# A white screen with black text Description automatically generated

# User’s Perspective

When the page loads, it queries the database for events.

The result will be an array of objects containing:

* Title
* Branch
* Content
* Timestamp

Using the filter buttons will change the query as needed. For example, a user could filter for:

* Specific title.
* Specific branch.
* Timestamp range.
* Events not RSVP’d to.

All events will be displayed from most recent.

All events are displayed as a grid, like in the diagram above.

The banner at the top and side navigation bar are consistent through all pages.

Clicking the RSVP button will send a POST request to the event (from its id) and insert the userid into the table. The button will then be disabled, so the same user can’t RSVP to the same event multiple times.

The only thing that can be interacted with for the events is the RSVP button.

# Manager’s Perspective

Will also display the events but has more functionality.

A manager can only add events from a branch they are managing.

There will be a form (text area, etc.) that a manager can input:

* Title
  + Text
* Content
  + Text area
* Branch
  + Dropdown

The timestamp will be automatically calculated.

Once the manager clicks submit, it converts the data to JSON from above and saves it on the database. It will also query the database again to fill the events.

# Routes

The page will be something like /events/userId

Where userId is the id of the user logged in.

The website will know if the user is a manager by querying the database for that role.

This means a user must be logged in to see events.

So, we need to make sure the correct user is logged in. Else, redirect to log in page.

# Database Schema Plan

Each user has an id (which starts at 1 since 0 is the main admin and increments each time a new user log in).

## Users Table

* Primary key
  + UserId
* Secondary keys
  + Email
  + Password (hashed)
  + Username
* Foreign keys
  + RoleId

## Roles Table

* Primary key
  + RoleId
* Secondary keys
  + Value
    - 0 for manager.
    - 1 for admin.
    - A verified user is a user, no need for a role.
* Foreign keys
  + UserId
    - Used for inner joining.

## Events Table

* Primary key
  + EventId
* Secondary keys
  + Title
  + Content
  + Timestamp
* Foreign keys
  + BranchId

## Branches Table

* Primary key
  + BranchId
* Secondary keys
  + Name