**Software Process Selection and Project Plan**

**P03:MunasibMall.PK**

**team member names & ids**

|  |  |
| --- | --- |
| **Student ID** | **Name** |
| **22100168** | **Daniyal Mumtaz** |
| **22100134** | **Abdur Rehman Masood** |
| **22100199** | **Waqar Ul Haq Khatana** |
| **21100034** | **Waliullah Aitemad** |
| **21100161** | **Muhammad Muzammil Khan** |

|  |  |  |
| --- | --- | --- |
| **Content** | **Totals** | **Obtained** |
| Software Process Selection | 25 | 25 |
| Project context analysis | 10 | 10 |
| Gantt chart | 25 | 0 |
| Development environment preparation | 20 | 20 |
| Deployment platform | 10 | 10 |
| Who did what | 3 | 3 |
| Review checklist | 2 | 2 |
| Overall formatting/template | 5 | 5 |
| Late submission penalty | -20 |  |
| **Total** | **100** | **75** |
| Review | 20 |  |
| **Grand Total** |  |  |

**Table of Contents**

[1.](#_gjdgxs) Introduction 3

[2.](#_30j0zll) Software Process Selection 4

[3.](#_1fob9te) Gantt Chart 5

[4.](#_3znysh7) Risk Management 5

[4.1 Potential Risks and Mitigation Strategies 5](#_2et92p0)

[5.](#_tyjcwt) Development Environment Preparation 6

[6.](#_3dy6vkm) Deployment Platform 6

[7.](#_1t3h5sf) Who Did What? 7

[8.](#_4d34og8) Review checklist 7

# Introduction

Munasib Mall is a mobile e-commerce application. It will allow different merchants to have multiple dynamic web stores and customers will be able to buy from them on this platform. Merchants will have complete control over their store(listing products, removing products) and will be able to customize their store according to their wants. They will also be given some customizability in designing their store page. Customers will be able to order products of all the merchants/stores. Customers will be able to look for specific products across various stores. Orders will be processed by the merchant who will also handle the delivery.

Payment options like easy paisa or COD will be shown to the customer who can choose their preferred method when purchasing. The system aims to provide a platform that gives more autonomy and the ability to handle orders to the merchants. Reviews and ratings will help the customer choose the right merchant.

# Software Process Selection

# Waterfall Methodology

# 

# Waterfall is one of the earliest software development methodologies. It is a linear method where each phase of development is well documented right at the start.

# 

# Pros:

# The end product is crystal clear.

# Potential development issues are highlighted in the design phase and plans to tackle them are charted out accordingly.

# Since all the requirements including budget are specified before creating the actual product, there is very little financial surprises. The total expenses are almost the same as initially expected.

# Testing is easier.

# Cons:

# Since this is a linear methodology that requires all the requirements to be elicited before actually doing the coding, there is almost zero room for changes.

# It requires everything to be documented, hence there is a lot of paper work involved.

# The end product is only released once fully developed. Hence, takes a long time.

# Customers can only evaluate the complete end-product. It requires a lot of time, effort, and money to refactor the code to accommodate those changes.

# 

# Agile Methodology

# 

# Agile is an iterative software development methodology where requirements are discovered and the product is developed side-by-side.

# 

# Pros:

# Due to its iterative nature, this software development methodology is pretty flexible as changes can be accommodated as they arise.

# Early delivery is one of the core principles of Agile methodology. Product is released in increments, with each increment increasing the functionality of the product.

# This method does not require extensive documentation.

# The product is tested continuously which makes it easy to spot bugs.

# Changes according to customer’s feedback are accommodated in real-time, ensuring customer’s satisfaction.

# Cons:

# Since this method is based on adaptive planning and evolutionary development, it is difficult to estimate the total cost right from the beginning. Many times, the total final cost is higher than what it was expected to be.

# It requires teams to be strictly disciplined. The end-product is only as good as the collaboration between the team members.

# Agile methodology is a fast-paced development methodology, therefore requires experts. Having inexperienced developers will only slow the work down.

# 

# 

# Software Process for MunasibMall.pk

# 

# We will be using the Agile Methodology in our project. The main reason is that we are creating a mobile app version of an e-commerce store and it is hard to define all the requirements beforehand. We expect changes along the way that we might want to accommodate as they arise.

The potential loss because of bugs and defects is low because the product is tested continuously after increments.

Since agile is fast-paced, it requires developers with high experience, but the product is delivered in time.

The rate of requirements change is accommodated since the product is being developed in increments.

The team size can be from 3 to 9, but the team has to be dedicated and coordination is important.

Organizational culture is adaptive to change to a great extent since the team chosen is dedicated to work together and complete the product in increments.

There is less pressure to develop early releases, and it gets balanced because the product is being completed in increments and tested continuously.

The business staffs commitment to work extensively with the development team is moderate, as the development team will be giving deliverables continuously.

Some members of the developing team should have experience with similar systems. Since the team members will be coordinating with each other they can share their experiences with each other including those who are not familiar with the required systems.

There is high reuse of components in agile methodology, which saves time.

# Gantt Chart

Chart can be accessed using this link: [There is nothing in the Gantt chart. I accessed it at 9:30am on 20-Oct-2021]

<https://app.gantt.io/embedded-gantts/4d907f58-2325-41dc-a7f1-6e83b45ca038>

# Risk Management

## Potential Risks and Mitigation Strategies

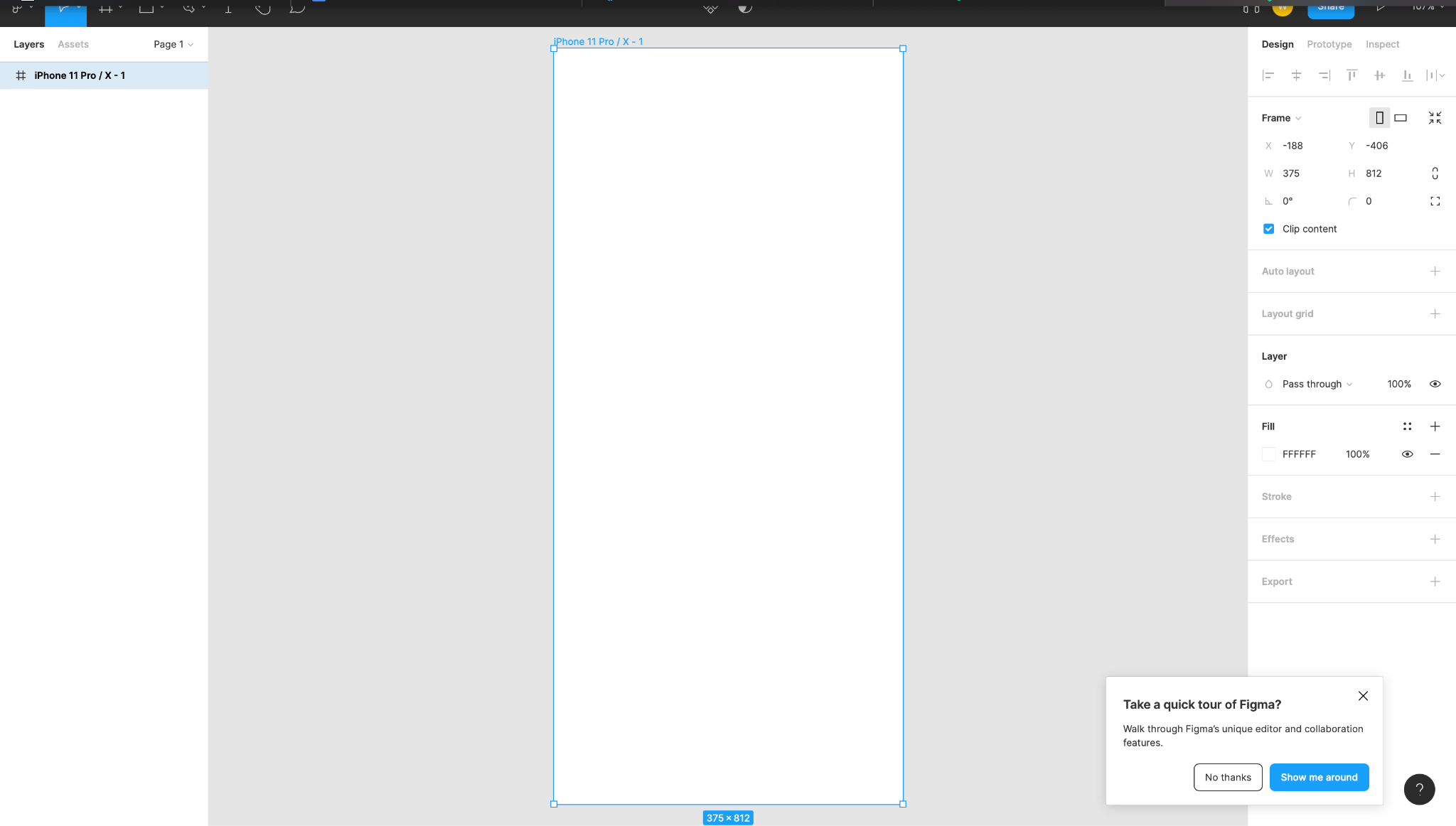
|  |  |  |
| --- | --- | --- |
| **Sr.** | **Risk Description** | **Mitigation Strategy** |
|  | Staff Illness | The work of the staff can overlap so that upon unavailability of a member, the group can reorganize and divide the work of the missing member in order to complete on time. |
|  | Product Competition | In order to increase the profits from the project and minimise the effect of competitors, better marketing strategies can be implemented, for example, more attractive discounts and concessions in the case of an e-commerce website. |
|  | Time Constraints of Deliverables | The team will manage time and track the progress of each deliverable by obtaining feedback from the members on a daily basis. |
| 4. | Requirement Change Risk | The customer will be informed that if the requirements change in the future, more time will be required to complete the project, as modifying the project requires time. |
| 5. | Economic Risks (Policy Changes of external organisations) | If there is an external policy change, for example, changes in the policies of payment applications (Easypaisa app) and methods that are recommended on the website. The customers will be informed of the updated policies and new policies for the payment option will be designed, for example if easypiasa app introduces a policy of delayed payments, the customers will be required to pay in advance of the product ordered. |
| 6. | Server Risk | If the server used for the project is low-powered and upon increase of website traffic, there are delays, the management will be suggested to purchase a high-powered server. |
| 7. | Skills Risk (Some members not familiar with the tools and technologies being used) | The tools and technologies will be discussed in detail with the staff and time and other group members with experience will be allocated to the members who are not familiar with the tools decided. |
| 8. | End-User Risk (end-users have problems while using the software) | The end-users will be surveyed on a regular basis so that if they are facing difficulties while using the application, the required changes will be made. |
| 9. | Operational Risk (If there is a problem in the working of the software e.g. unexpected crashing of software) | The problem will be identified by the team, and stakeholders will be informed of the unexpected problem. The team will work on solving the problem after giving a possible timeframe to the stakeholders. |
| 10. | Faults in Reusable Components of Software | The reusable components of the software will be kept in check by the team, and upon a fault occurring, the team will rectify it so that it can be reused when required. |

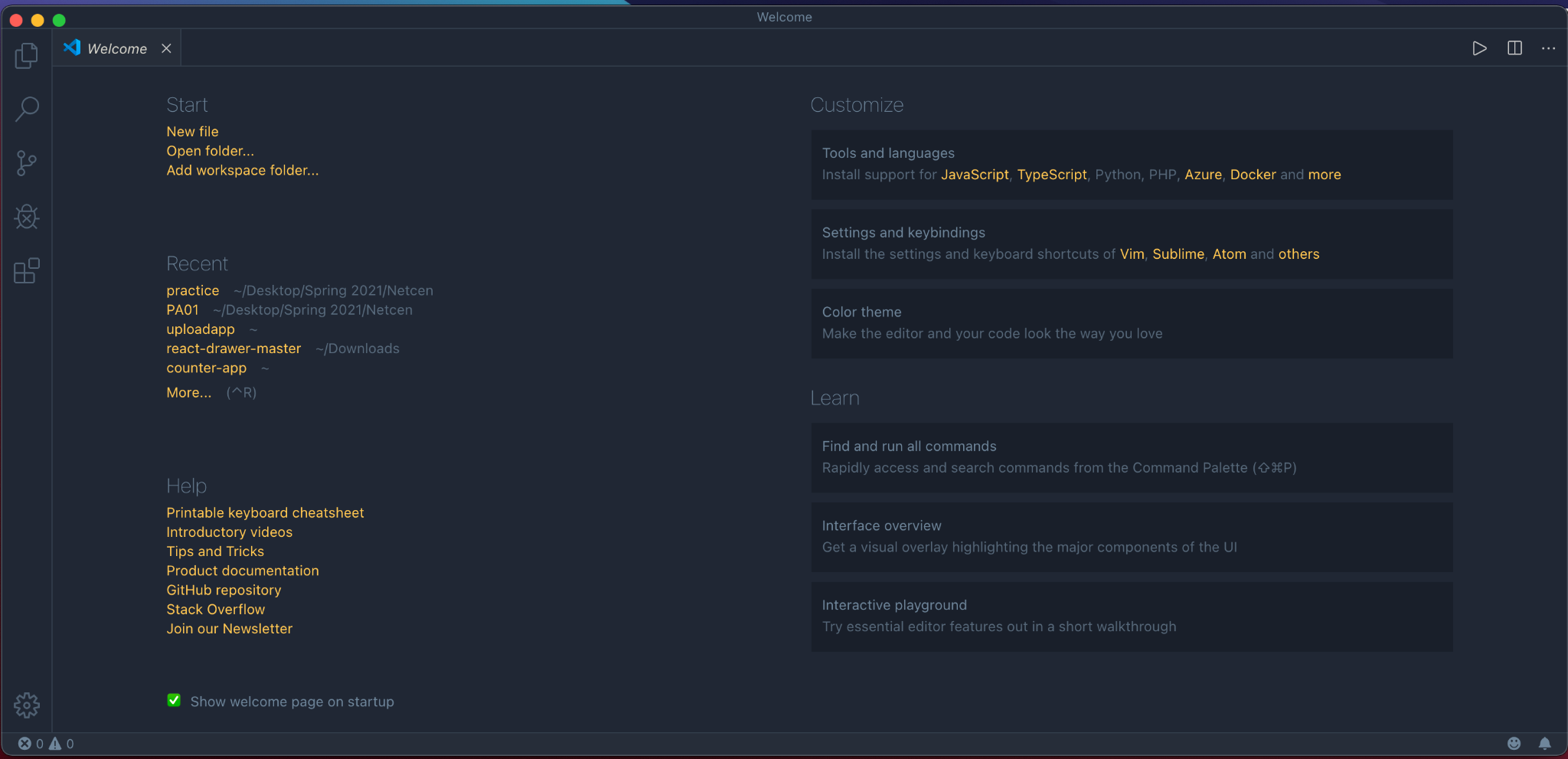
# Development Environment Preparation

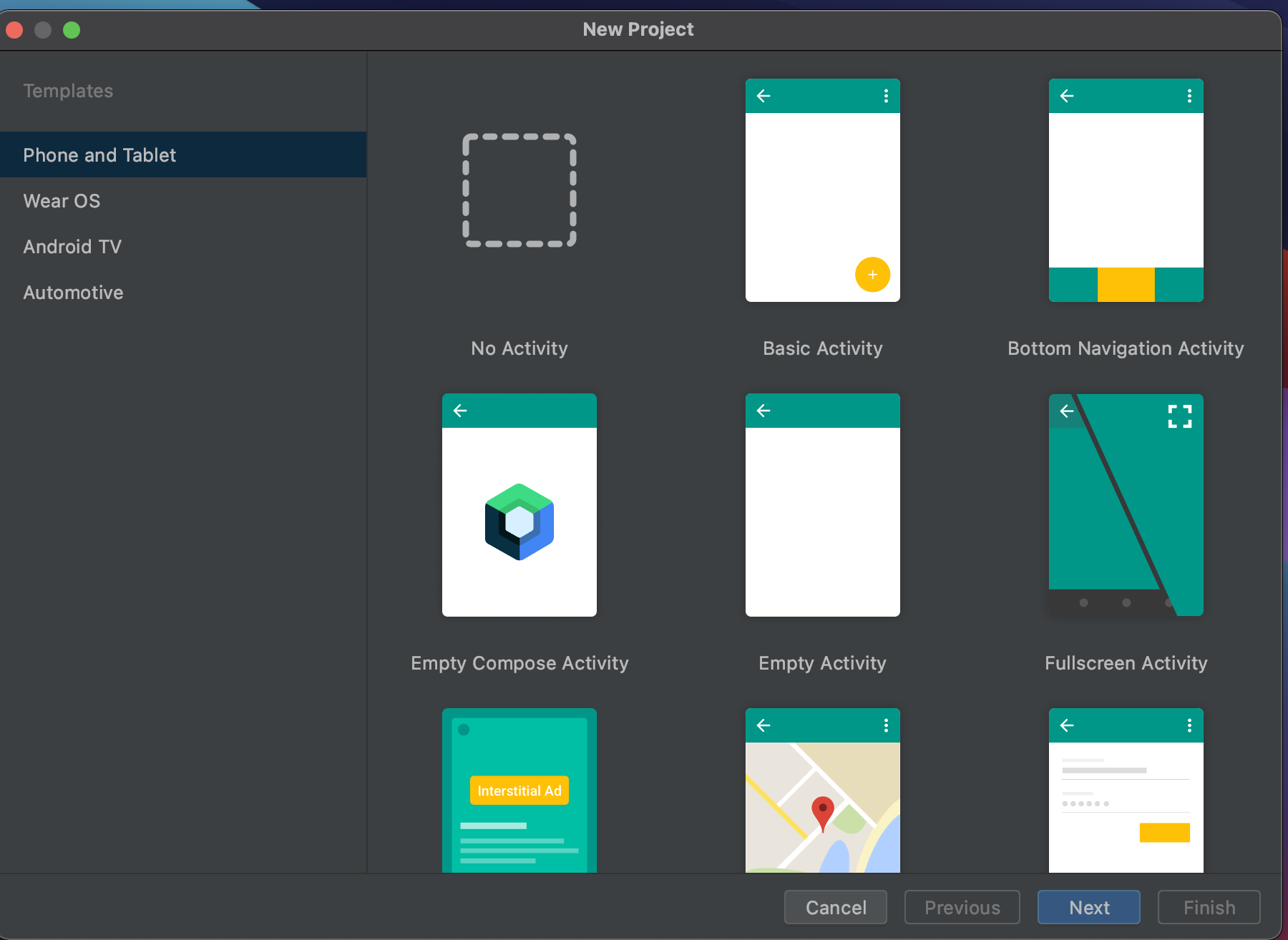
The primary tools and technologies we will be using are:

* Android studio as an IDE.
* Figma to design screens.
* MongoDB as a database.
* Nodejs and Expressjs to host.
* React Native to develop the front end.
* Android Virtual Device manager to test app with different screen sizes.
* VScode as an editor

We have also set up all downloadable tools. Some of the screenshots are attached.







# Deployment Platform

We will be using **HEROKU**, which has free plans available to deploy our application.

Heroku is a container-based cloud Platform as a Service (PaaS). We as developers will use Heroku to deploy, manage, and scale our apps. The platform is elegant, flexible, and easy to use, offering our team of developers the simplest path to getting our app to the market.

Heroku is fully managed, giving us the freedom to focus on our core product without the distraction of maintaining servers, hardware, or infrastructure. The Heroku experience provides services, tools, workflows, and polyglot support—all designed to enhance developer productivity.

Heroku’s free cloud services begin with the apps - apps that can be deployed to [dynos](https://www.heroku.com/dynos) - its lightweight Linux container that is at the heart of the [Heroku platform](https://www.heroku.com/platform). When we sign up with Heroku, we will automatically get a pool of free [dyno](https://devcenter.heroku.com/articles/dynos) hours to use for our app. When our app will run, it will consume dyno hours. When it idles (automatically, after 30 minutes of inactivity), or when we scale it down, the app will stop consuming dyno hours.

Source: https://www.heroku.com/

# Who Did What?

|  |  |
| --- | --- |
| **Name of the Team Member** | **Tasks done** |
| Daniyal Mumtaz | Introduction, Risk Management, Deployment Platform |
| Abdur Rehman Masood | Introduction, Risk Management, Deployment Platform |
| Waqar ul Haq Khatana | Software Process Selection |
| Waliullah Aitemad | Development Environment Preparation |
| Muhammad Muzammil Khan | Gantt Chart |

# Review checklist

|  |  |
| --- | --- |
| **Section** **Title** | **Reviewer Name(s)** |
| Deployment Platform | Daniyal Mumtaz |
| Software Process Selection | Abdur Rehman Masood |
| Risk Management | Waqar ul Haq Khatana |
| Development Environment | Waliullah Aitemad |
| Gantt Chart | Muzammil Khan |