

## **Sprint 2 Report**

### **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)**

#### **Backend Implementations:**

After this Sprint, our system is now able to push event posts directly to a locally hosted copy of the Glorious Recovery website. We can now be much more confident that any features we implement on this page will work the same on the actual live website. This is thanks to the Wordpress Rest API which allows us to have a database-less approach to our software. The flow of the event data is now: *Text file containing organization URLs -> Web Scraper -> Wordpress Page.*

When the scraper sends the post to the page, we can specify whether to send it to “drafts” or just immediately publish it to the page. This is a much more elegant solution than what we had before which had relied on updating the page via the client side.

Other than being able to scrape from multiple events at once, the scraper now has functionality built in where it will skip over any event URLs it has already scraped from. This prevents duplicate events from being posted to the website and also has the added benefit of being more efficient.

The final backend implementation we made was allowing the scraper to automatically authenticate with Google Calendar. We were able to store the google credentials inside of an encrypted pickle file. It only has to create this the first time it runs.

#### **Frontend Implementations:**

In terms of frontend implementations, users are now able to see any upcoming events on the events page. They are able to search by a keyword in the sidebar and the matching events are shown. Users are also able to click a “see more” button to read the rest of the details about the event. Each event also has its own corresponding thumbnail. From the dropdown menu on the events page, users can also access an updated version of our Master Calendar. We didn’t explicitly make this a story because we had a calendar in the previous sprint, we just put it in a prettier skeleton.

## **User stories accomplished in Sprint 2:**

*As a user, I want to be able to see upcoming events, so that it is easier to see which ones I can attend.*

*As a user, I want to have a search function for events, so that I can look up events by keyword.*

*As a user, I want to click a “see more” button on each event, so that I can see more details about the event.*

*As a user, I want to see a thumbnail associated with each event, so that the webpage is more visually pleasing.*

*As a scraper, I want to push event posts to a wordpress site, so that they can either be immediately published, or sent to drafts for approval by the customer.*

*As a scraper, I want to iterate through a list of organization links and grab all upcoming events from them, so that I can create posts based off of every new event.*

*As a scraper, I want to avoid posting/reading events I have already seen, so that I am more efficient and no duplicate posts are made to the website.*

*As a scraper, I want to automatically authenticate with Google Calendar, so that the admin does not have to manually re-enter credentials each time the script is run.*

**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)**

- *Story changed:*
  - We broke down the “see upcoming events” story into two parts; seeing the events and putting the events in order. When we got the events to post on the events page, we realized that they are posting in the order that they’re being scraped. This is what ultimately led us to adding another story, “see events in order by date”, which we will implement in sprint 3.
- *New story created:* see a thumbnail associated with each event.
  - Since we’re now working with a copy of the Glorious Recovery page, we want to make sure that the scraper pushes events in the same manner that Aleksa would like to see them, which includes having a thumbnail image of the event.

- *New story created:* push event posts to wordpress site.
  - Since Aleksa's page is on Wordpress, we figured we should make sure that the scraper can push the events to her page. We decided to implement this during this sprint because we had time to do so.
- *New story created:* iterate through list of organization links and grab all upcoming events.
  - Initially, the scraper has been working with one link to grab event information from. After a conversation with Aleksa, we realized that we needed the scraper to pull from multiple links. We decided to implement this during sprint 2 because we discarded a few stories and had time to do so.
- *New story created:* avoid posting/reading events I have already seen.
  - After the project update presentation, it was noted that we may run into issues with duplicate events. We decided to create a story to overcome that issue and implemented this during sprint 2 because we had time to do so.
- *Story deleted:* Have a database.
  - We've shifted gears in how we want to develop our software. The need for a database was eliminated in the process.
- *Story deleted:* Push event data to the database.
  - Because we do not need a database anymore, we also don't need to push to one. It was a post-condition to the previous story.

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

- *Don't over-engineer it: Find the path of least resistance*
  - There were many ways to approach this project and we originally chose the most complicated path. An example of this was when we tried to use DynamoDB on AWS to store our data. We did more research and found a more efficient way to get the job done and ended up not needing AWS at all.
- *The importance of security (especially when hosting data on a server)*
  - Since we're working with different types of credentials, we needed to have another conversation with Prof. [redacted] about how to use and store them securely. An example of this was putting them into a password protected file that only a root user can access and then after development, deleting

the root user from the server. This would prevent any bad actors from getting access to the sensitive data.

- *Having a good sense of time constraints*
  - We had a big shift in our project trajectory which led to the deletion and creation of different stories. We had to be able to gauge our time appropriately in order to adapt to the change.
- *Relying on and giving your support to your team is important for success*
  - As the end of the semester approaches it's easy to get overwhelmed and stressed so when someone needs a break communicating with your group is important for the project to continue.

**Provide an updated numbered list of all user stories yet to be implemented; indicate pre- and post-conditions (1 pt)**

ID	As a/an	I want to...	so that...	Notes	Priority	Size
1	User	see upcoming events in order	i can see what events come first		high	3
2	User	filter the events page based on interests, time, organization, and location	I spend less time looking at the wrong events	<i>Precondition: Story 1</i>	Medium	5
3	User	see if I can purchase tickets to the event	I can buy them if they're required	To be shown on the "see more details" *client requested*	medium	8
4	Admin	Add and delete URL's from the scraper list	So I don't have to re-upload the list	Most likely will use google forms	High	3
5	Admin	view analytics related to the calendar and events	I can have the data		Low	3
6	Admin	have the ability to delete events	I can remove potentially inappropriate events		High	2
7	Scraper	run on a server	So I can be accessed/run anywhere	<i>Precondition: Server needs to be created</i>	Very High	3
8	Scraper	be automated	update the webpage without admin interaction	cron job	High	5
<b>SUM</b>						32