**Software Process Selection and Project Plan**

**<P05>:<INSTASHOP>**

**<team member names & ids>**

| **Student ID** | **Name** |
| --- | --- |
| **23100016** | **MUHAMMAD AFFAN ASHRAF** |
| **23100233** | **MUHAMMAD UMAIR MOHSIN** |
| **23100254** | **SALMAN MASOO** |
| **23100268** | **FARVA TALIB** |
| **23100091** | **MUHAMMAD BILAL SHAHID** |

**Table of Contents**

[1.](#_heading=h.gjdgxs) Introduction 3

[2.](#_heading=h.30j0zll) Software Process Selection 4

[3.](#_heading=h.1fob9te) Gantt Chart 5

[4.](#_heading=h.1t3h5sf) Risk Management 5

[4.1 Potential Risks and Mitigation Strategies 5](#_heading=h.4d34og8)

[5.](#_heading=h.3znysh7) Development Environment Preparation 6

[6.](#_heading=h.2et92p0) Deployment Platform 6

[7.](#_heading=h.tyjcwt) Who Did What? 7

[8.](#_heading=h.3dy6vkm) Review checklist 7

# Introduction

<Give an overview of the project here. The overview must highlight the overall objectives of the project and its potential users and customers.>

InstaShop will be a web-based portal which aims to revolutionize influencer marketing. For those who may be unaware, influencer marketing is a recent development in the digital world which involves a brand collaborating with an online influencer to market one of its products or services. These influencers are usually found on social media platforms (Instagram and Snapchat) and have a decent number of followers to whom they market a particular brand’s products or services.

There are two main parties involved in this process, the client and the influencer. For better understanding of the motivation behind this project, we must first walk through the process that is generally followed in influencer marketing from the perspective of both the client and the influencer.

For the client, the first step is perhaps the most tedious which is related to finding the right influencer for your brand. This is usually achieved by manually searching social media platforms or relying on word of mouth from friends/family regarding a particular influencer. The second step is contacting the shortlisted profiles. This is either done through direct messages or emails, both of which are again tedious tasks with no guarantees of a timely response. The final step (assuming the contract has been fulfilled) is the issue of payment. This is perhaps the most troublesome aspect because there is no guarantee that the influencer will produce content that is upto the mark and not run off with any advance payments made.

For the influencer (unless they have a huge following) it is usually hard to find clients for collaborations or for sponsored content. Since these influencers are very active on social media, their inbox and comment sections are almost always flooded which means they tend to miss out on potential business opportunities just because they weren’t able to see the direct message. Some profiles do have designated emails for business inquiries but their response times are in most cases not ideal because they just don’t check their email that often. Finally, the issue of payment also exists. There is no guarantee that the client will pay the influencer the full amount in a timely manner even if the work has been done upto the client’s standards and deadline.

As visible, both of these parties are in dire need of a platform that could automate most of these tasks for them as well as provide payment guarantees. This is where InstaShop comes in. It aims to streamline all the steps involved in this process by providing an easy to use web application thus saving time and effort for both parties involved. For the client, it makes it easier to search relevant influencers courtesy of our database and filtering method, connect with shortlisted influencers (via email or live chat) and have your payment secured (via escrow).

For the influencer, you essentially get access to a marketplace where you can find potential clients, not miss out on potential business opportunities just because your inbox was too cluttered and have a guarantee that the client will pay you for your work.

# Software Process Selection

< (1) Discuss the pros and cons of waterfall and agile (scrum) processes in your own words.

(2) Select one of the above processes for your project development.

(3) Justify your selection with clear reasoning. Refer to “Project Context Analysis” in the slides to get help for writing this section.>

1. Waterfall Model :

Pros :

* It's helpful for developers and people to work quickly and up to speed since this model follows in order such as technical documentation which is helpful for them to understand the objectives.
* Helps testing process made easier and transparent
* The timescales are kept in order because in the waterfall model the phase developments enforces discipline which makes each step easy to monitor.
* Helps in dealing with issues in the design phase.

Cons:

* Needs can be difficult to define because of a structured plan. If a client wants changes to be made in any phase it has to be re-engineered to a large extent and follow the phases in steps.
* Potential lack of flexibility to cater for new development changes or requirement changes.
* Longer time for project delivery. As it takes much longer time for processes to complete in an order as compared to iterative approach methods such as Agile.

1. Agile Model :

Pros:

* Faster delivery of project to client. If you need a faster project completion in urgency can be completed by this method , it does not have the perfect increments but might help build the task.
* It’s Adaptable since the increments are small it's easy to modify and adapt the process of the projects according to the circumstances.
* Quickly helps in detection of problems since testing is done incrementally so if a problem occurs you can precisely solve it and fix it before the next cycle.
* It’s a transparent approach and all your workings are constantly shown.
* It’s a collaborative approach as Agile requires a lot of feedback back and forth between teams , clients etc so it helps for a collaborative environment and innovates creativity.

Cons:

* It causes some tricky paradigm shift because as some industry people use this approach but for some people however some people can’t adjust to fast paced rapid work
* There is a lack of overall cohesion as it's very easy for a continuous process to run itself and goals to get lost in details.
* There is a neglect of paperwork as well during the process. This approach often requires quick shifts from one aspect of a project to another. This may leave little time for doing the proper paperwork at each stage. Record-keeping is important, but it is often a casualty of agile working methods.

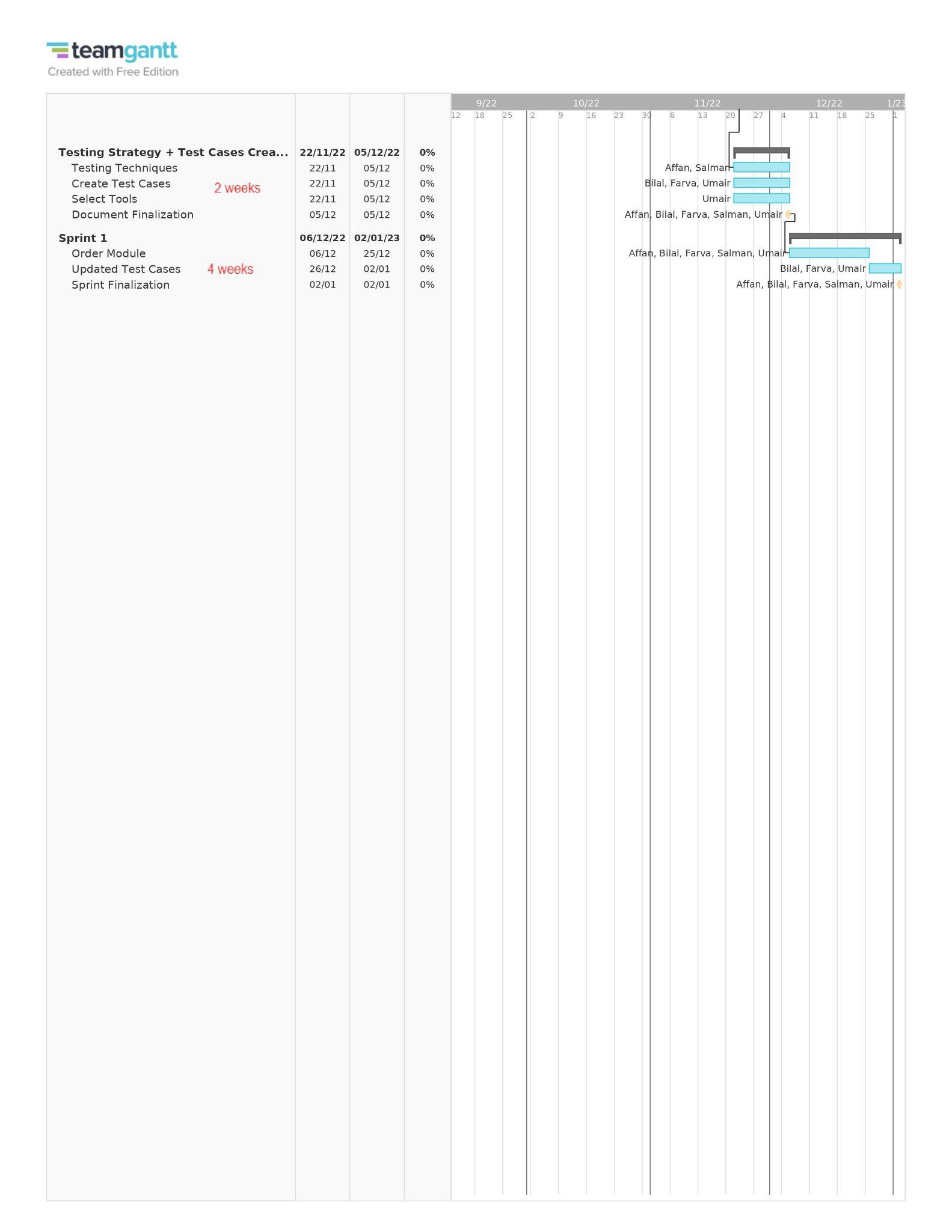
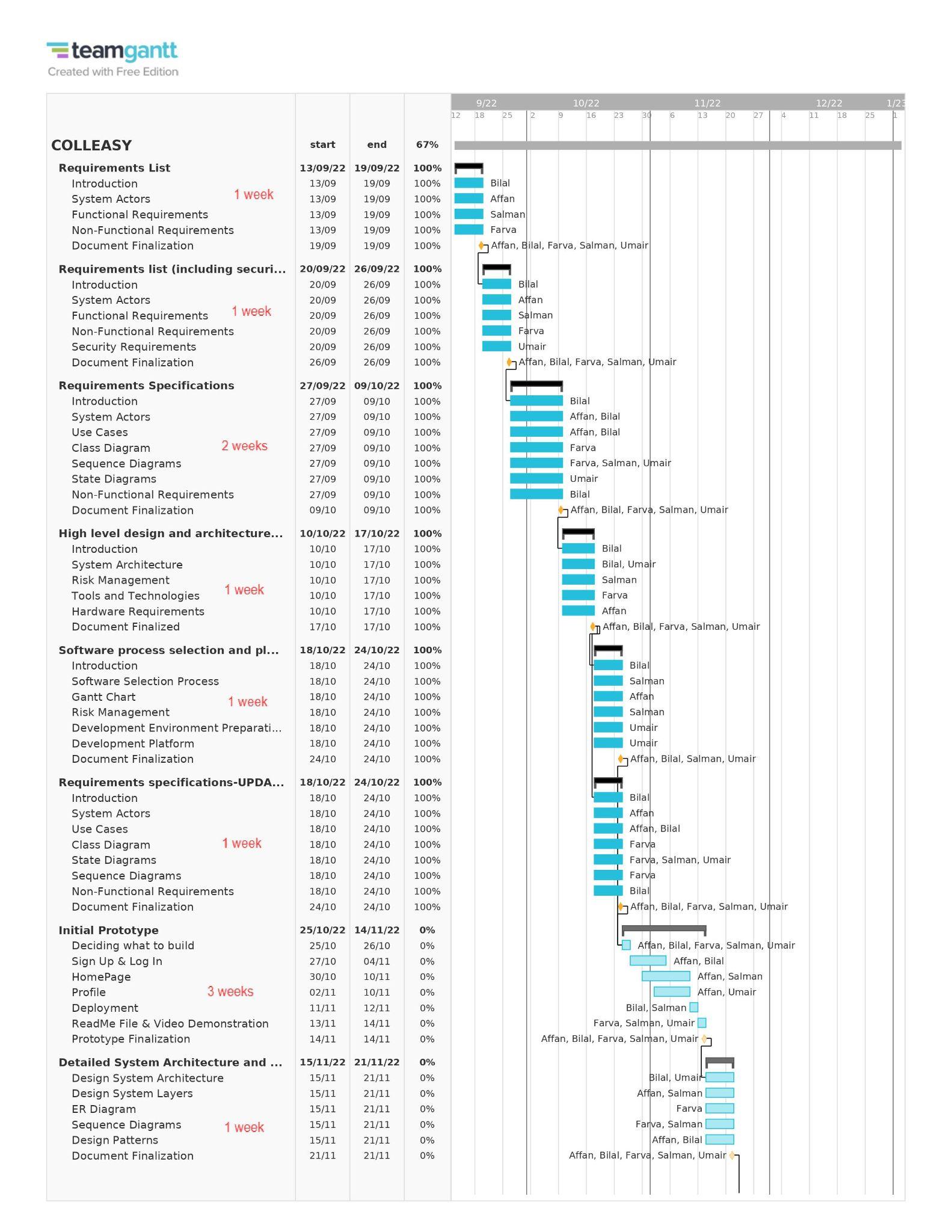
# 2) We will be following an agile model for our application.

# 3) The flow of our team work and requirements of the application make us inclined towards an agile model. Since this project is based on a time constraint therefore the requirements might need some flexibility or change in future as the development proceeds. Due to the short time and pressure of developing an early working prototype within the semester, our team requires an adaptive temperament where the changes are made instantly. Therefore it has a high rate of requirement change. Moreover, as our project requires constant feedback from the instructor, a flexible approach would suit the application which can only be obtained if we follow the agile methodology. This continuous feedback would also minimize risks in the application development and would help us better meet the user needs. Since agile methodology works in sprints which requires repetitive planning and meetings, the application is developed in small portions and is then added up to the previous version, this can cause unexpected errors and defects in the program. We need to have a model that has a low potential loss due to these defects. Fortunately, the agile model facilitates the developers by reducing the loss as, for instance, if an error arises in a sprint, it is only confined to that particular sprint and not the entire program. Only that sprint needs to be corrected. Therefore, we chose agile as it would help us save time in software testing.

# Gantt Chart

<Draw a Gantt chart that illustrates your project’s schedule. The Gantt chart should show at least the following

* Tasks (tasks should not be too small or too large)
* Duration (in weeks)
* Milestones
* Team member names who are going to work on each task.



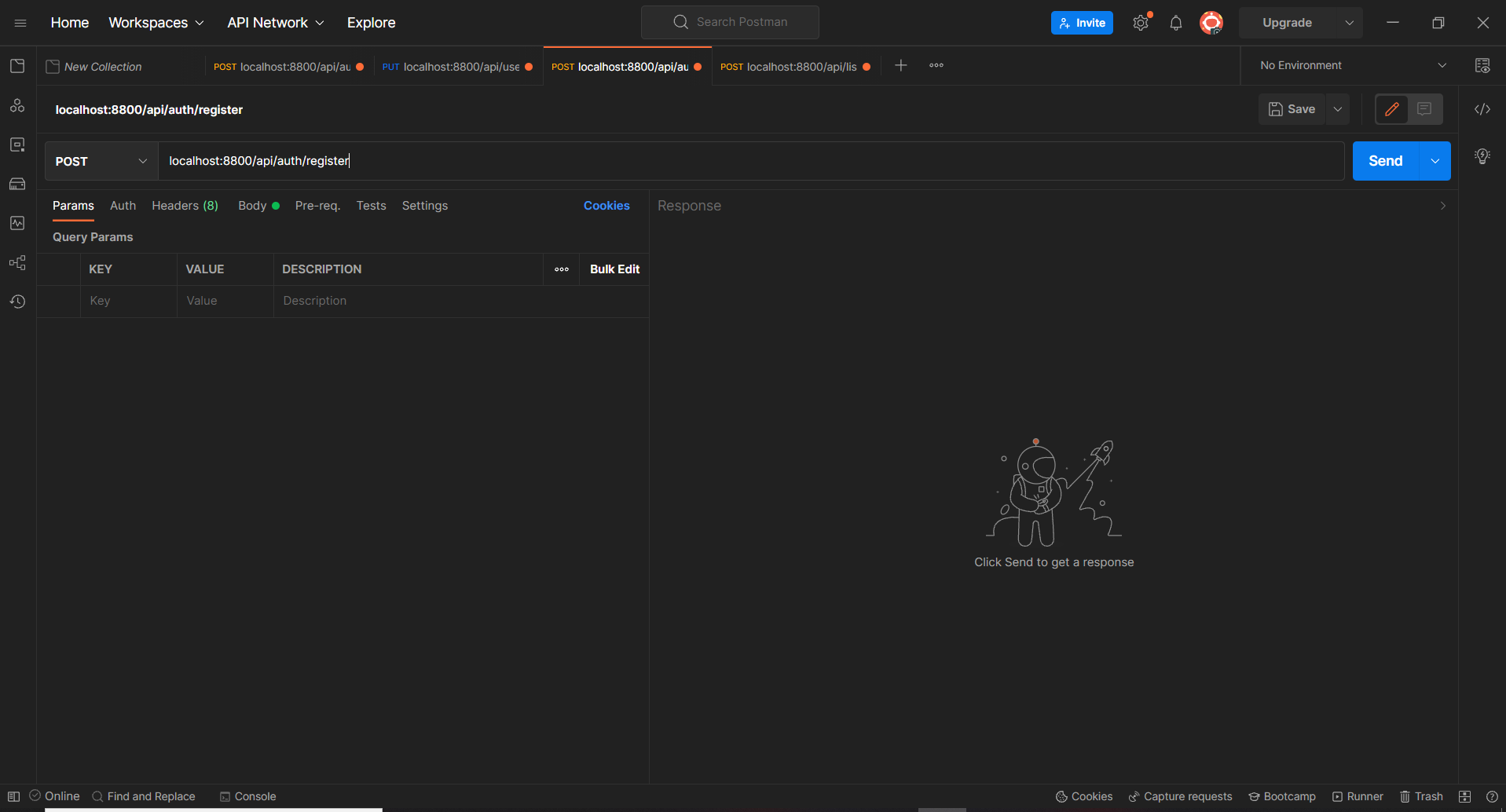
# Development Environment Preparation

< (1) List down tools and technologies that you will use for prototype development.

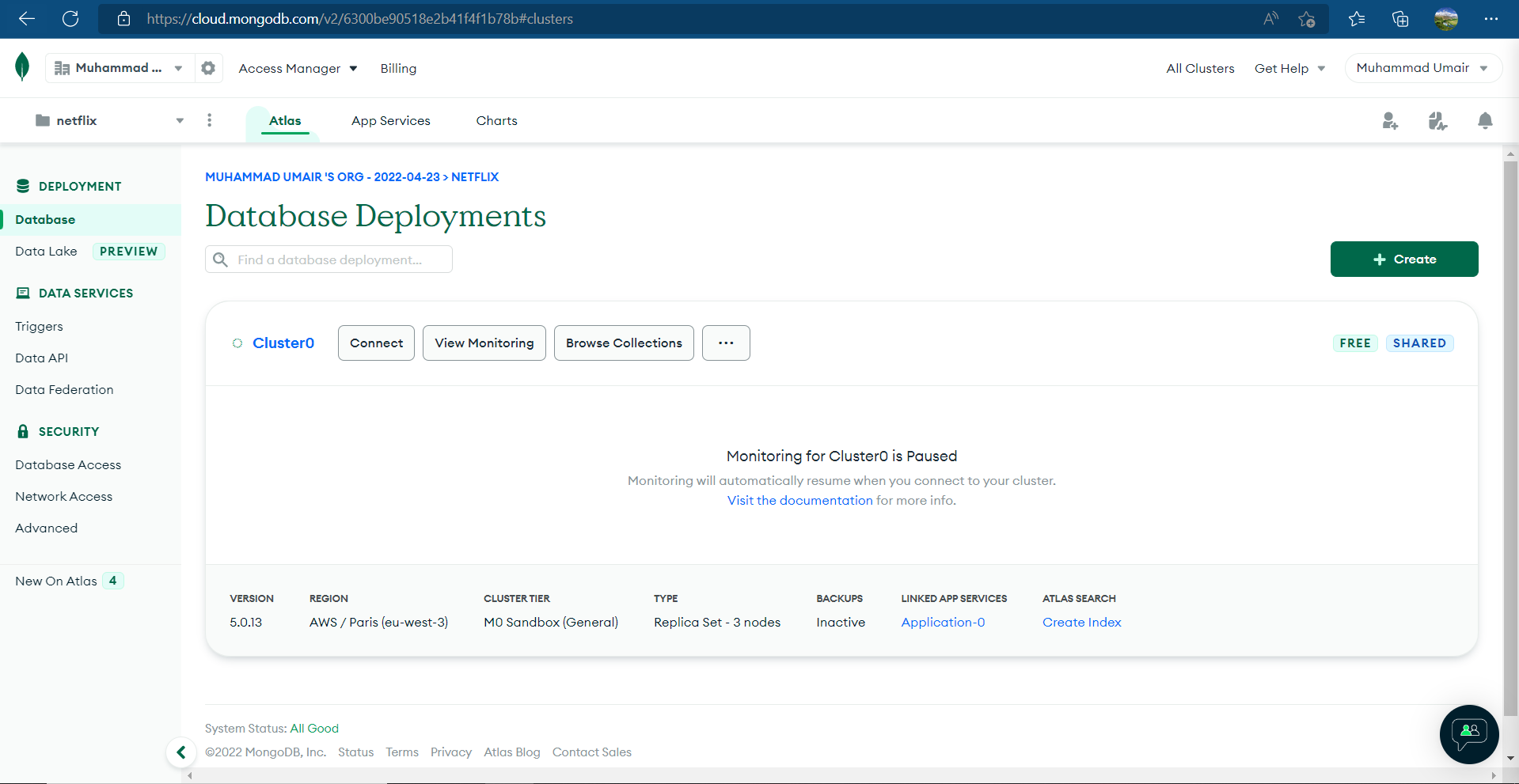
(2) Setup the development environment on your machines and mention here that you have actually setup the environment. Include three snapshots of the tool(s) that you are going to use for development. These snapshots must be taken from the tool(s) while they are actually running on your system.>

1. We will be using several tools for this development project. The main technology we will be using is MERN Stack and for the DataBase will use MongoDB atlas which stores all the data on the cloud hence much faster. Also for the APIs testing and building we will be using **POSTMAN** which will help us streamline the process much efficiently and faster.

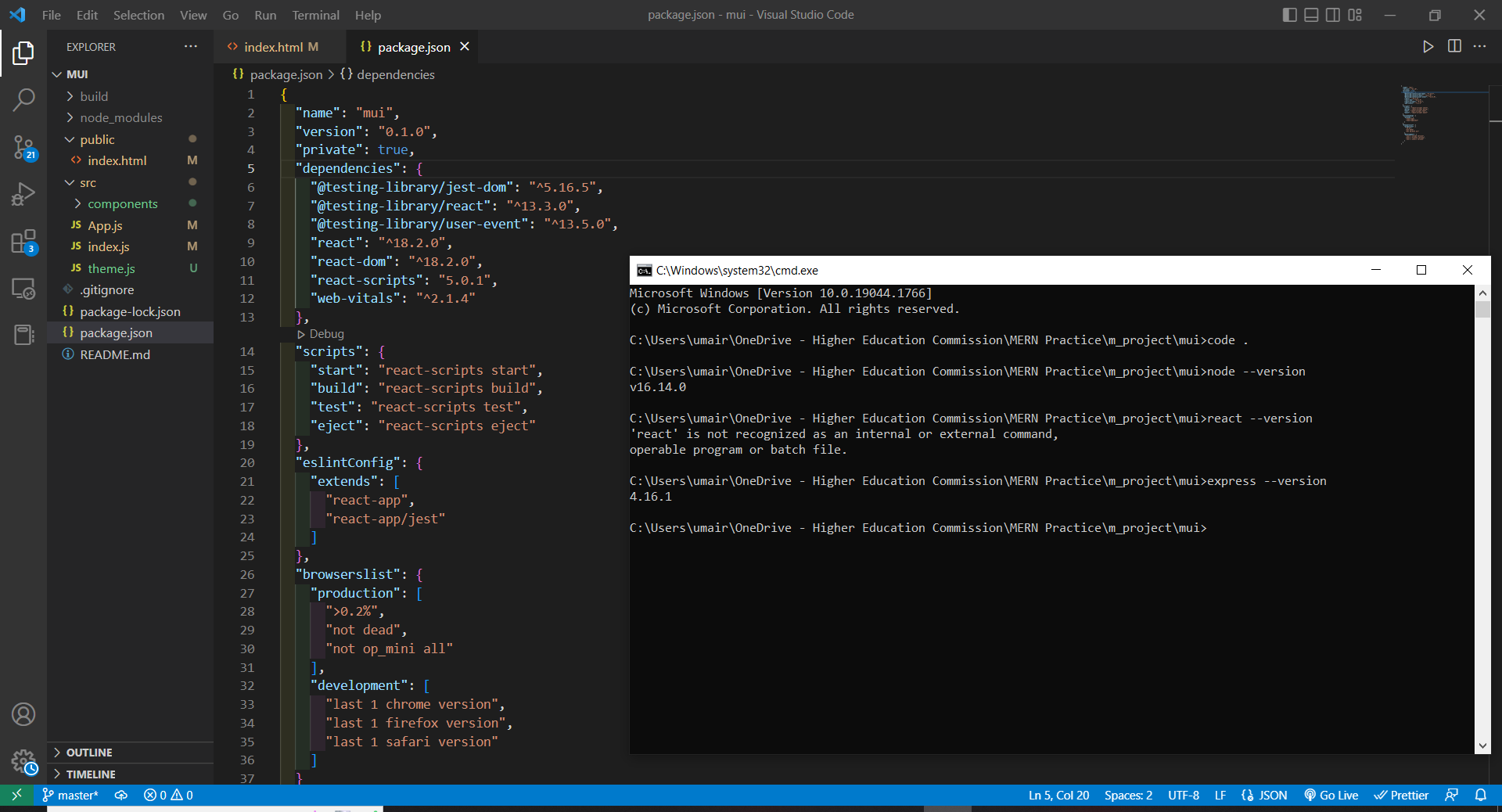
**POSTMAN**



**MongoDB Atlas**

****

**MERN STACK setup**

****

# Deployment Platform

<Find a free hosting service where you can deploy your prototype for anyone to access online>

So for using a free hosting service we will deploy it using Infinity Free hosting services. There are some limited good features offered by the website in their free package such as :

* Free subdomains
* 5 GB disk space
* Unlimited bandwidth
* 400 MySQL databases
* Knowledge base support

We will also try to use Hostinger which is a paid hosting website but with a minal fee of approx $3 per month we are able to get a lot of features such as **100 GB** SSD Storage , **Free** Weekly Backups , **Unlimited** Free SSL , **Free** Domain , **Unmetered** traffic (Unlimited GB) , **Free** Email which is very useful and gives a much faster results.

// we can also consider aws.

Our web application will be hosted by Amazon Web Services. AWS (Amazon Web Services) is a division of Amazon that offers a safe cloud computing platform with computing capacity, database storage, content distribution, and other features. We chose it for the reasons listed below:

User-Friendly:

Amazon Web Services offers a user-friendly interface called the AWS Management Console. Utilization of the platform is comparatively easy owing to the company's well-documented web services APIs.

Flexible:

You can select the operating system, programming language, web application platform, database, and other services you need through AWS. Any service or piece of software can be easily loaded into a virtual ecosystem using the platform.

Security:

AWS has an end-to-end strategy that incorporates physical, operational, and software measures to secure and harden its infrastructure.

Scalability:

AWS offers scalability through capabilities like elastic load balancing and auto scaling, which may be utilized to scale up or down an application in response to demand. Furthermore, you have access to storage and resources anytime you need them because it is supported by Amazon's extensive infrastructure.

# Who Did What?

| **Name of the Team Member** | **Tasks done** |
| --- | --- |
| Muhammad Umair Mohsin | Development Environment Preparation & Deployment Platform |
| Salman Masood | Software process selection second and third part |
| Muhammad Affan Ashraf | Gantt Chart |
| Farva Talib | Deployment alternative platform +Reviewed SRS doc(class diagram + some sequence diagrams) |
| Muhammad Bilal Shahid | Gantt Chart, Software Process Selection 1 |

# Review checklist

Before submission of this deliverable, the team must perform an internal review. Each team member will review one or more sections of the deliverable.

| **Section** **Title** | **Reviewer Name(s)** |
| --- | --- |
| Salman Masood | Process selection first part |
| Muhammad Umair Mohsin | Ghantt chart |
| Muhammad Affan Ashraf | Software Process Selection Part 2 & 3 |
| Muhammad Bilal Shahid | Gantt Chart, Deployment Platform |
| Farva Talib | Development Environment Preparation |