

**CS 491 (A)**

**Software Process Selection and Project Plan**

**P04: Dairy Farm Management System**

|  |  |
| --- | --- |
| **Student ID** | **Name** |
| **22100125** | **Abdullah Saleem** |
| **22100070** | **Furqan Athar** |
| **22100072** | **Khawaja Junaid** |
| **22100209** | **Saad Qadeer** |

**Table of Contents**

[Introduction](#_heading=h.gjdgxs) **2**

[Software Process Selection](#_heading=h.30j0zll) **3**

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

[Risk Management](#_heading=h.3znysh7) **6**

[Potential Risks and Mitigation Strategies](#_heading=h.2et92p0) 6

[Development Environment Preparation](#_heading=h.tyjcwt) **9**

[Deployment Platform](#_heading=h.3dy6vkm) **11**

[Who Did What?](#_heading=h.1t3h5sf) **12**

[Review checklist](#_heading=h.4d34og8) **12**

# Introduction

In attempts to modernize the dairy industry in Pakistan, which is one of the greatest producers and consumers of milk and other dairy products in the region, our proposed and developed system will present a solution to better manage the dairy farms and optimize sales and help in better record keeping.

Furthermore, multiple dairy farm owners will have the opportunity to set up their accounts on our management system, as we aim to develop a multi-tenant software, which will aid them in getting more customers and for the better management and record keeping of their farms. The system will allow the customer to keep track of daily milk production, sales, expenses and maintenance. Moreover, the system will also provide an interface for the customers of the dairy farm to check their delivery and monthly invoice.

Potential users of our management system include dairy farm owners (who wish to digitize their records and keep an updated track of their farms) and the customers who wish to order dairy products, be it for domestic or commercial use.

The main purpose of this product will be to provide multiple tenants a way to see various trends in their milk production, expenses and income streams and to make better decisions for the future.

# Software Process Selection

**Waterfall:**

* **Pros**
  + Simple, well defined and straight forward phases which are sequential; this helps in better and more efficient management of the project. In addition to this, the end goal becomes more predictable i.e., it is relatively easier to predict what the product will most likely look like because earlier phases have been completed.
  + Since each phase is well documented and the flow is sequential, each phase must be completed properly before moving on to the next phase. This allows for more efficient interphase flow of data.
  + There is more clarity about the project phases for the development team and easier to keep track for them.
* **Cons:**
  + This method can be troublesome for a project where the goal of the project is not clearly defined, and requirements are changing frequently.
  + Many businesses do not have the same requirement so updating makes it hard for the teams to manage.

**Agile:**

* **Pros:**
  + The Agile methodology offers more flexibility than the waterfall. Allows more room for improvisation even in the later phase, as you can move back and forth between phases.
  + The Agile methodology is much faster. Using the Agile methodology, the product can reach the market faster. This can be used to gain feedback from the users based on the minimum viable product, that feedback can be incorporated into the MVP and a more user centered product can be produced.
* **Cons:** 
  + Moving back and forth between phases excessively may cause the market launch of complex products to suffer and take up more time. A very good level of collaboration and communication is required for inter phase communication between the development team to keep up. At times it gets difficult to maintain these standards.
  + For most clients there is a need for a known deadline, but with agile methodology it is hard to predict when a deliverable will be due.

**Software Process for our Project:**

**Chosen process:** Agile(scrum) methodology

**Justification**

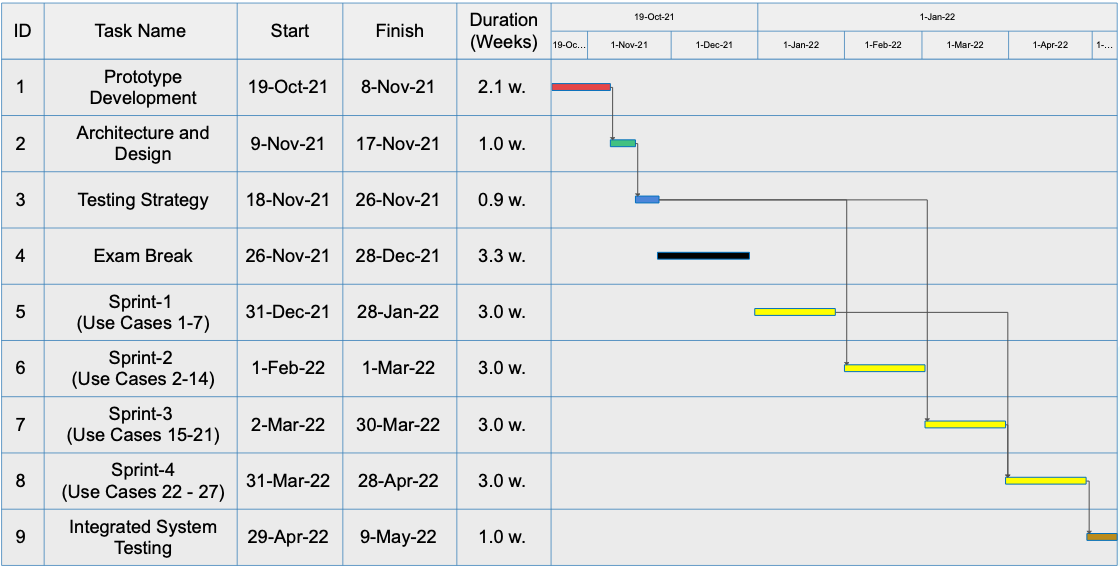
For the purpose of our project the agile scrum methodology, taking in considerations from the project contextual analysis. Agile methodology enables work to be done in continuous sprints, so the product is getting better iteratively, and bugs are taken care of in these sprints. It adjusts well to requirements change given our product is in its early stages of build so a prototype can be shown to the client that can help them better assess requirements.

A team size of 4 people would be more suitable to the agile methodology, since it enables more collaboration effectively and delivers a product in a short span of time. The agile methodology follows a flow in which there is a prototype product produced so there is no pressure to rush through phases such as in the waterfall model. An iterative process would mean that the product has many components that have been made in the earlier stages that can be used for later stages as well, there will be high availability of components.

# Gantt Chart

The Gantt Chart for our project is given below. The specific dates mentioned are tentative and subject to change depending on the workload and customer satisfaction in the previous task.

Currently it is planned that all the team will work on each task. The division of sub-tasks among the team members based on expertise will be decided during each task.



# Risk Management

## Potential Risks and Mitigation Strategies

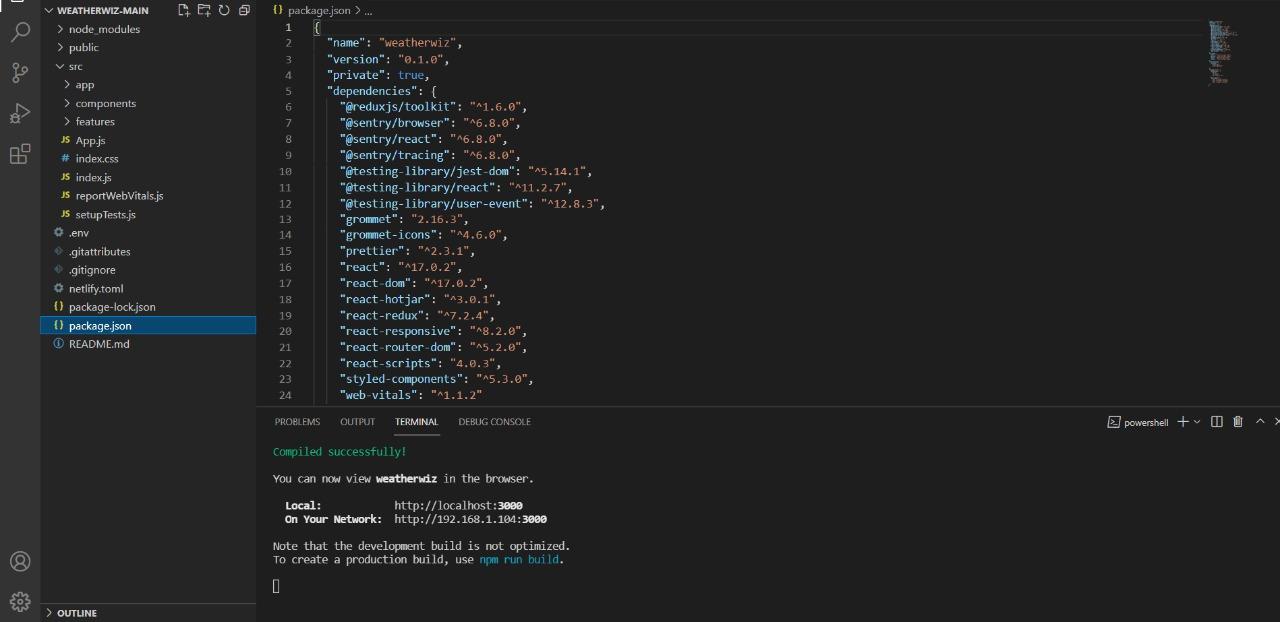
|  |  |  |
| --- | --- | --- |
| **Sr.** | **Risk Description** | **Mitigation Strategy** |
|  | Scope variations, when the project tends to overrun the agreed upon time. | Reduction, the scope changes can be reduced by taking in account the changes that are possible along the way with the client this includes agile methodology which gives time for the client to reflect, after small iterations. |
|  | Inadequate Productivity of the development team causing them to fall behind the work schedule and not meeting milestones/targets that were set initially. | One possible way to deal with such situations is to first set targets that are realistic and achievable within the given time frame. Another way can be for the development team members to keep check on each other and meet up at fixed periods. Furthermore, group member screening can also be done, to make sure that all the members are almost of the same calibre and to assign tasks based on the skill set of each member. To achieve more organization a manager role can be assigned to keep check and balance. |
|  | Imprecise estimation,  arises since a timeframe must be given to the client and other estimations. | reduction, Developers can consider the uncertainty involved in the estimations and certain allocation for estimations such as time frames. Reengineering process so more interaction with the people needed from the client’s organization. |
| 4 | Lack of a clear product vision. This is a potential risk that may occur if the development team and or client are unclear on what they want for the final product. This may also occur if they are not on the same page regarding the final product outcome. | A mitigation technique to deal with this may be to frequently ask questions from the client regarding the final product they want, meet up with the client, show the client the product developed in the early stages and what the development team is planning to do in the coming time. To make the development of the product roll swiftly there needs to be a clear final product vision about what you want to achieve. |
| 5 | Involvement with the end user, making softwares for client’s users(external user) is difficult since they are opposed to change. | Avoidance, test with users using a beta version of the software or have feedback mechanism that considers the opinions of the end user, conduct meetings with the end user to understand the design decisions. |
| 6 | Over simplistic or Over complicated design of the product. This problem may arise after the product has been developed. The client may find the product design to be over simplistic(the product does not offer an adequate number of features) or over complicated(the feature are so much that they overwhelm the user) | . The mitigation strategy to deal with this is to start with a minimum viable product, launch it, get feedback from the user and integrate the received feedback into the product. This can be even done repeatedly to further enhance the product. |
| 7 | Poor quality code, it would entail poor coding practices, released without testing and rushed. | Reduction, Following coding good practices, conduct unit testing for the developed components of the code. There can be a dedicated manager to monitor the code base and along with frequent code reviews. |
| 8 | Inadequate engagement/involvement of the client with the development team which may lead to a gap between what the client actually wants and what the developer team thinks the client wants. | One possible and obvious way of dealing with the situation, which may ultimately lead to what the client might consider a disaster, is to keep the client more involved/ engaged with the development team. The client should be able to clearly put forth their expectations of the product and the development team must be able to effectively and efficiently convey to the client what is achievable in what time span and at what cost. |
| 9 | Lack of adequate resources, developers can leave a project at a point in time, creating problems for the project’s completion. | Avoidance, Proper documentation of the code base can be maintained, or considering having new team members with a training period. |
| 10 | Disagreement between the client stakeholder and the development team. | The mitigation strategy for this risk can be more client involvement and keeping them informed. Both the clientele and development team need to be vocal and expressive about their concerns and ideas. Either of them should not feel that they are left our and that their ideas and opinions are not being taken under consideration. |

# Development Environment Preparation

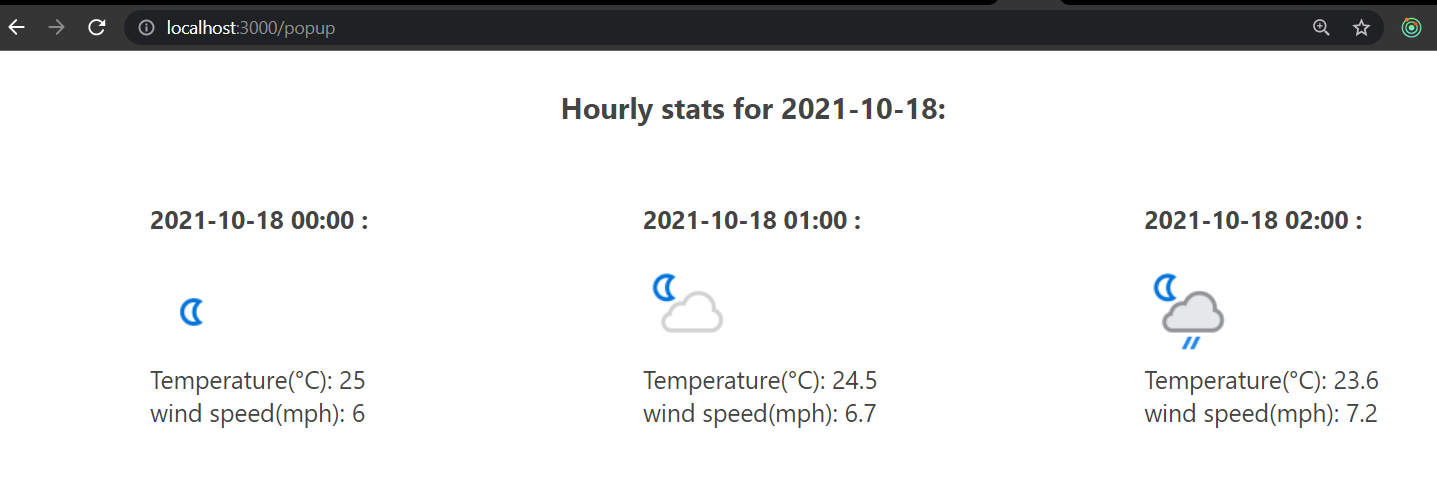
The tools and technologies which will be used in our system are as follows:

* [ReactJS](https://reactjs.org/) version react@16.14.0 for front end development of the webapp.
* Front end tools like [Redux sagas](https://redux-saga.js.org/) redux@4.0.4 for state management and API call management.
* [React-router](https://reactrouter.com/) react-router@5.2.0 a tool to navigate between components.
* [NodeJS](https://nodejs.org/en/) version v14.17.6 for backend deveoplment.
* [MongoDB](https://www.mongodb.com/) a noSQL serverless database MongoDB4.4
* The app would be deployed on [AWS](https://aws.amazon.com/codedeploy/) (2021 - latest version)

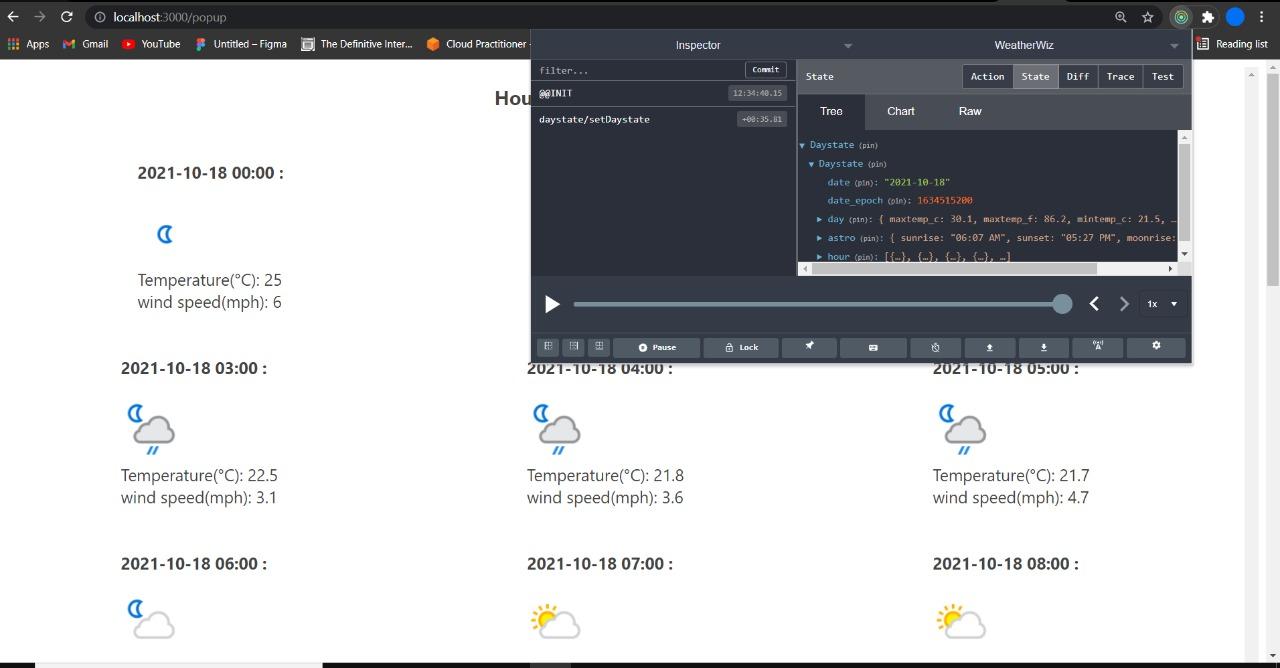
**ReactJS** app running , below are the packages installed which show ReactJS, redux, and router dom from VScode.



The path shows the change in components have taken via “**React-router**”

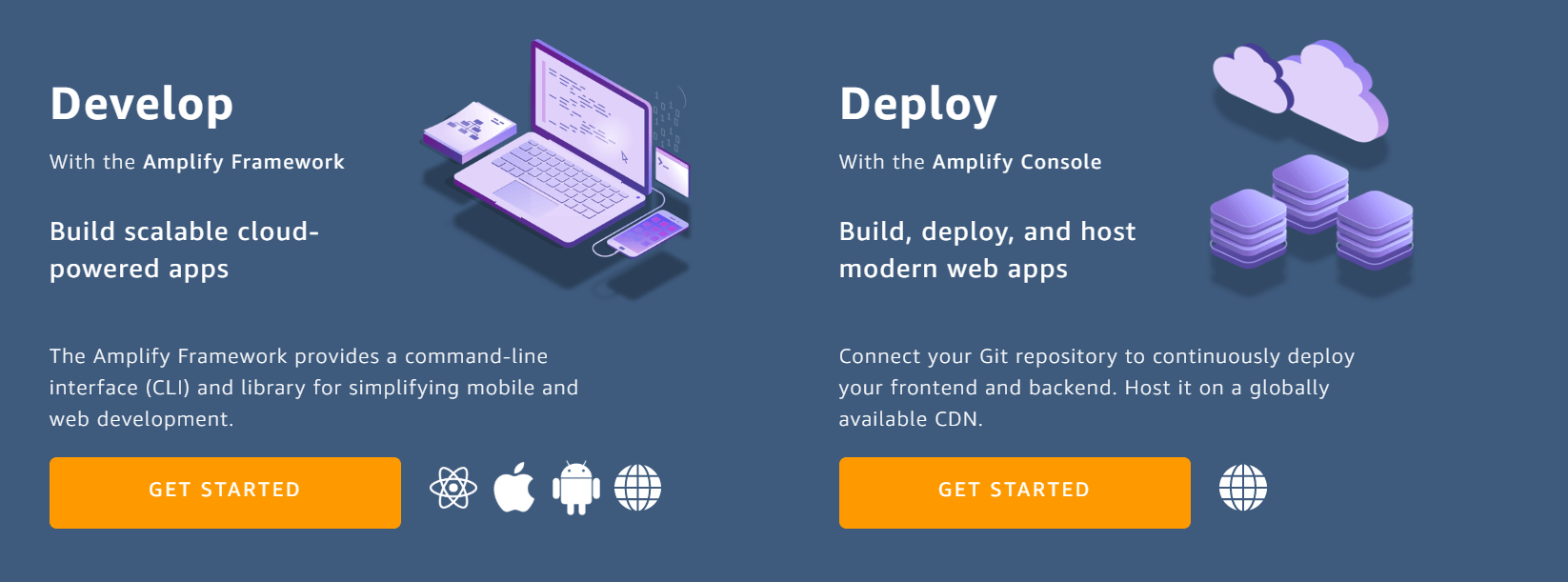


The **React redux** store responsible for state management is logging states fetched from the API call or updated within the component.



# Deployment Platform

**AWS amplify console** provides hosting facility for webapps. websites can be hosted from github repositories. It has a git based workflow that means it supports continuous deployment. You will not be charged for services in the free tier.



# Who Did What?

|  |  |
| --- | --- |
| **Name of the Team Member** | **Tasks done** |
| Furqan Athar | * Gantt Chart |
| Khawaja Junaid | * Development Environment Preparation * Deployment platform |
| Saad Qadeer | * Software Process Selection |
| Abdullah Saleem | * Gantt Chart |

# Review checklist

|  |  |
| --- | --- |
| **Section** **Title** | **Reviewer Name(s)** |
| Gantt Chart | Khawaja Junaid |
| Software Process Selection | Furqan Athar |
| Development Environment Selection | Abdullah Saleem |
| Development Platform | Saad Qadeer |