Sprint 4 Retrospective
R&M Software Solutions
[student names redacted]

What functionality does the system have at the end of this sprint? List user stories that you successfully implemented during this sprint (1 pt)

Frontend Functionality:

Users are able to go to the events page on the Glorious Recovery website and see all events in a list view. They are also able to toggle to a day or calendar view. They can type a query in the search bar and if that keyword matches *any* of the metadata of the event, it will display as a result. Additionally, users are able to click on any event to see more details such as: title, start/end time, description, hosting organization, link to tickets, link to Facebook page, or google maps link with directions. If the user decides they would like to add the event to their own calendar, they can scroll to the bottom of the page and hit either 'Add to google calendar' or 'Add to iCalendar'.

Backend Functionality

The scraper is able to pull all the relevant data off of every upcoming event off of every organization URL that it reads from. If the event has already been scraped, it checks the event data on [project partner]'s website and compares it to event data on Facebook. If there is a difference in either the title, description, start time, or end time, it pushes those changes to her website.

User stories accomplished in Sprint 4:

As a user I want to filter the events page based on interests, time, organization, and location so that I spend less time looking at the wrong events.

As a scraper I want to be able to insert new events in between previously posted events so that new events are also put in proper chronological order

As a scraper I want to check Facebook for updates so that I can update the wordpress page if there are changes to an event.

As a scraper I want to work with the Events Calendar REST API instead of the wordpress REST API so that I can post events to the website instead of blog posts.

As an admin I want to add and delete URLs from the scraper list so I can change which organizations are being scraped from.

As an admin I want to be able to view analytics for my website so that I can see which events are getting the most clicks.

Did you end up making any changes to any of these user stories? Did you break down further any of the user stories? Did you identify any new user stories during this sprint and, if so, did you add them to the product backlog or decide to implement them right away? Explain (1 pt)

- New story created: *Check for Facebook event updates*
  - We realized that our scraper would skip over events that have already been posted. If an event was updated on Facebook, we wanted to be able to reflect those changes on the Glorious Recovery website as well. That way, if something like a date changed, people wouldn't be showing up at the wrong time.
- New story created: *Use the Events Calendar REST API*
  - When we saw [project partner] had updated her event page, our old version of the scraper wasn't going to work anymore. It was originally working with the Wordpress REST API which is only used to push media and blog posts. The Events Calendar takes event objects as the parameter. Luckily, their API is built on top of the wordpress API with just some minor syntax differences. Now, all scraped events post as Event objects instead of Blog post objects.

What are the "lessons learned" at the end of this sprint? What would you do differently next time? Explain (1 pt)

- Making sure that the majority of the work is done in the first 3 sprints due to time constraints and the 4th sprint not being a full sprint
- Keeping in contact with your client at all times and throughout the whole entire project is very important for updates or changes that need to be implemented
- Having great communication with your team members is the key to a successful project.

As a group we decided that what we would have done differently was to work on the local copy of [project partner]'s website sooner rather than later. If we had used the local copy sooner we would have completed all of our story points.

Are there any user stories left unimplemented in the backlog? Are there any new user stories that you would consider adding to the backlog. List these user stories and explain them.

Unimplemented story:

As a scraper I want to be automated so that I can update the website without admin interaction.

This is the only story we planned to complete that we didn't fully complete. It is in a bit of a gray area. From [project partner]'s point of view, her website automatically updates every single day whenever there are new events added to her chosen organizations.
For us though, it isn't entirely automatic. After a conversation with her, she said she would be thrilled to keep working with us next semester. We agreed that it would be okay for one of us to run the scraper as a cron job once a day off of our own computer until next semester when we can get it running properly on a server. We tried one approach using pythonanywhere but the requests library wouldn't work with Facebook from their IP address. We will deal with running it ourselves until development picks up again next year.

New story to add to backlog:

As a scraper I want to multithreaded so that I can scrape more events in less time.

As the list of organizations gets longer, it would be nice to have the scraper multithreaded so that it doesn't get bogged down when there are a lot more events to pull data from. Multithreading would allow the scraper to pull data all at the same time instead of going one-by-one through each event URL.