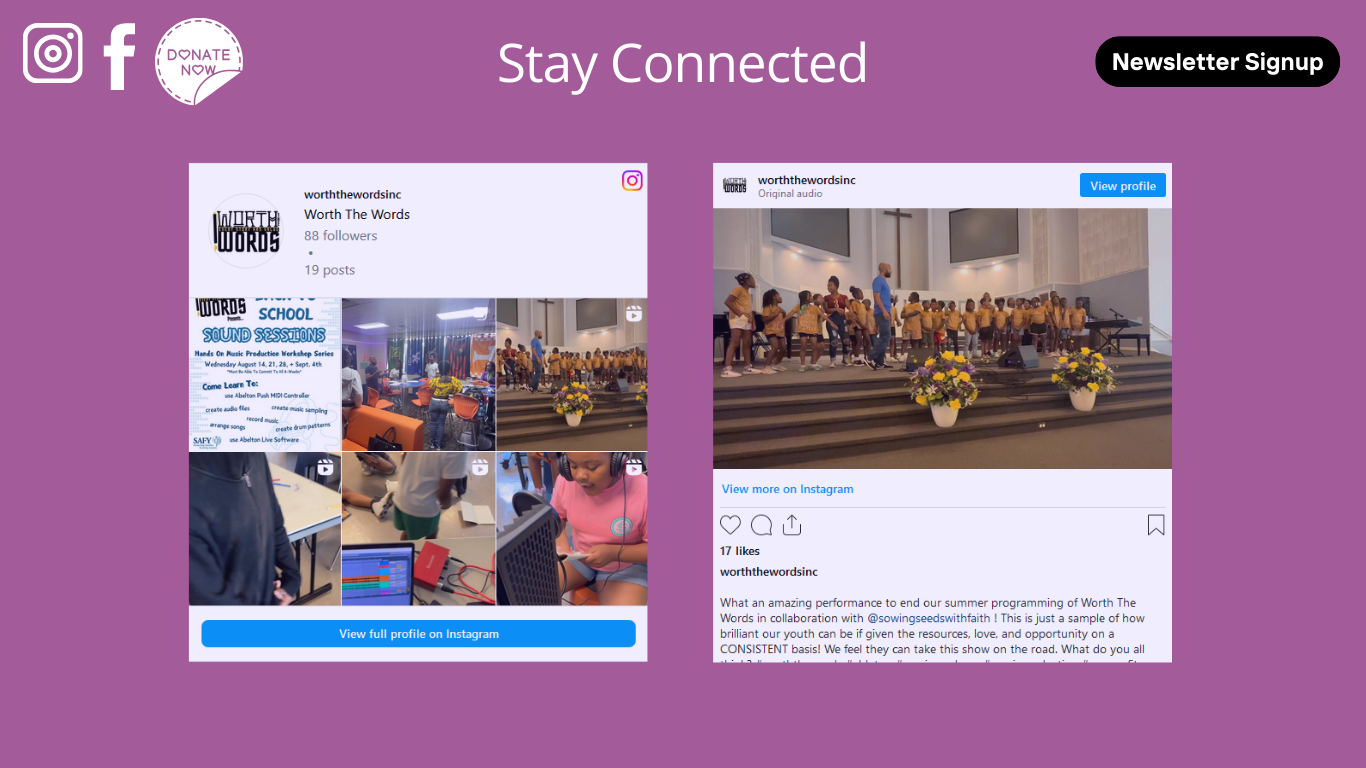
Donation System

Newsletter Signup Dropdown



Reviews Section

Social Media Integration



View Program/Workshop Information A screenshot of a computer

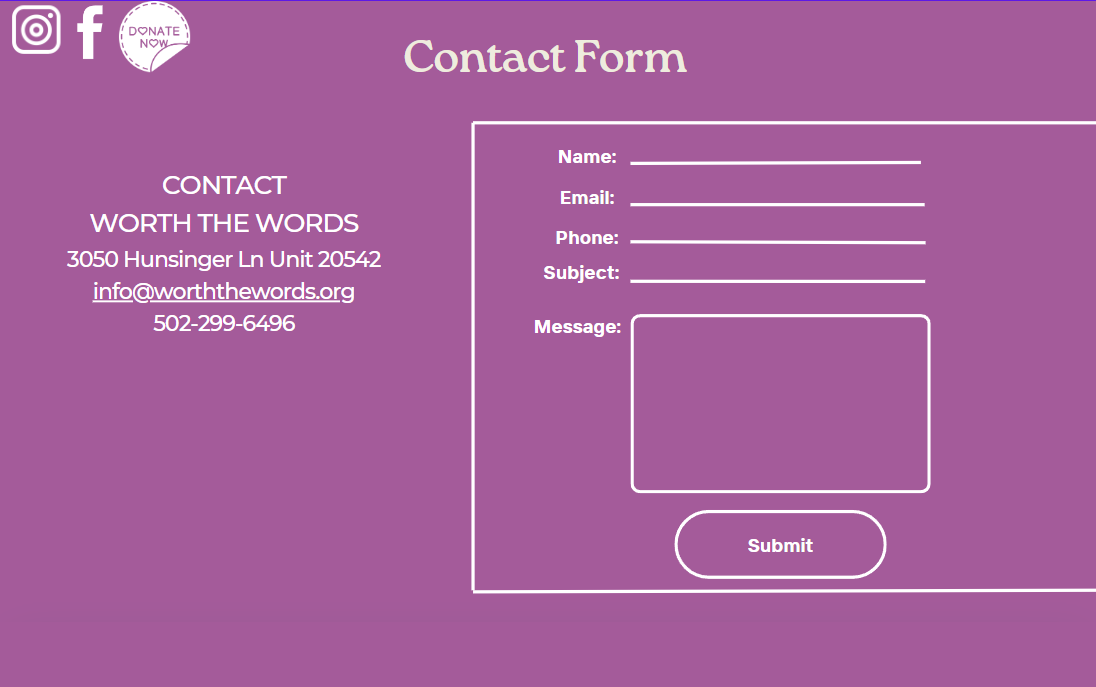
Description automatically generated

Interacts With Calendar

A screenshot of a computer

Description automatically generated

Fill Out Contacts

View Mission Statement  
A group of people sitting at a desk with laptops

Description automatically generated