PROJECT INCEPTION

FIT2101 - Team Terang Bulan

https://git.infotech.monash.edu/fit2101-s2-2020/Terang_Bulan/project

A.Project Plan

1. Project vision

For Students and Teachers
Who wants to keep track of tasks allocated in a project and track the time
The TERBUL™ Tracker
Is Student and Teacher project tracking system
That help teacher with marking student based on their proportion of work in a project
Unlike any other project monitoring software
Our product can provide simple user experience

Vision Statement
“Our vision is to make an easy monitoring-tracking task allocation experience for students and teachers by providing simple user experience software.”

2. Team members

Name	Student ID	Contact	Role
Ivan Wijaya	29633524	iwij0002@student.monash.edu	Project Manager
Felicia Winna	31058825	fwin0002student.monash.edu	Scrum Master
Michael Shane	30666481	msha0080student.monash.edu	Technical Developer
Angeline Tanvy	30930227	atan0064student.monash.edu	Technical Developer
Sulthan De Neiro Raihan Syah Bungin	29906164	sbun0004student.monash.edu	Quality Assurance

3. Responsibilities of Each Role:

(a) Scrum Master:
Keeps track of the product backlog, ensures that every goal and specifications of the product owner are well understood by the entire team and collaborates with the project manager to plan the week to deliver the product owner’s requirements on time.

(b) Project Manager:

Plan a timeline and organize the plan for the week such as what progress needs to be achieved in order to deliver the product on time and on specification, communicate with other team members to make sure that everyone is upholding their end of the work, make changes to the timeline if any problems arise in the middle of the project, help to develop the code each week, and help to resolve any conflicts that arise in the project or between team members.

(c) Technical Developer:
Understand the weekly goal and develop the code to meet the weekly goal or surpass it, collaborate with the other members of the team if the current state of the product seems to deviate from the final product specification.

(d) Quality Assurance:
Collaborate with the project manager to make sure that the goals for each week will help achieve the product requirement on time and propose changes to the weekly plan, help develop and monitor the quality of the product each week and report to the team of anything that needs to be improved or changed to further meet the client's requirement.

4. Roles

Role	Level	Stakeholder Management
Project Team Member	Actively Engage	Whatsapp Group
Kamala (Client)	Keep Satisfied	Email and Weekly Meetings
Student Users	Monitor	Feedback
Teacher Users	Monitor	Feedback

5. Team Process Model:

The Software Process we are using is the Agile process model. Agile process model suits this project as the project duration is relatively short (1 semester), with some consideration of unpredictable requirements that can be added throughout the progress of the project. Because we are using Agile methodology, we keep the product requirements as a base that can be improved later on, depending on the feedback of other stakeholders on each project iteration.

Some of the parts in the agile framework that we are using are almost similar to Scrum. The difference is that we don't do daily standup meetings, instead, we do weekly meetings outside the Friday 1-4PM MYT workshop time. We will document each meeting discussion and how the progress of the project develops in each meeting, therefore the number of documents will be relatively less than when we use the original scrum.

We believe by doing this process model, it is more suitable to us and we can work efficiently more in this way. Weekly meetings will give us more time to do our assigned tasks rather than daily meetings. In addition, our team members will not feel rushed or pressured, as we are given a week to report on what we have done. If we conducted daily meetings, it would be strenuous on our time as students. Thus, it will maximize our work effort to give our best to our client, in which we have set the product owner to be Ms. Kamala, who is our FIT2101 lecturer and tutor.

6. Definition of Done:

We will consider our project as the final product if our project has met all of the client's requirements (user stories) and it is fully functional. Code testing by our quality assurance engineer will play a crucial role here, making sure that our product's quality serves the desideratum of our client. Besides just fulfilling the client's requirements, ensuring their satisfaction with the features we have implemented is also our top priority. For more detailed Definition of Done for each user stories, it is already listed in each of the user stories in our product backlog in Trello.

7. Scrum Ceremony:

Sprint Planning: Zoom Meeting each week (in the beginning of the week)
Daily Scrum: WhatsApp Group daily report on task progress (usually not done daily)
Sprint Review: Meeting to review each week process (by the end of the week)
Retrospective: Conducted by the end of every iteration

8. Task and Time Allocation:

This project will be broken down into multiple tasks and allocated to each member for each week. The task allocation will be managed by the project manager to each member of the team. Each member has to finish their task in the allocated week.

Task Allocation Table
https://docs.google.com/spreadsheets/d/1hSjX_738DjWjBHIZo7WogAj1Nnfz4vhJyJuK0vjMgPk/edit#gid=0

9. Storage and Managing of Backlogs

To store our project backlogs and also organize them, we use GoogleDoc sheet as our brief display. For the overview of the product backlogs implementation history, we use Trello. For more specific detail of the backlogs, we will be using GitLab through SourceTree to update our progress to our repository as it has provided the whole project timeline.

Trello for Product Backlog
https://trello.com/b/xFU7KLWJ/fit2101

GoogleDoc for Brief Display

Product Backlog

B. Analysis of Alternatives

Our Choice:
Programming Language: **HTML, CSS, Javascript, Python**
Frontend Framework: **None**
Backend Framework: **Django**
Deployment: **Heroku**

Alternative 1:
Programming Language: HTML, CSS, Javascript
Frontend Framework: Vue
Backend Framework: Express and MongoDB
Deployment: Netlify

Alternative 2:
Programming Language: HTML, CSS, Javascript
Frontend Framework: ReactJs
Backend Framework: Firebase
Deployment: Github.io

1. Platform:

Firstly, there are two approaches we can make for this project, **an application** or **web-based**. However, after considering the client's requirements and user stories, we decided it should be completely web-based as it will suit the client's requirements perfectly, which is to be able to access it from both desktop and mobile wherever they are. For that reason, web-based applications are far more efficient in fulfilling client requirements rather than making multiple apps for multiple type devices.

2. Programming Language:

HTML and CSS:
The programming language to display the website user interface. Easy to learn and necessary for web development, however team members do not have much experience in this language.

Javascript:
The Programming language to support display, interactivity, logic, and other aspects of the websites. Team members have slight experience in this language.

Python:
One of the basic programming languages that most of the team members are familiar with. This language will be used to handle the logical system of the product.

3. Framework:

Framework is the support structure which helps the developer to develop web applications without having to do tedious tasks and also provide extra functionality which may alleviate difficulties in web developing experiences. Although developing a frameworkless website is possible, using frameworks may reduce the time needed to develop the website since the developer may not need to hook-up some front-end features compared to developing a website without having any framework. However, this benefit comes with the cost of time, due to the time needed to master a framework will be greater than the time needed to learn a frameworkless method.

For this project, since our aim is to emphasize on developing the product rather than learning such high level software, therefore we prefer to **not use any frontend framework as our main choice**. If the condition to fulfill our end-product is heavier than we expected, therefore we will use the framework as our alternative.

Frontend Framework:
Frontend Framework is the framework that helps to display data as part of user interface and user experience.

Aspect	**Vue**	**ReactJs**
Backed by	Open-Source / Community	Facebook
Startup time	48ms	72ms
Memory used	7MB	9MB
Code Reusability	CSS and HTML	CSS

There are several well-demand frameworks used by companies and after analyzing and researching about multiple frameworks, we have shortened our choices to **Vue** and **ReactJS.** Both frameworks have Javascript as their base language. This will work to our advantage as the team members are already familiar with language, along with the low difficulty level to learn those frameworks. Regarding the comparison of Vue and ReactJs, we have summarized up the some aspects differences:

- **Company Backing**
  From the table we can infer that ReactJs is more on advantage as it is backed by a well-managed company (Facebook) while Vue does not have a formal company backing, it is an open-source framework developed by the community.

- **Developer Friendliness**
  Even though ReactJs is more popular among web-developers, we have found out that Vue has more readable code and a user-friendly environment compared to ReactJs.

- **Startup Time**
  ReactJs needs a slightly longer time for its startup compared to Vue.

- **Memory Used**
  In case of run-time memory, Vue consumes less memory than ReactJs. Thus, Vue will save us memory on this project.
- **Code Reusability**
  Regarding code-reusability, Vue allows us to reuse our code, functions and templates with more variety of languages as ReactJs offers a very limited choice.

After much consideration of the advantages and disadvantages above, we decided to prioritize **Vue** as our first alternative and **ReactJs** as the second alternative. **However**, because it might take longer to learn these frameworks than actually implementing them, we decided to prioritize using only **HTML, CSS, & Javascript** as our main option and took **Vue** as our alternative.

Backend Framework:
Backend Framework is a framework that supports the logic of how the website works, including fetching or storing data. We have stated that it may not be necessary to have a frontend framework as part of our project, however it is necessary for backend due to its simplicity of usage which saves time both learning or developing the product.

After some research, the alternatives that are suitable for this project are Django, Express, Firebase.

Django
- Advantage: Is Python based which is mastered by all the team members
- Disadvantage: May release a lower quality end product compared to the other 2 frameworks.

Express
- Advantage: Supports the Vue JS Framework and is based on Javascript which is quite familiar to the team member's experiences.
- Disadvantage: Requires an extra 3rd party application such as MongoDB which may take extra time for team members to learn.

Firebase
- Advantage: Supports the React Framework and has a large community to support any difficulties team members might face.
- Disadvantage: The framework itself is the toughest to learn and master among the 3 chosen frameworks which may cause the team members to spend extra time to learn it instead of implementing it.

As previously mentioned above, our team would prioritize the backend framework that takes the shortest time to learn so that we could focus on implementing the actual product, hence we are choosing **Django as our main choice** followed up by **Express** and **Firebase** as our alternatives.

4. Deployment

Deployment is the step to place a website on an online server. For this project, we want to choose an option to deploy a website that **does not require any funds**. This project is also not high-scaled therefore it is unnecessary to use complex website development tools to accomplish the goal of this project since it may cost more time to learn them.

After doing research, the possible options for web deployment are Netlify, Heroku, Github io.

Heroku

- Advantage: Backed-up by a large community and has add-ons to support backend. There are also plenty of learning resources that will help in deployment.
- Disadvantage: Server is asleep after some inactivity and has extra set up to configure that may slow down product development

Github IO

- Advantage: Easy to learn and very user-friendly which may reduce the time needed for the project to be finished
- Disadvantage: May not work if its developed with website that rely on Framework and potentially create some extra problem decelerate the development

Netlify

- Advantage: Intuitive for beginner and Supportive with Vue JS Framework which may not burden team members when using it.
- Disadvantage: Team member's learning experience may not be easy due to not much support from the community

As previously stated, our team choses **Heroku as our main choice** due to it able to deploy dynamic website and plenty of learning resource available among the other 3 followed by **Netlify as the first alternative** and **Github.io as the last alternative**.

5. Communication Tools:

Chat: Whatsapp
Voice communication: Zoom

Alternatives:
Chat: Line
Voice communication: Discord

C. Risk Register

Project risks are not avoidable when working in a team. Therefore, we listed down several risks that are likely to happen in order to keep them monitored and to prevent them from happening in the future.

Here are the possible risks:

1. Team member becomes ill or has another personal emergencies and can't deliver their work before the deadline
    - Monitoring Strategy: Frequent communication between team members
    - Mitigation Plan: Discuss within Team Members then, re-allocate tasks

2. Team member is incompetent to master required skills on time which may reduce the time to implement the product
    - Monitoring Strategy: Frequent communication between team members, also set deadline for skill learning
    - Mitigation Plan: Learn the corresponding skill faster or find the alternative skill to develop the project

3. Sudden major project changes by stakeholder that the project can't be finished on time
    - Monitoring Strategy: Keep in contact with high influence stakeholders (client)
    - Mitigation Plan: Negotiate the major project change based on the earlier requirements.

4. Conflicts occur when developing the project, causing delay in progressing and might threaten the project.
    - Monitoring Strategy: Conduct weekly scrum meetings to avoid misunderstanding and miscommunication, keep team's harmonization to ensure a pleasant work environment.
    - Mitigation Plan: Discuss and resolve the conflict open-mindedly while respecting others' opinion too. So that in the end, all of us can be on the same ground and also understand each other's perspectives.

Meeting Minutes

Below is the report from our previous meeting for this project:

1. 28th August 2020: Create Project Inception and Project Prototype
   Outcome ⇒ The base of Project Inception was created in Google Docs, and the base of Project Prototype was created in Google Slides.

2. 31st August 2020: Continue creating Project Inception
   Outcome ⇒ Most of Project Inception is done

3. 4th September 2020: Finish Project Inception, and possibly add user stories
   Outcome ⇒ Edit the Project Inception, and now it is ready for submission

4. 8th September 2020: Finish Project Prototype and share Django learning source
   Outcome ⇒ Finish Project Prototype, Django learning source shared, started on sign up and sign in page

5. 11th September 2020: Present Project Prototype, ask about what needs to be done for iteration 1 and finish them (except the coding)
   Outcome ⇒ Presentation is delayed to next week, done with Planning Poker for all user stories

6. 15th September 2020: Create add project page; design the UI for sign up page, main page, and add project page; create project database
   Outcome ⇒ Create base for sprint review & product backlog and divide task for UI design

7. 18th September 2020: 1st Sprint review with stakeholders to get feedback on the product backlog and product demo
   Outcome ⇒ Client provided feedback on the project, fixed the task and time allocation management.

8. 20th September 2020: Prepare for 2nd Sprint review, specify all user stories for the next sprint review
   Outcome ⇒ Confirm the user stories for the next iteration with the stakeholder

9. 25th September 2020: Implement the Assign student to a project and divide the rest of tasks to all members
   Outcome ⇒ Assign student to a project is implemented and managed to divide the task equally to all members

10. 29th September 2020: Fix bug & navigation issues, implement multiple task contribution
    Outcome ⇒ Fixed bug & navigation issues, implemented multiple task contribution, and improved the UI design

11. 7th October 2020: Check preparation on Iteration 2, finish sprint review 2 report
    Outcome ⇒ Finish checking on user stories for iteration 2, and update product backlog report for iteration 2

12. 15th October 2020: Prepare Sprint Review 2, add some developer stories
    Outcome ⇒ Sprint Review Preparation, UI Fixes, Quality Testing, User Story Testing, Added remove student from task, Bug Fixes, change add student to task algorithm

13. 21th October 2020: Fix Time and Task Allocation, finish revamp the UI for navigation bar and profile page, change the design of sign up and sign in page
    Outcome ⇒ Fix Time and Task Allocation, finish revamp the UI for navigation bar and profile page

14. 23th October 2020: Ask user for features to be done in iteration 3
    Outcome ⇒ User stories for iteration 3 are already received, done 1 of the user stories

15. 29th October 2020: Prepare Time tracker and Product for Sprint Review 3
    Outcome ⇒ Time tracker done, and update product backlog report for Iteration 3