Project Planning Phase

Date	22 October 2022			
Team ID	PNT2022TMID17835			
Project Name	Personal Expense Tracker Application			
Maximum Marks	8 Marks			

Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint 1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	Balavignesh
		USN-2	As a user, I will receive confirmation email once I have registered for the application	1	High	Gokulnathan
	Login	USN-3	As a user, I can log into the application by entering email & password	1	High	Hariprakash
	Dashboard	USN-4	Logging in takes to the dashboard for the logged user.	`2	High	Harish

Bug fixes, routine checks and improvisation by everyone in the team *Intended bugs only

Workspace	USN-1	Workspace for personal expense tracking	2	High	Gokulnathan				
Charts	USN-2	Creating various graphs and statistics of customer's data	1	Medium	Hariprakash				
Connecting to IBM DB2	USN-3	Linking database with dashboard	2	High	Harish				
	USN-4	Making dashboard interactive with JS	2	High	Balavignesh				
	USN-1	Wrapping up the server side works of frontend	1	Medium	Hariprakash				
Watson Assistant	USN-2	Creating Chatbot for expense tracking and for clarifying user's query	1	Medium	Harish				
SendGrid	USN-3	Using SendGrid to send mail to the user about their expenses	1	Low	Balavignesh				
	USN-4	Integrating both frontend and backend	2	High	Gokulnathan				
Bug fixes, routine checks and improvisation by everyone in the team *Intended bugs only									
Docker	USN-1	Creating image of website using docker/	2	High	Harish				
Cloud Registry	USN-2	Uploading docker image to IBM Cloud registry	2	High	Balavignesh				
Kubernetes	USN-3	Create container using the docker image and hosting the site	2	High	Gokulnathan				
Exposing	USN-4	Exposing IP/Ports for the site	2	High	Hariprakash				
	Charts Connecting to IBM DB2 Watson Assistant SendGrid Bug fixes, ro Docker Cloud Registry Kubernetes	Charts Charts USN-2 USN-3 USN-4 USN-1 USN-1 Watson Assistant USN-2 Watson Assistant USN-3 USN-4 USN-4 USN-4 USN-1 USN-4 USN-1 USN-1	Charts USN-2 Creating various graphs and statistics of customer's data Linking database with dashboard USN-4 USN-4 Making dashboard interactive with JS USN-1 Wrapping up the server side works of frontend USN-2 Creating Chatbot for expense tracking and for clarifying user's query Watson Assistant USN-3 USN-3 USN-3 Using SendGrid to send mail to the user about their expenses Integrating both frontend and backend Bug fixes, routine checks and improvisation by everyone in the team *Intended Docker USN-1 Creating image of website using docker/ UsN-2 Uploading docker image to IBM Cloud registry Kubernetes USN-3 Create container using the docker image and hosting the site	Charts USN-2 Creating various graphs and statistics of customer's data 1 Connecting to IBM DB2 USN-3 Linking database with dashboard 2 USN-4 Making dashboard interactive with JS 2 USN-1 Wrapping up the server side works of frontend 1 Watson Assistant USN-2 Creating Chatbot for expense tracking and for clarifying user's query 1 SendGrid USN-3 USN-3 Using SendGrid to send mail to the user about their expenses 1 Bug fixes, routine checks and improvisation by everyone in the team *Intended bugs on Docker USN-1 Creating image of website using docker/ 2 Cloud Registry USN-2 Uploading docker image to IBM Cloud registry 2 Kubernetes USN-3 Create container using the docker image and hosting the site 2	Charts USN-2 Creating various graphs and statistics of customer's data 1 Medium Connecting to IBM DB2 USN-3 Linking database with dashboard 2 High USN-4 Making dashboard interactive with JS 2 High USN-1 Wrapping up the server side works of frontend 1 Medium Watson Assistant USN-2 Creating Chatbot for expense tracking and for clarifying user's query 1 Medium SendGrid USN-3 Using SendGrid to send mail to the user about their expenses 1 Low USN-4 Integrating both frontend and backend 2 High Bug fixes, routine checks and improvisation by everyone in the team *Intended bugs only Creating image of website using docker/ 2 High Cloud Registry USN-2 Uploading docker image to IBM Cloud registry 2 High Kubernetes USN-3 Create container using the docker image and hosting the site 2 High				