Taskforce

In-house Project Management Software

Sprint Three

Version 1.0

Written by:
Brian Nge Jing Hong
Muhammad Ibrahim bin Mohd Yusni
Chua Xian Loong
Lucas Wee
Koe Rui En
Diana Wijaya

Taskforce

Contents

1. Sprint Planning

1.1 Important Links

Link to Project Inception: Project Inception V2.0

Link to Trello board:
https://trello.com/invite/b/XmURix1g/ATTIcca68202ded99301ef54aa2fd5dc1e114A503979/project-1

Link to the meeting minutes: Meeting Minutes 9 - 5/10/2023

Link to Refined User Stories: User Stories

Link to Task Allocation: Task Allocations

1.2 Product Backlog

The product backlog consists of all the user stories that the team has decided for the project. The product backlog is logged on Trello. The link to the Trello board is provided in section 1.1. To decide on the user stories as well as its story points, a meeting was held on 5 October 2023, Thursday from 12:30 PM to 01:40 PM. The meeting minutes for the meeting are accessible through the link provided in section 1.1. It is confirmed that the product backlog follows the DEEP criteria. Additionally, the user stories are refined according to the INVEST criteria before placing it into Trello. The refined user stories can also be accessible through the spreadsheet link "Refined User Stories" in section 1.1.

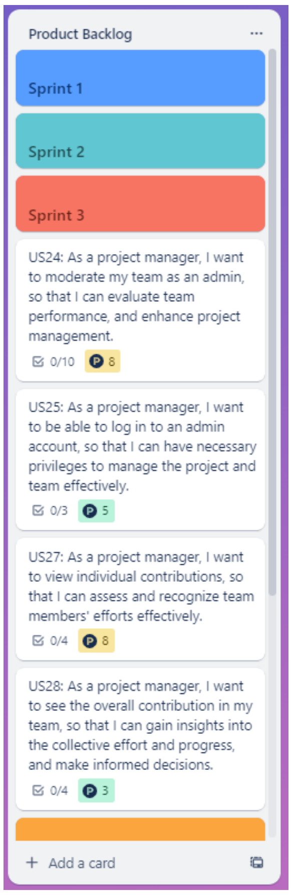
Several spike stories are also added in Trello, for team members who are not familiar with the programming languages to work on their skill sets while working on the project. Each spike story includes a link to a youtube video or a website (online resources and courses) that is sufficient for team members who are not familiar with the programming language to gain necessary knowledge for the project. Team members who were not familiar with any programming languages used in the project, or aspects needed to complete the project, assigned themselves with a task, and also logged their time spent during the learning process.

Access to essential links such as the UI/UX design decisions and the web application prototype are also attached in trello, allowing convenient access to the important links for team members.

Some documentation examples for the product backlog and spike stories are attached below.

Taskforce

Product backlog:



Product Backlog

Sprint 1

Sprint 2

Sprint 3

US24: As a project manager, I want to moderate my team as an admin, so that I can evaluate team performance, and enhance project management.
0/10 8

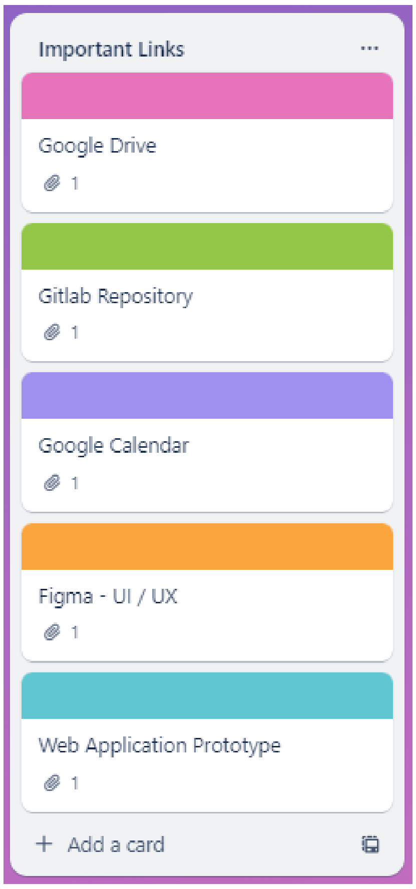
US25: As a project manager, I want to be able to log in to an admin account, so that I can have necessary privileges to manage the project and team effectively.
0/3 5

US27: As a project manager, I want to view individual contributions, so that I can assess and recognize team members' efforts effectively.
0/4 8

US28: As a project manager, I want to see the overall contribution in my team, so that I can gain insights into the collective effort and progress, and make informed decisions.
0/4 3

+ Add a card

Essential links:



Important Links

Google Drive
1

Gitlab Repository
1

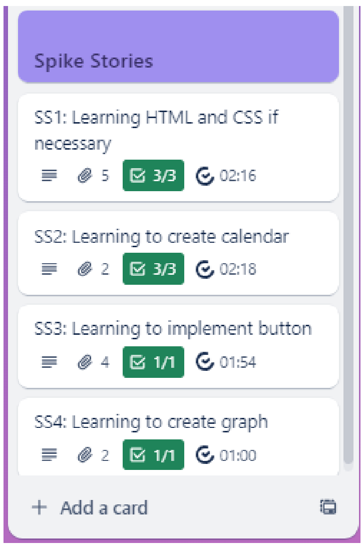
Google Calendar
1

Figma - UI / UX
1

Web Application Prototype
1

+ Add a card

Spike stories:



Spike Stories

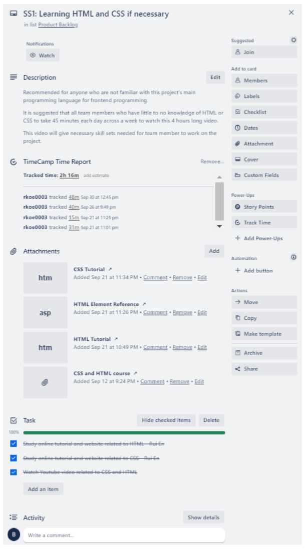
SS1: Learning HTML and CSS if necessary
5 3/3 02:16

SS2: Learning to create calendar
2 3/3 02:18

SS3: Learning to implement button
4 1/1 01:54

SS4: Learning to create graph
2 1/1 01:00

+ Add a card



SS1: Learning HTML and CSS if necessary
in list Product Backlog

Notifications
Watch

Description
Edit

Recommended for anyone who are not familiar with this project's main programming language for frontend programming.

It is suggested that all team members who have little to no knowledge of HTML or CSS to take 45 minutes each day across a week to watch this 4 hours long video.

This video will give necessary skill sets needed for team member to work on the project.

TimeCamp Time Report
Remove...

Tracked time: 2h 16m add estimate

rkoe0003 tracked 48m Sep 30 at 12:45 pm
rkoe0003 tracked 40m Sep 26 at 9:49 pm
rkoe0003 tracked 15m Sep 21 at 11:25 pm
rkoe0003 tracked 31m Sep 21 at 11:01 pm

Attachments
Add

htm CSS Tutorial
Added Sep 21 at 11:34 PM • Comment • Remove • Edit

asp HTML Element Reference
Added Sep 21 at 11:26 PM • Comment • Remove • Edit

htm HTML Tutorial
Added Sep 21 at 10:49 PM • Comment • Remove • Edit

CSS and HTML course
Added Sep 12 at 9:24 PM • Comment • Remove • Edit

Task
Hide checked items Delete
100%

- [x] ~~Study online tutorial and website related to HTML - Rui En~~
- [x] ~~Study online tutorial and website related to CSS - Rui En~~
- [x] ~~Watch Youtube video related to CSS and HTML~~

Add an item

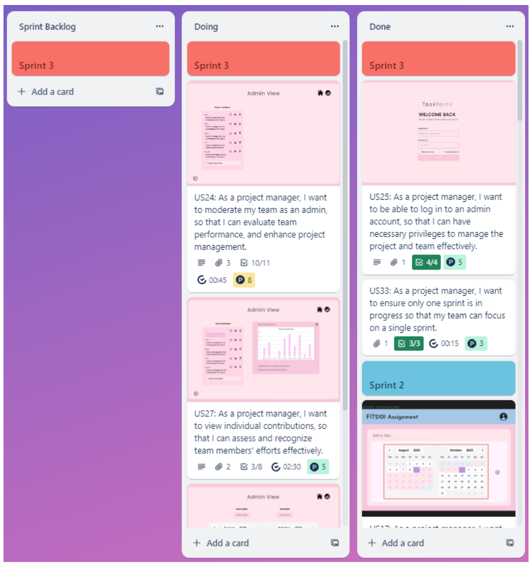
Activity
Show details

Write a comment...

1.3 Sprint backlog

The sprint backlog is logged into Trello, provided link is in section 1.1. All the user stories for the third sprint are decided during the first meeting after the third sprint started, as shown in meeting minute 9, linked in section 1.1. The story point for each user story was decided through planning poker, where each team member will decide on the story point individually, then discussions were raised when there were wide spreads of story point choices, ultimately resulting in one agreed story point per user story. The User Acceptance Criteria is included under each individual task, showing the expectations of each user story with clear definition and standard. The User Acceptance Criterias were later used for testing. The story point is clearly indicated under each user story in the sprint backlog. We have also discussed the task allocation and each team member has assigned themselves to their respective tasks. The link to the summary of task allocations is attached in section 1.1.

Sprint Backlog:



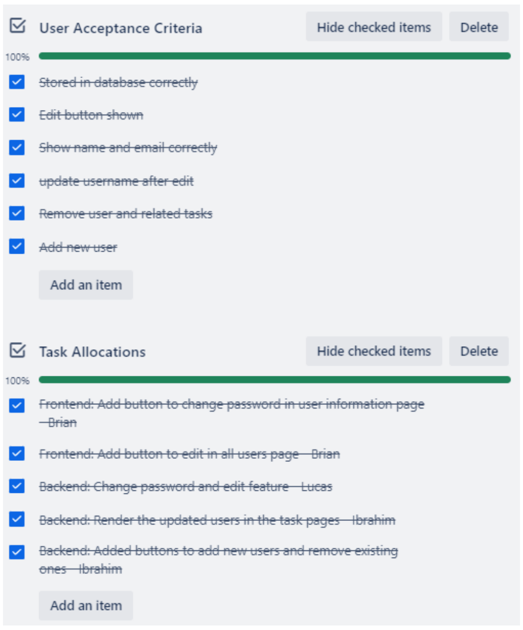
US24: As a project manager, I want to moderate my team as an admin, so that I can evaluate team performance, and enhance project management.
3 10/11 00:45 8

US27: As a project manager, I want to view individual contributions, so that I can assess and recognize team members' efforts effectively.
2 3/8 02:30 5

US25: As a project manager, I want to be able to log in to an admin account, so that I can have necessary privileges to manage the project and team effectively.
1 4/4 5

US33: As a project manager, I want to ensure only one sprint is in progress so that my team can focus on a single sprint.
1 3/3 00:15 3

User Acceptance Criteria and Tasks:



User Acceptance Criteria — Hide checked items — Delete

100%

- [x] ~~Stored in database correctly~~
- [x] ~~Edit button shown~~
- [x] ~~Show name and email correctly~~
- [x] ~~update username after edit~~
- [x] ~~Remove user and related tasks~~
- [x] ~~Add new user~~

Add an item

Task Allocations — Hide checked items — Delete

100%

- [x] ~~Frontend: Add button to change password in user information page - Brian~~
- [x] ~~Frontend: Add button to edit in all users page - Brian~~
- [x] ~~Backend: Change password and edit feature - Lucas~~
- [x] ~~Backend: Render the updated users in the task pages - Ibrahim~~
- [x] ~~Backend: Added buttons to add new users and remove existing ones - Ibrahim~~

Add an item

1.4 Sprint Goal

The sprint goal is to implement the admin view dashboard, admin privileges and display the team members' individual and overall contribution.

This sprint goal was decided during the meeting held on 5 October 2023, Thursday from 12:30 PM to 01:40 PM. To achieve this sprint goal, we have come up with extensive planning and task allocation throughout the team. The details of the task allocation can be found on page 3 of the meeting minutes for 5 October 2023 or in the Task Allocation document linked in section 1.1.

1.5 Project Inception

As linked in section 1.1, the project inception was updated accordingly, where the details for testing are included and updated. Besides, the tester in the team was self assigned and would be carrying out the testing task. The necessary changes for testing were then implemented and the link to the testing document is included in section 3.

2. Daily Scrum

2.1 Important Links

Link to Trello board:
https://trello.com/invite/b/XmURix1g/ATTIcca68202ded99301ef54aa2fd5dc1e114A503979/project-1

Link to the meeting minutes: Meeting Minutes 10 - 8/10/2023 Meeting Minutes 11 - 10/10/2023 Meeting Minutes 12 - 12/10/2023 Meeting Minutes 13 - 14/10/2023 Meeting Minutes 14 - 15/10/2023

Link to the Web Application Prototype: https://taskforce.pages.dev/login
Credentials to Login:
- Email: admin@gmail.com
- Password: 123456

Link to GitLab repository:
https://git.infotech.monash.edu/FIT2101-S2-2023/MA_Thursday11am_Team1/project

2.2 Weekly Stand-Up Meeting

During the third sprint, the team conducted 5 standup meetings in total, on the 8th, 10th, 12th, 14th and 15th of October. The meeting minutes for these meetings are attached in section 2.1.

During the standup meetings, each team member spoke on the following points:
- What has been accomplished since the last meeting?
- What will be done before the next meeting?
- What obstacles are in the way?

The team had the first stand-up meeting on 8 October 2023, from 09:00 PM to 09:10 PM. The link to this meeting minutes can be accessible via the link in section 2.1. With this, each team member is able to understand what tasks each team member is working on. During this meeting, only one team member reported that she is facing an obstacle as the code for the admin view was not in a separate HTML file. Instead it was combined with the user's information page, making it difficult to implement the user's information page properly. However, other team members reported that there were no obstacles faced currently.

On 10 October 2023, the group held the second standup meeting, where each member updated their progress, what will be done before the next meeting, and the obstacles along the way. During this meeting, all team members have reported that they are currently facing no obstacles.

On 12 October 2023, the group held the third standup meeting, where each member updated their progress, what will be done before the next meeting, and the obstacles along the way. During this meeting, one of the team members has reported that she is facing a configuration issue in her gitlab. This prevented her from pushing or pulling the code from the firebase branch and delayed her progress for a short period of time.

On 14 October 2023, the group held the fourth standup meeting, where each member updated their progress, what will be done before the next meeting, and the obstacles along the way. During this meeting, all team members have reported that they are currently facing no obstacles.

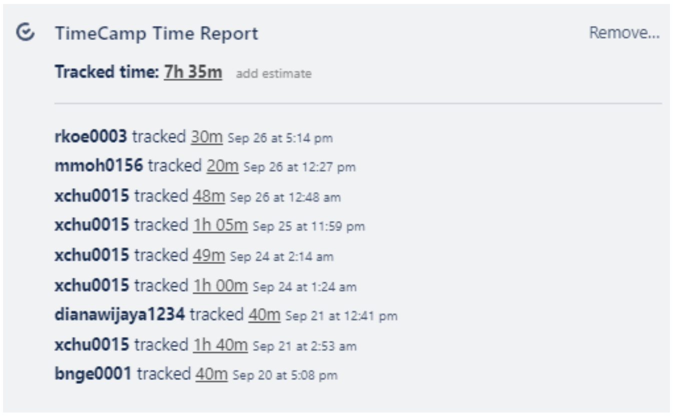
We had the last standup meeting for the sprint on the 15th of October, the meeting minute for this meeting is linked in section 2.1. During this meeting, everyone updated on their progress of work and the team made sure that all the tasks were completed, and the software was ready to be deployed.

Throughout the sprint, we have utilised group chats to keep each other updated on current task status. Besides, we have also iterated the importance of the git policy that we have outlined in your project plan and made sure to only push to the “Firebase” branch and not the main branch.This is to prevent confusion of the code and ensure that everyone is able to work on their end without much merge conflicts. It is also a good industry practice to work on a branch rather than on the main repository.

Throughout the standup meetings, the team was able to complete the following user stories. Further details on the user stories can be found in the trello link provided in section 2.1.

1. US24: As a project manager, I want to moderate my team as an admin, so that I can evaluate team performance, and enhance project management.
2. US25: As a project manager, I want to be able to log in to an admin account, so that I can have necessary privileges to manage the project and team effectively.
3. US27: As a project manager, I want to view individual contributions, so that I can assess and recognize team members' efforts effectively.
4. US28: As a project manager, I want to see the overall contribution in my team, so that I can gain insights into the collective effort and progress, and make informed decisions.
5. US33: As a project manager, I want to ensure only one sprint is in progress so that my team can focus on a single sprint.
6. US34: As a client, I want to view the documentation so that it is easier to track the team's progress.

The time spent by each team member on each task is also logged and kept track of via a Trello power up used, which is known as “TimeCamp”. A screenshot is attached below showing an example of the time tracking feature.



TimeCamp Time Report Remove...

Tracked time: 7h 35m add estimate

rkoe0003 tracked 30m Sep 26 at 5:14 pm
mmoh0156 tracked 20m Sep 26 at 12:27 pm
xchu0015 tracked 48m Sep 26 at 12:48 am
xchu0015 tracked 1h 05m Sep 25 at 11:59 pm
xchu0015 tracked 49m Sep 24 at 2:14 am
xchu0015 tracked 1h 00m Sep 24 at 1:24 am
dianawijaya1234 tracked 40m Sep 21 at 12:41 pm
xchu0015 tracked 1h 40m Sep 21 at 2:53 am
bnge0001 tracked 40m Sep 20 at 5:08 pm

This is the time spent for each team member and individual time allocated on each user story/spike story:

SS4: Learning to create graph

1. Brian Nge Jing Hong - 1h

Total: 1h

US24: As a project manager, I want to moderate my team as an admin, so that I can evaluate team performance, and enhance project management.

2. Muhammad Ibrahim bin Mohd Yusni - 45m
3. Lucas Wee - 30m

Total: 1h 05m

US25: As a project manager, I want to be able to log in to an admin account, so that I can have necessary privileges to manage the project and team effectively.

1. Lucas Wee - 1h 30m

Total: 1h 30m

US27: As a project manager, I want to view individual contributions, so that I can assess and recognize team members' efforts effectively.

1. Brian Nge Jing Hong - 1h
2. Muhammad Ibrahim bin Mohd Yusni - 2h
3. Lucas Wee - 4h
4. Diana Wijaya - 2h 30m

Total: 9h 30m

US28: As a project manager, I want to see the overall contribution in my team, so that I can gain insights into the collective effort and progress, and make informed decisions.

1. Brian Nge Jing Hong - 30m
2. Muhammad Ibrahim bin Mohd Yusni - 2h 20m
3. Diana Wijaya - 2h 10m
4. Chua Xian Loong - 6h 52m

Total: 11h 52m

US33: As a project manager, I want to ensure only one sprint is in progress so that my team can focus on a single sprint.

1. Muhammad Ibrahim bin Mohd Yusni - 15m

Total: 15m

US34: As a client, I want to view the documentation so that it is easier to track the team's progress.

1. Brian Nge Jing Hong - 3h
2. Koe Rui En - 8h 48m
3. Diana Wijaya - 45m

Total: 6h

Summary:

1. Brian Nge Jing Hong - 5h 30m
2. Muhammad Ibrahim bin Mohd Yusni - 5h 20m
3. Chua Xian Loong - 6h 52m
4. Lucas Wee - 5h 50m
5. Koe Rui En - 8h 48m
6. Diana Wijaya - 5h 25m

3. Sprint Review

3.1 Important Links

Link to Trello board:
https://trello.com/invite/b/XmURix1g/ATTIcca68202ded99301ef54aa2fd5dc1e114A503979/project-1

Link to the meeting minutes: Meeting Minutes 15 - 15/10//2023

Link to the Web Application Prototype: https://taskforce.pages.dev/login
Credentials to Login:
- Email: admin@gmail.com
- Password: 123456

Link to the Demo Video: https://youtu.be/rYWXWCqLBMM

Link to Full Live and Updated Risk Register: Risk Register Updates - Sprint 3

Link to GitLab repository:
https://git.infotech.monash.edu/FIT2101-S2-2023/MA_Thursday11am_Team1/project

Link to the testing report: Testing Report

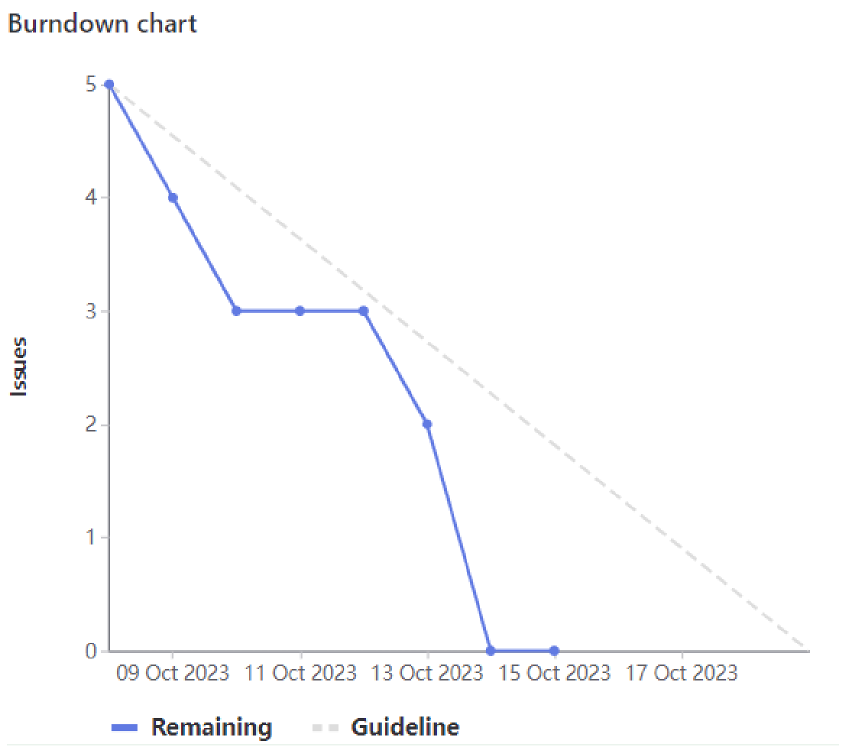
3.2 Burndown Chart

A burndown chart is a visual tool used in project management, specifically in Agile and Scrum methodologies, to track the progress of work over time. It helps teams and stakeholders monitor how tasks or user stories are being completed and whether the project is on track to meet its goals and deadlines. The team has utilised GitLab for its burndown chart feature. The user stories from Trello were placed into GitLab and issues were closed when we implemented that feature in the web application. The burndown chart and burnup chart can be found here:

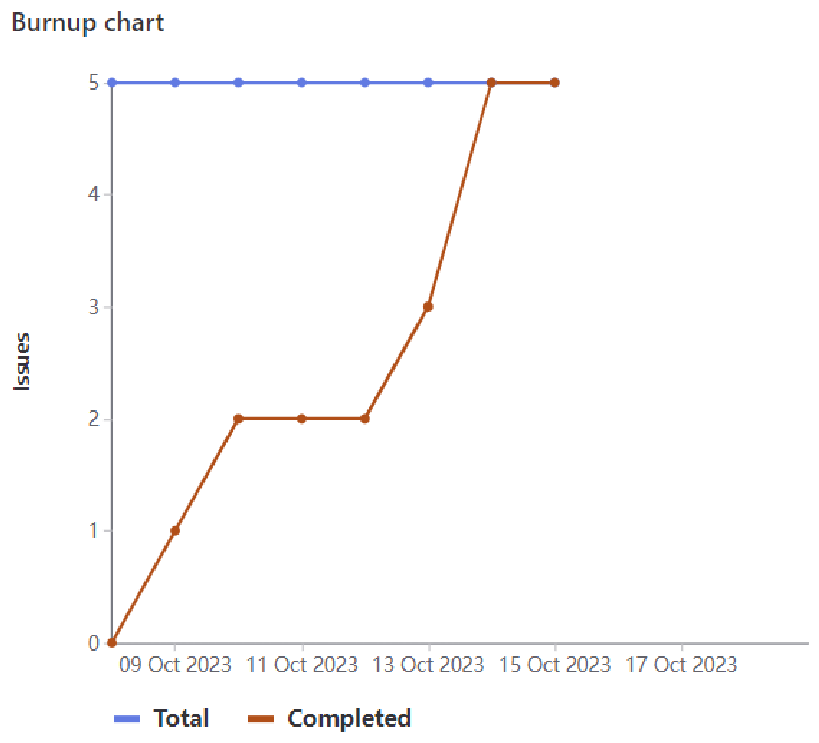
Link to GitLab Milestone:
https://git.infotech.monash.edu/FIT2101-S2-2023/MA_Thursday11am_Team1/project/-/milestones/4#tab-issues

An overview of the burndown chart and burnup chart by the end of the sprint is shown below. As shown in the charts, all the tasks are successfully completed on time with no issue.

Burndown Chart:



Burnup Chart:



3.3 Risk Register

In Scrum, a risk register is a document or tool used to identify, assess, and manage risks that may impact the successful delivery of a project or a sprint. It is a proactive approach to risk management and it is beneficial in a project management setting. The team has updated the risk register accordingly. A full overview of the live and updated risk register, along with the existing risk register is included in a link provided in section 3.1. A full overview of the live and updated risk register, along with the existing risk register is included in a link provided in section 3.1. The new risks discovered during this sprint is shown in table 3.3.1. There were multiple risks in the risk register that was re-visited during this sprint. The live and updated risk register is shown in table 3.3.2.

ID	Date raised	Risk Description	Likelihood of the risk occurring	Impact if the risk occurs	Severity	Owner	Monitoring Strategy	Mitigation Plan
28	13/10/2023	Configuration on git risk	Low	Medium	Low	Scrum team	Ensure that the token in gitlab is set up properly and the expiry date is after the project end time.	1. Configure a new gitlab token and fix the configuration issue as soon as possible. 2. Make sure that changes made to the code are immediately pushed into the branch, to prevent any data loss.
29	13/10/2023	Conflict issue during commits in git risk	Low	High	Low	Scrum team	Always pull changes made by other team members in the branch before pushing the code.	1. Check git history for the different versions of code to check which parts were not changed. 2. Regularly test and check the code to make sure all the things done are still there.

Table 3.3.1: New Risks Discovered

ID	Occuration Date	Risk Description	Severity	Details	How the issue was resolved	Future Mitigation
3	5/10/2034	Team	High	1. During this	We held more	1. Meetings should

		member burnout risk		sprint, it was throughout the end of the semester. 2. All team members felt burnout due to the amount of commitments and workload. 3. Some time was wasted and some team members were not able to commit to the project fully.	standup meetings, to constantly make updates. Then, the team committed to the project well.	be held not long after each sprint end, to re-motivate team members. 2. Make sure that all team members are feeling fine to continuously work on the next sprint.
22	6/10/2023	Data Loss Risk	Low	1. One team member pushed and committed the changes made into the main branch. 2. This resulted in some inconsistencies and some changes made were gone.	Changes were reverted and the lost data was retrieved from the commit histories. No data was lost officially.	1. Ensure that backup copy is made and saved after every change. 2. Utilise gitlab for version control to save code to a repository. 3. Ensure that all team members have their own backup of the new code.
5	10/10/2023	Lack of communication risk	Medium	1. A few team members had a miscommunication on the task that was done in different ways. 2. This resulted in both teammates having different ideas and goals on the project.	This was solved during a standup meeting that was held on the same day, and the same goal was maintained.	1. More standup meetings must be conducted.
6	12/10/2023	Unrealistic or unreasonable requirement	Medium	1. Group had a misunderstanding on the requirements that the teacher	The group had a client meeting with the client, discussing the requirement, and negotiated. Came to	1. Communicate with the client when needed. 2. Negotiate for any unrealistic

		s		set. 2. This resulted in the group not being able to continue on the idea they had in mind.	a conclusion and was finally able to continue on the project.	requirements.
16	12/10/2023	Inconsistent user experience across platforms risk	Medium	1. A few team members had different experiences on the same app that was being worked on. 2. The things worked on were not being shown on another team member’s end.	Found out later that the changes made by one member were overseen, so necessary changes were made. Solved in a timely manner.	1. Have frequent discussions on the changes made and the experience on the app. 2. Frequently update the team.
28	12/10/2023	Configuratio n issue on git risk	Low	1. One team member had a configuration issue in git, which did not allow the user to pull or push any code from the branch. 2. This delayed the progress of the team member for a short time.	The issue was quickly resolved on the same day as a new token was generated and the cloned git repository was re-configured. No data was lost during the process.	1. Configure a new gitlab token and fix the configuration issue as soon as possible. 2. Make sure that changes made to the code are immediately pushed into the branch, to prevent any data loss.
29	13/10/2023	Conflict issue during commits in git risk	Low	1. There were some conflict issues between two team members. 2. Changes made in git weren't pulled first before carrying out the task but no conflicts were detected when merging. 3. This resulted in changes made by the previous	We checked through the commit histories to find out the parts that were missing, and added it. No data was lost and it was a quick fix.	1. Ensure that all changes are pushed to git. 2. Pull the code from the branch, before pushing and committing the code.

				team member not being carried to the newest commit.		

Table 3.3.2: Live and Updated Risk

3.4 Sprint Review Planning

This is the sequence for the demo:

1. Brian, the Scrum Master and Assistant Programmer, will briefly introduce the team, sprint goal and project information.
2. Ibrahim, the Product Owner and Front End Developer, will briefly show the sprint backlog and user stories in Trello.
3. Brian, the Scrum Master and Assistant Programmer, will demonstrate the login page as admin.
4. Ibrahim, the Product Owner and Front End Developer, will briefly go through the software features from sprint 1 and 2.
5. Brian, the Scrum Master and Assistant Programmer, will briefly go through the user information page.
6. Diana, the UI/UX Designer, will introduce the change password feature and the individual contributions page.
7. Lucas, the Database Programmer, will demonstrate the individual contribution functionality along with the graph function as well as the edit and remove functionality.
8. Chua, the Head Programmer, will demonstrate the overall contribution functionality which includes the calendar page.
9. Rui En, the Technical Writer and Tester, will demonstrate the functionality that limits the number of sprints in progress to only one.
10. Ibrahim, the Product Owner and Front End Developer, will wrap up the demo.

3.5 Product Examination

To ensure that the product is complete and shippable after the third sprint, the tester in the team has performed an intensive review of the product and cross-checked it with the user stories recorded in the sprint backlog of sprint one, sprint two and sprint three. The testing method used is user acceptance testing and integration testing. The tester had produced a report after the testing (see the Testing Report linked in Section 3.1). The tester has also reviewed the user acceptance criteria and tasks checklist for each user story to ensure that the user stories in the sprint backlog are actually completed. The designated user stories were confirmed as "done". Other than that, integration testing was done by the team collaboratively contributing to multiple unexpected inputs to challenge the software's functionality. These inputs were designed to find any potential weaknesses or unanticipated scenarios that users may attempt on the website. Through this, we enhanced the software's overall quality, making sure that it is ready to be deployed.

4. Sprint Retrospective

4.1 Main Criteria

Full discussions on the sprint retrospective can be found in the meeting agenda for meeting 3, attached in section 3.1.

Based on the template given, the team has performed sprint retrospective on sprint two. We then, evaluate the sprint based on the following points:

1. What went well?
2. What were the problems encountered?
3. What could have been done better?
4. What will we try next?
5. What questions do we have?

What went well?	1. We had more standup meetings during this sprint compared to the previous 2 sprints and therefore were able to keep each other accountable and make sure that everyone performs their tasks on time. We made sure to update the group more often during this current sprint, which made us to be more aware of each other's task and help out where necessary. These factors ensured that we were able to progress well as a team as we were more efficient this time compared to previous 2 sprints. 2. The team efficiently and proactively completed and added new required task allocations to each user story that was in the progress of getting completed which resulted in a very efficient workflow. Our forward thinking approach also ensured all aspects of a user story were addressed comprehensively. 3. During this sprint, the tasks were allocated more based on the member's interests and the skills they are good at, this resulted in all members being able to complete tasks on time within a short time frame. Everyone was able to enjoy the process more. This allowed smooth and fast progress for the team. 4. This time around, more meetings were held and thus more frequent updates on tasks that are being worked on hence everyone knows what's going on making sure everyone has a clear understanding of the ongoing work and making all the work go smoother and more efficient. 5. Integration testing is done at an earlier stage, so more errors can be detected during sprint 3. This can improve the overall quality of the software. Besides, the sprint can go smoothly as all the bugs are addressed at the earlier in the development cycle. 6. The team completed all sprint tasks ahead of schedule, allowing for additional integration testing so that bugs can be found and fixed earlier. This means lesser time spent on finding the bugs and fixing them later on.
What were the problems encountered?	1. The problem encountered was instances of understanding the content required to code the website because most of us did not have prior knowledge in javascript, html and css. Therefore, there was a high level of difficulty experienced when it comes to coding and understanding the languages required to code the website. However, we were able to overcome this issue much more effectively during this sprint as we had more experience gathered from previous 2 sprints in searching online resources for available help that we could use for self learning.

	2. There were a few bugs resulting in 404 errors due to an inconsistency in the code and the website hosting platform redirect links. These inconsistencies led to navigation issues for users trying to access specific pages. Fortunately, the team was able to identify and rectify it speedily. 3. There were some instances where team members were quite unclear with the specifications for this sprint, which resulted in some time lost on the team feeling confused, and was not able to proceed with the task. Some of the user acceptance criterias and features were also changed completely due to our misunderstanding in the tasks. This also resulted in some conflict amongst team members during discussions, due to different ideas that everyone had in mind. 4. There were quite a few issues with the functions of the code which were quite complex and took quite a while to fully understand what needed to be fixed. This complexity is often stemmed from intricate interdependencies between various parts of the codebase, making it challenging to pinpoint the exact source of the problem. 5. There was difficulty in doing integration testing due to not being familiar with it. This is because integration testing requires the tester to understand all function files and it would take some time to understand some functionality. This problem was solved by seeking help from team members. 6. Some of the documentation for completed tasks was often incomplete or unclear, causing delays when we had to build upon that work. This is because some tasks require understanding for the other tasks done by other team members.
What could have been done better?	1. We could have spent more time on testing the software. This would ensure that we would be able to identify and discover the bugs existing in the code much more effectively. With this, we would be able to resolve the issue much earlier and would be able to be more effective in progress. We could have also done more iterations of testing so that we would be able to discover more bugs existing in the current codebase and would be able to resolve them earlier. 2. We could improve our understanding of the deployment and database environment to ensure every team member was able to use the respective services as needed. This collective knowledge reduces the dependency on a few experts within the team. 3. Before the sprint started, we could have asked the client for more information regarding this sprint, so that we could have a clearer understanding of the tasks that we had to do during this sprint. By doing this, we could have understood what to do earlier on the sprint, and would be able to start doing the tasks without many obstacles on the way. This way, everyone has a clear idea and the same goal in mind too. 4. We could have communicated better on how the functions actually needed to be changed or fixed when there was a bug or error. This lack of clear communication often led to misunderstandings within the team causing delays in addressing issues and sometimes resulting in suboptimal fixes that didn't fully resolve the root cause of the problem. 5. We could have learnt how to use automation testing earlier to detect and prevent sudden, undetected bugs in our software. By doing automation testing, the tools can run a large number of test cases in a relatively short amount of time. Thus, we can obtain early feedback from the tools we use. 6. We can designate a team member each sprint to be responsible for overseeing and ensuring thorough documentation of the code. Assigning this critical role ensures that there's a consistent standard of documentation maintained

	throughout the project. Proper documentation not only makes the codebase more understandable and accessible to current team members but also aids any future developers who might work on the project.
What will we try next?	1. We would try to prepare ourselves much more in advance before coming into the sprint. This could be by having more self learning before the sprint so that we would be able to know how to code the website out much more efficiently during the sprint. This would decrease the time spent on figuring out the bugs as we would be able to have prior knowledge coming into the sprint and would be able to prevent most bugs from occuring in the first place with the knowledge that we obtain. 2. We would try to implement incremental testing of deployment to help with the early detection of issues related to the compatibility of our code and the web hosting service. This is because this proactive approach would ensure smoother deployments and reduce last-minute fixes. 3. We would try to get more information regarding the sprint earlier on the sprint, by conducting more client interviews in the beginning of the sprint. This is to ensure that all the necessary information needed for the sprint is clear early on, to prevent any redundant work or confusion. This way, the team will also be able to work on the tasks with clearer instructions and that the work done would completely suit the client’s needs. 4. We would try to have more coding meetings on how to tackle the issue or bugs whenever there is one. These meetings can serve as a proactive forum for team members to share insights and brainstorm solutions and ensure swift and effective response to any challenges that may arise. 5. We would try automation testing to test the software, so we can detect more bugs in it rather than manual testing. Using automation testing can increase the efficiency, effectiveness and coverage of the testing process and reduce the risk of human error, which is common in manual testing. 6. We would try to organise weekly code review sessions to maintain consistency in coding practices and catch potential issues early. This could be done physically or online through Discord. These sessions can serve as a platform for ensuring code quality but also promote continuous learning and skill enhancement within the team.
What questions do we have?	1. How can we be more efficient during the sprint and what additional tools can we incorporate into our practice that would improve our overall productivity? 2. What indicators should we be aware of to know if we're moving in the right direction? 3. How can we make sure that all the tasks that will be done completely suit the client’s needs? 4. How can we ask more questions about the tasks that needed to be done to ensure clarity on the given task to have higher efficiency? 5. How can we deploy our software and ensure its maintenance post-deployment? 6. How can we maintain a balance between delivering features and ensuring the quality of the code?