**FSE Final SBA – WEBAPI and ANGULAR Project**

**Table content**

|  |  |
| --- | --- |
| **Modules** | **Page no** |
| **Overall project details** | **2** |
| **Angular UI**   * **Project**   + - **User**     - **Task**     - **View Task** | **3** |
| **Angular Testing** | **29** |
| **WebAPI** | **29** |
| **NUnit Testing** | **34** |
| **SQL Database** | **35** |
| **IIS hosting**   * **To create new port**   + **To host Angular UI**   + **To host Web API** | **36** |
| **GitHub Repository** | **42** |
| **Jenkins** | **44** |

This project is build with the angular UI ,webAPI to connect with the SQL database and perform various operations.

|  |  |
| --- | --- |
| **MODULES DONE IN THIS ASSIGNMENT** | **PATHS/TOOLS/VERSIONS** |
| Full assignment is present in IIHT VDI | C:\User\Administrator\Documents\MyFinalSBA |
| Front end | Angular,Bootstrap,CSS,HTML, JavaScript |
| Angular JS(UI) | Visual code |
| WebAPI | Visual studio 2017 |
| Database | SQL Server Management Studio 2012 |
| Repository | Git |
| Git Hub path | <https://github.com/LaaziyaBegum/ProjectTaskManagerUI>  <https://github.com/LaaziyaBegum/ProjectTaskManagerApi>  <https://github.com/LaaziyaBegum/SQL_Scripts> |
| NUnit | Visual studio 2017 |
| Jenkins | Localhost:8080  C:\Program Files (x86)\Jenkins\Workspace |
| Jasmine | 3.3.1 |
| IIS hosted application names | PMApiBL  PMUI |
| IIS hosted under | Default Website application |
| IIS hosted port | 5555 |

**ANGULAR UI APPLICATION**

This UI allows the user to add a project, new user details, task and view the existing details of the task and project. It also allows the user to edit the details of the project and task and also to update them with the latest details. With the help of this UI, user can sort the user details with their firstname,lastname,ID and sort the project using the startdate,enddate,projectname,completed status.

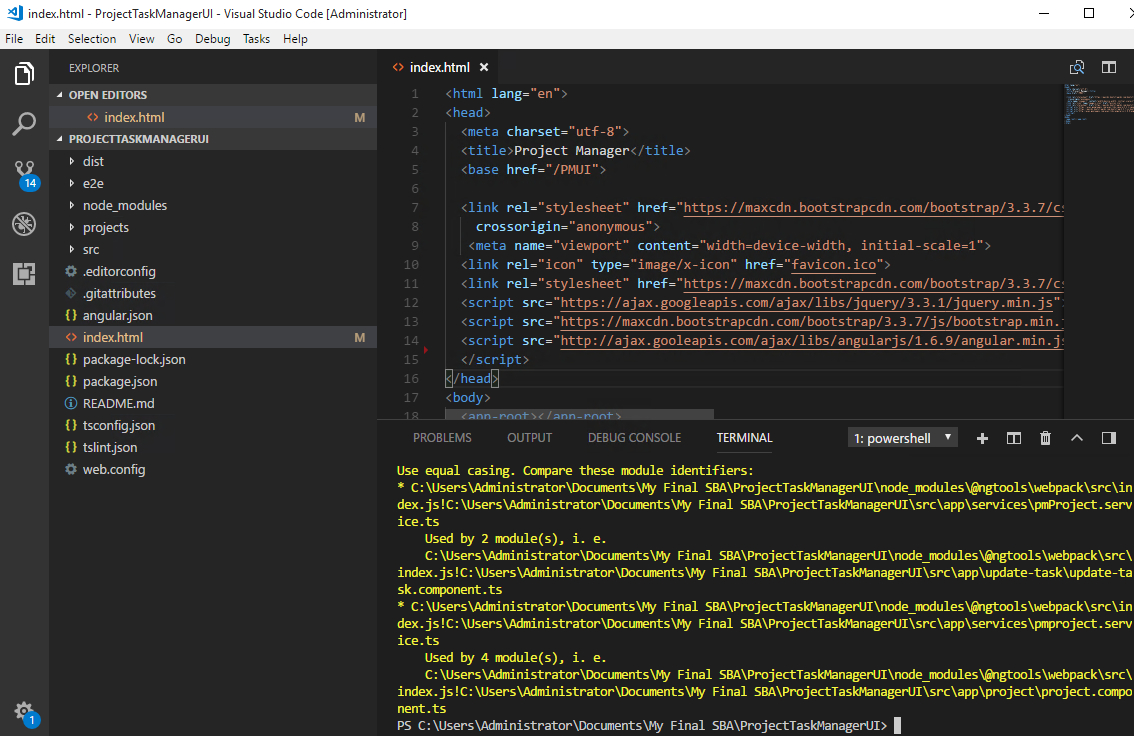
It also allows the user to select the existing project name, project manager name using the search button. User can end the project which is suspended or in progress to completed state.

This UI calls the localhost port , which there fore perform all the DB actions. Hence there fore attached all the reference screenshots below

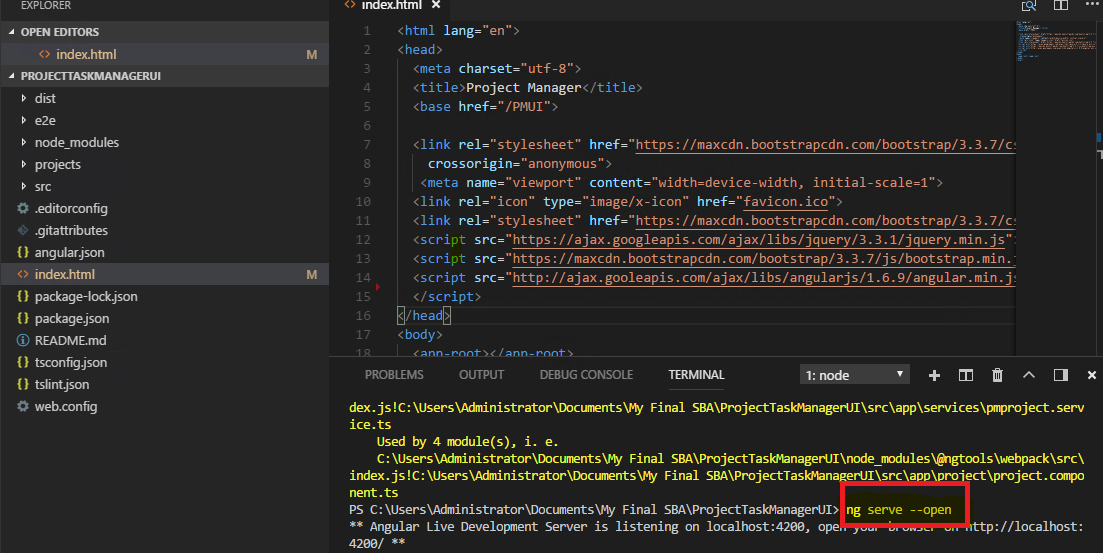
1. To start with the application, below are the steps involved in installing Angular CLI:

* npm install -g @angular/cli
* ng new ProjectTaskManagerUI
* cd ProjectTaskManagerUI
* ng build --prod
* ng serve –open

1. Below is the screen of Angular code compiled and executed successfully



1. Below is the command which is used to open the project in the browser.



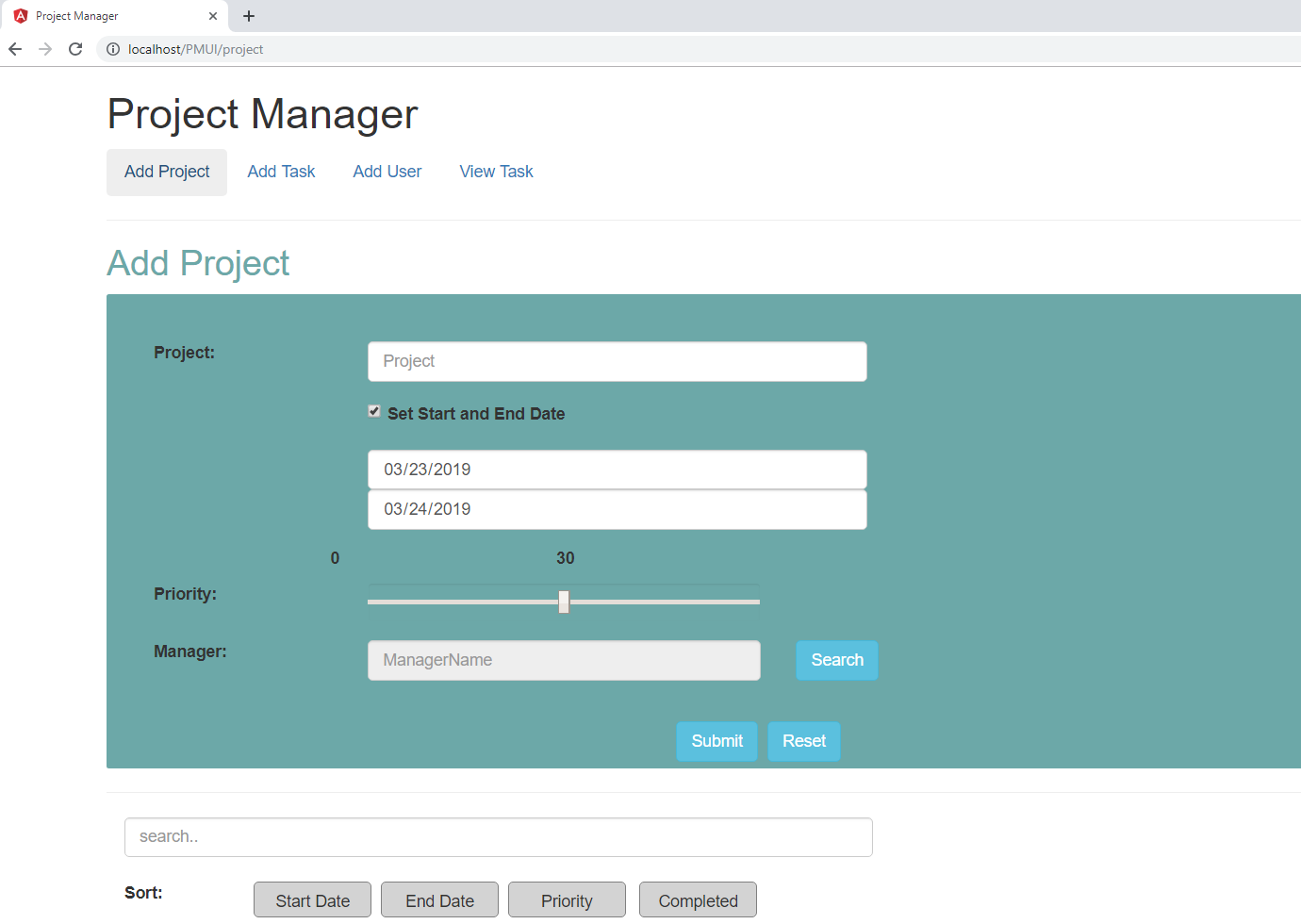
1. In output, UI has 4 menus for the user. Add project,Add User,Add Task, View project. Each menus and their actions is detailed below,

PROJECT

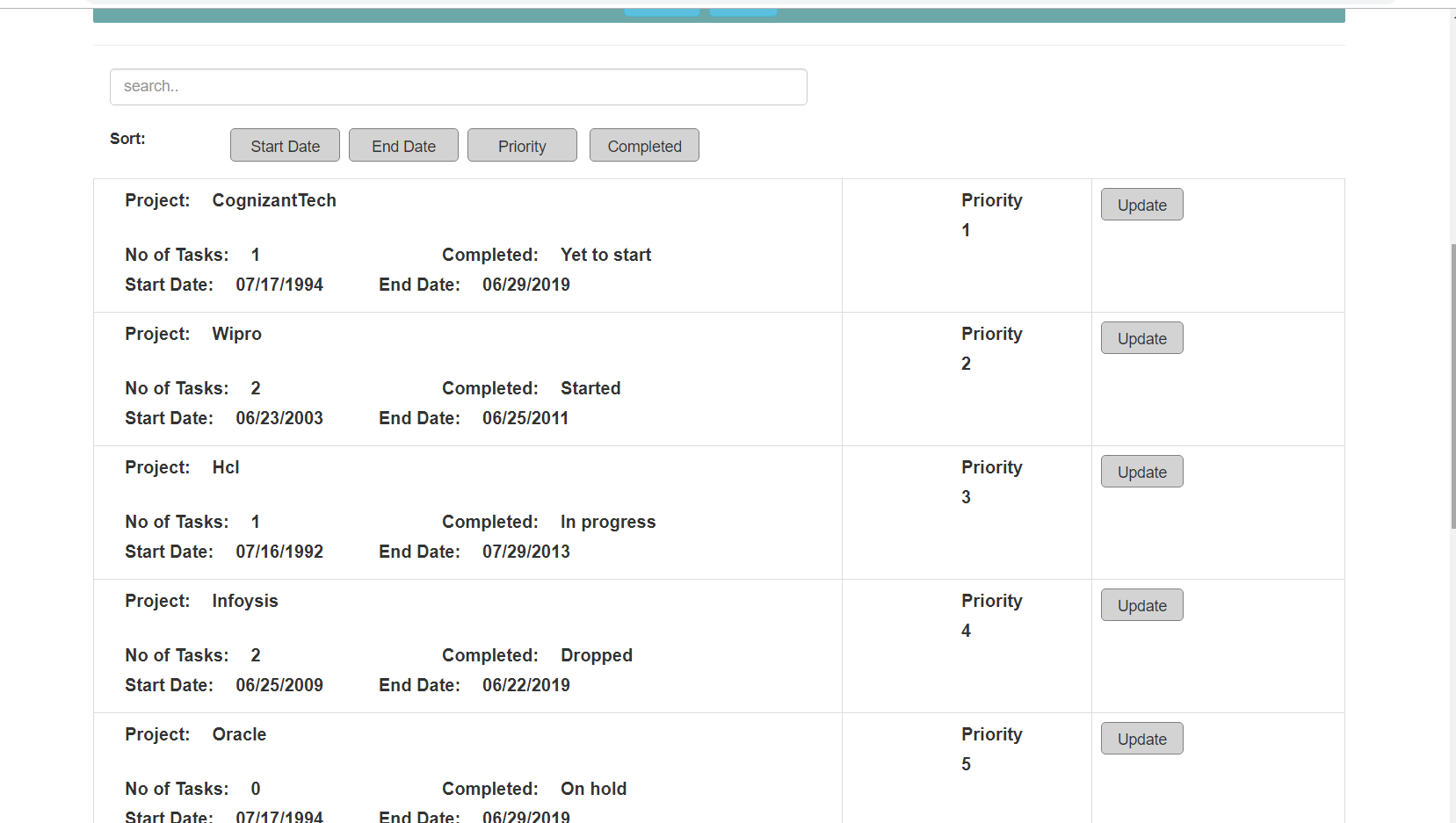
Project page allow the user to enter the project name,set the start date/end date of the project, to set the priority of the project and to select the manager who are already existing using the search button. Submit button allows to user to insert the data successfully and reset button used to clear the text fields so user can rewrite the data.

Apart from this, user can search for the specific project which is already existing. Sort option allows user to sort the existing project related details in start date/end date order , with priority and with the completed status.

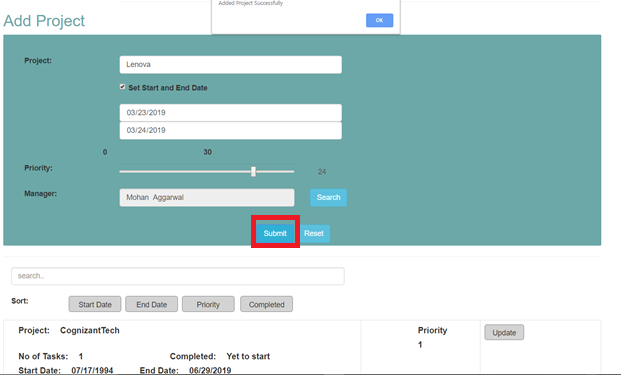
UI where user can add new project :



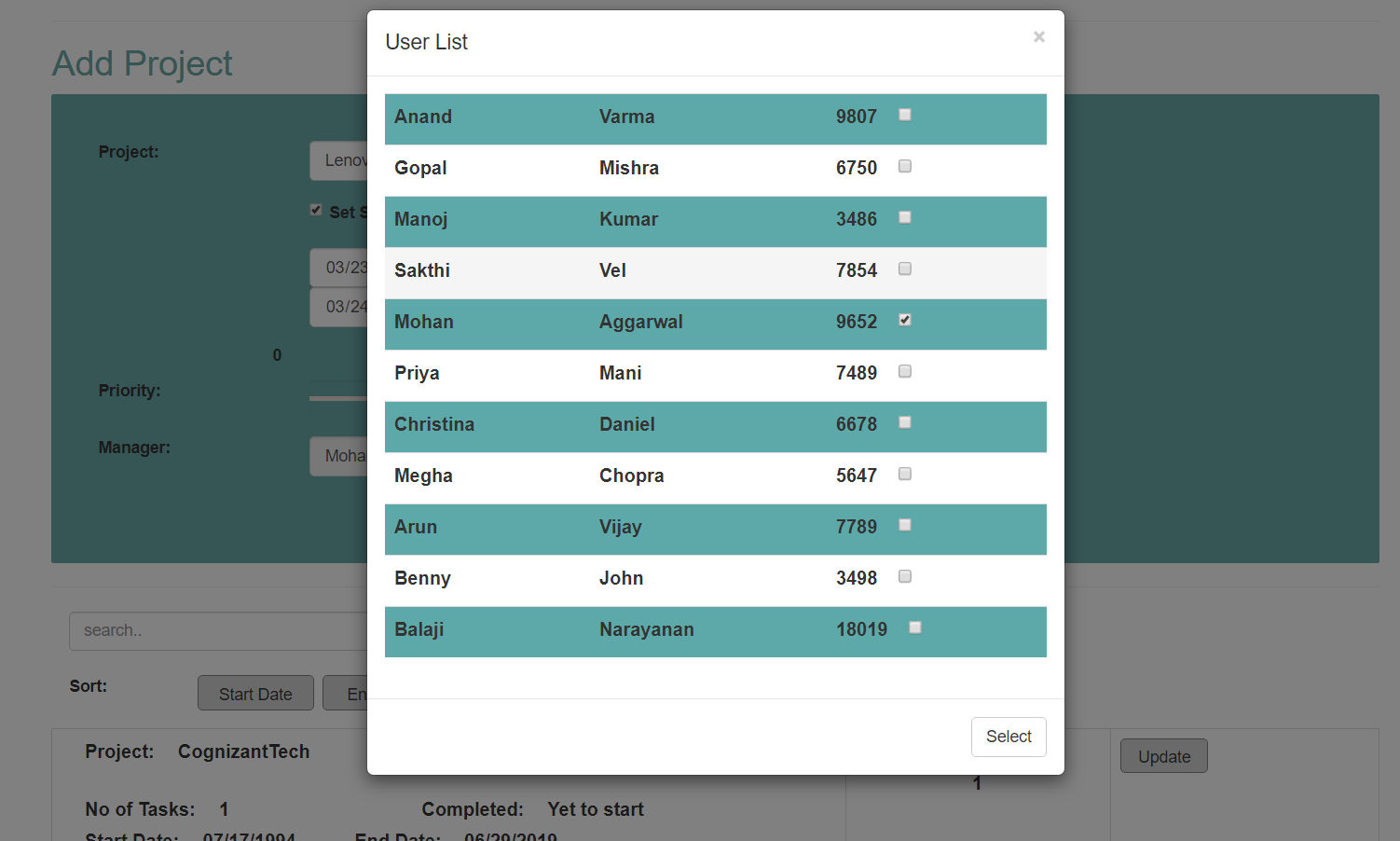
UI where user can view the existing project details:



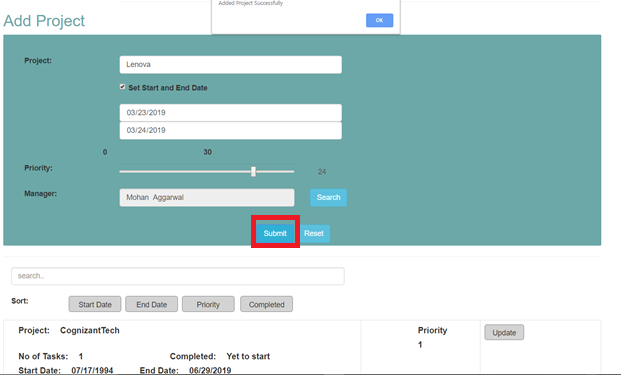
Sample Project has been added :



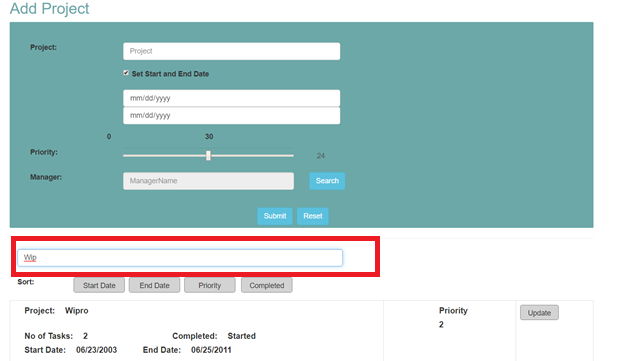
On clicking search button, searches for existing Manager details:



Submit button , allows the user to add the data:

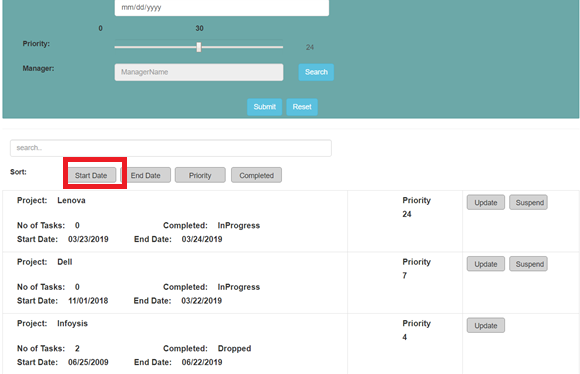


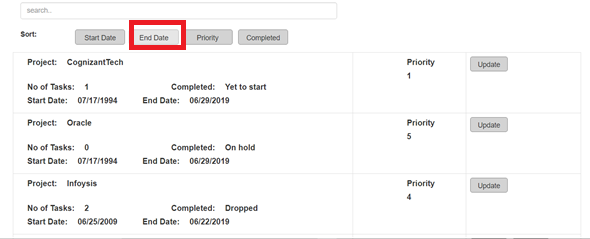
Search option for searching the project:



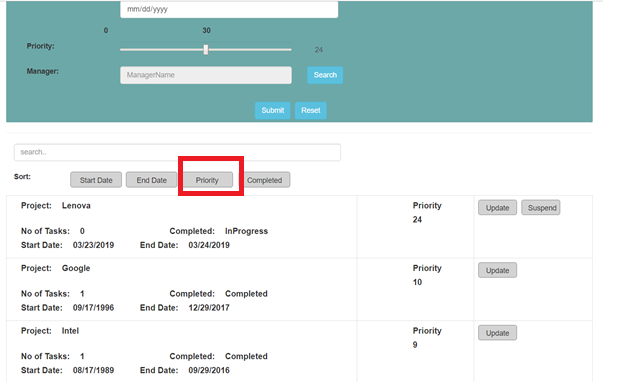
Sort options:

Sorting with the date:

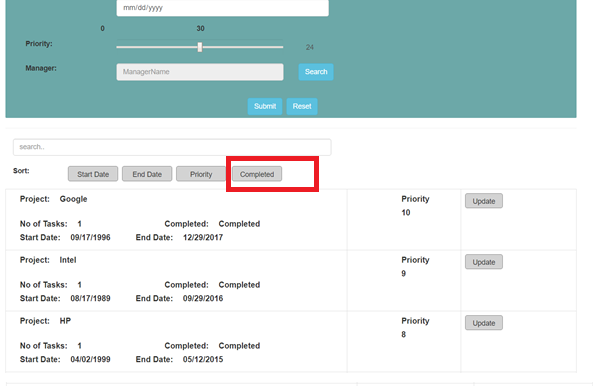




Sorting with the priority:

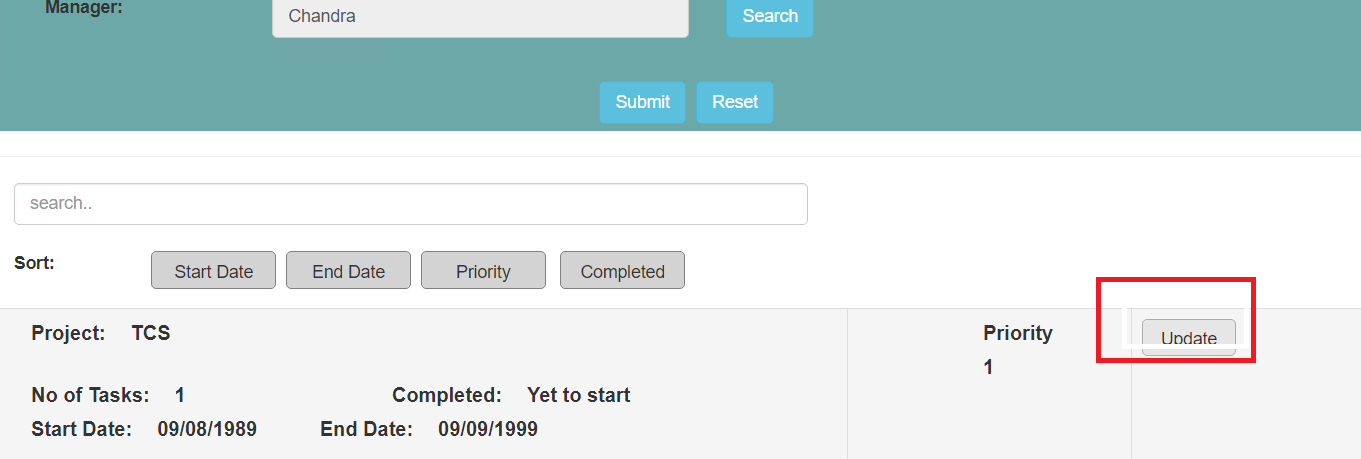


Sorting with the completed status:



On clicking update button:

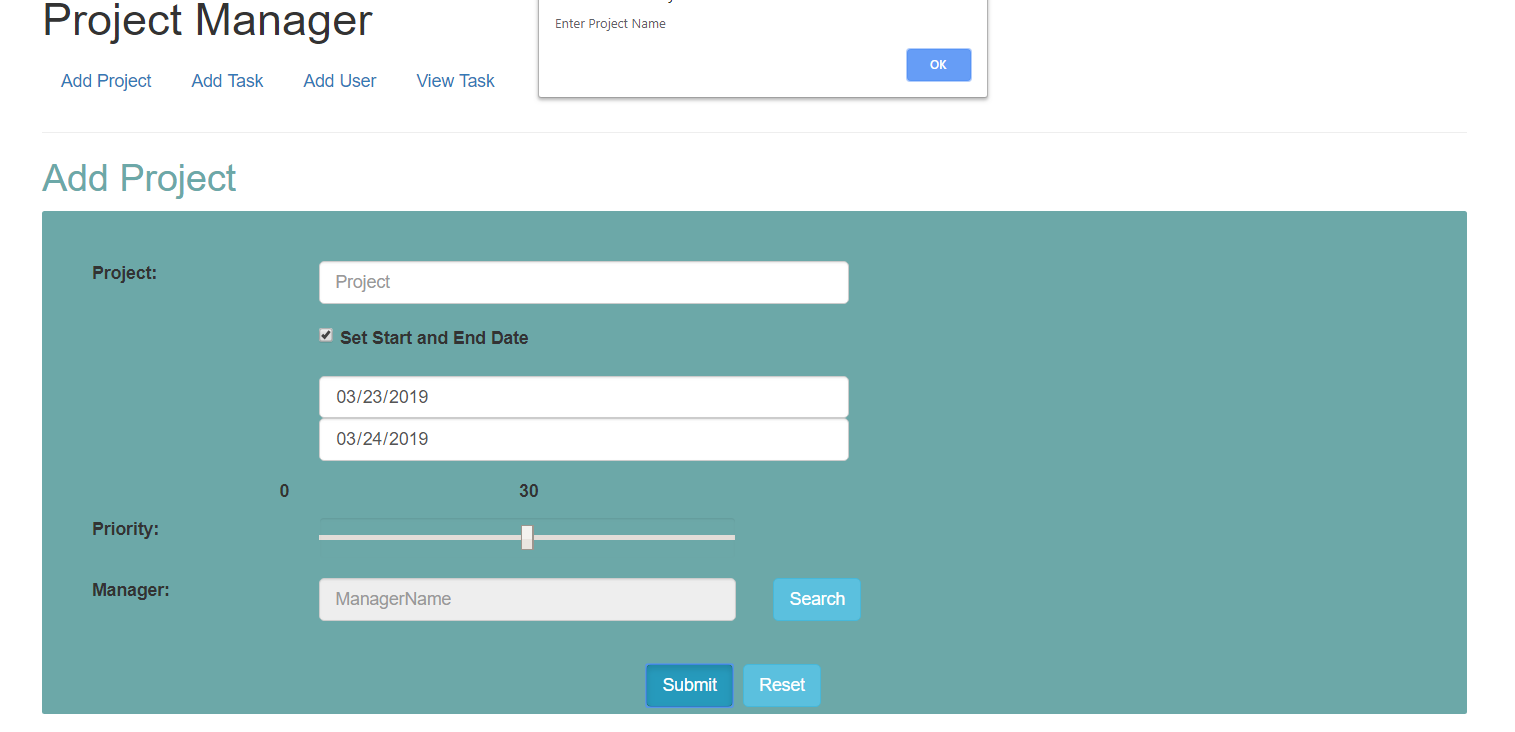
On clicking the update button, details are shown up in the add project screen and allows the user to edit the details and submit the data again.



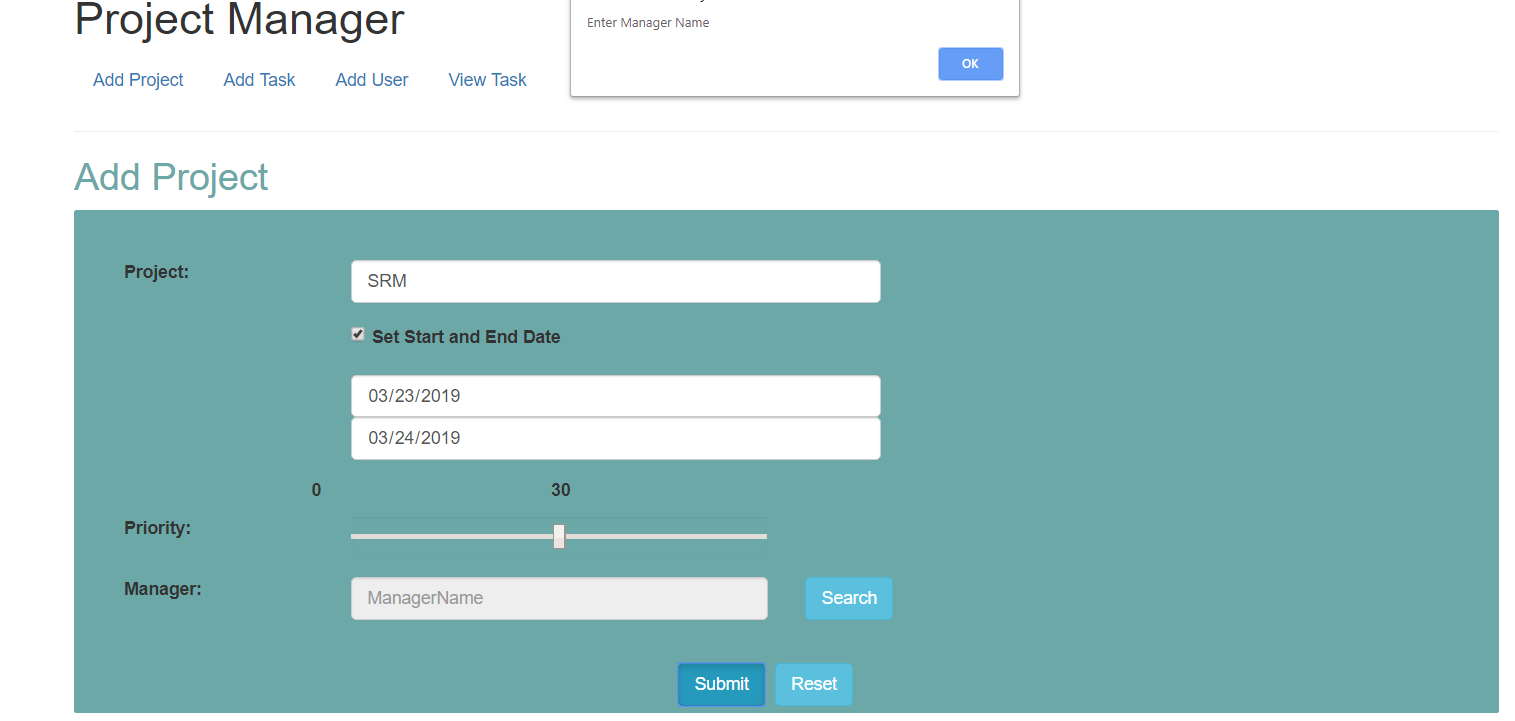
On Negative cases:

If user has not entered the project name, has not selected the dates(start date/end date) and has not selected the manager name, then the error pops up and will not allow the user to submit the project.

**Project name not entered:**



**Manager name not entered:**



TASK

Task page allows the user to enter the new task, but here project has been selected from already existing one which is present in the database. User needs to add the new task name,can set whether it has to be parent task, can set the priority for the task, has to select the parent task which is already existing using search button, needs to set start and end date for the task, needs to select the existing user name which is already present.

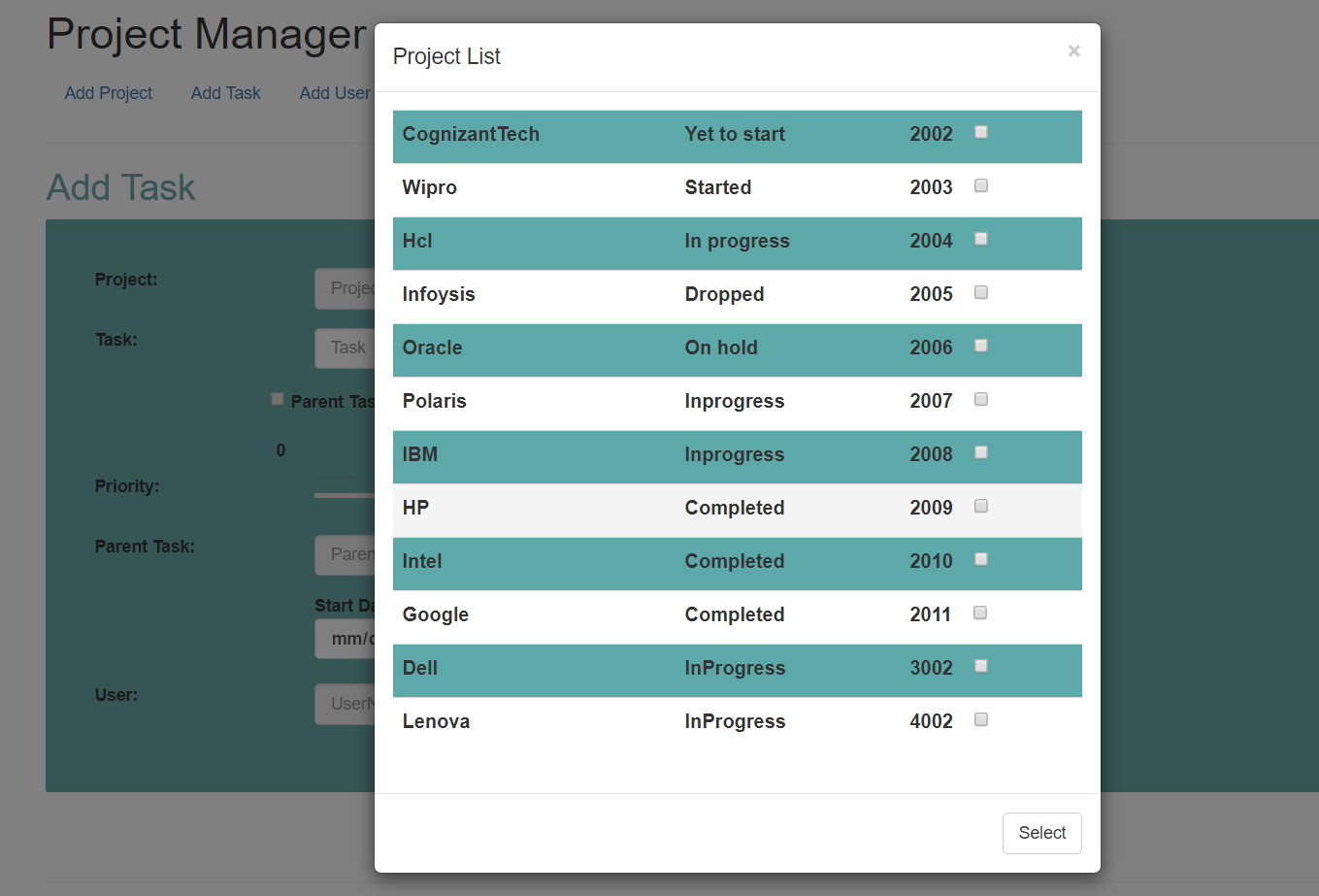
Submit button allows the user to insert the data and reset button allows the user to clear the text field so that user can re-enter their data.

UI where user can add new task :

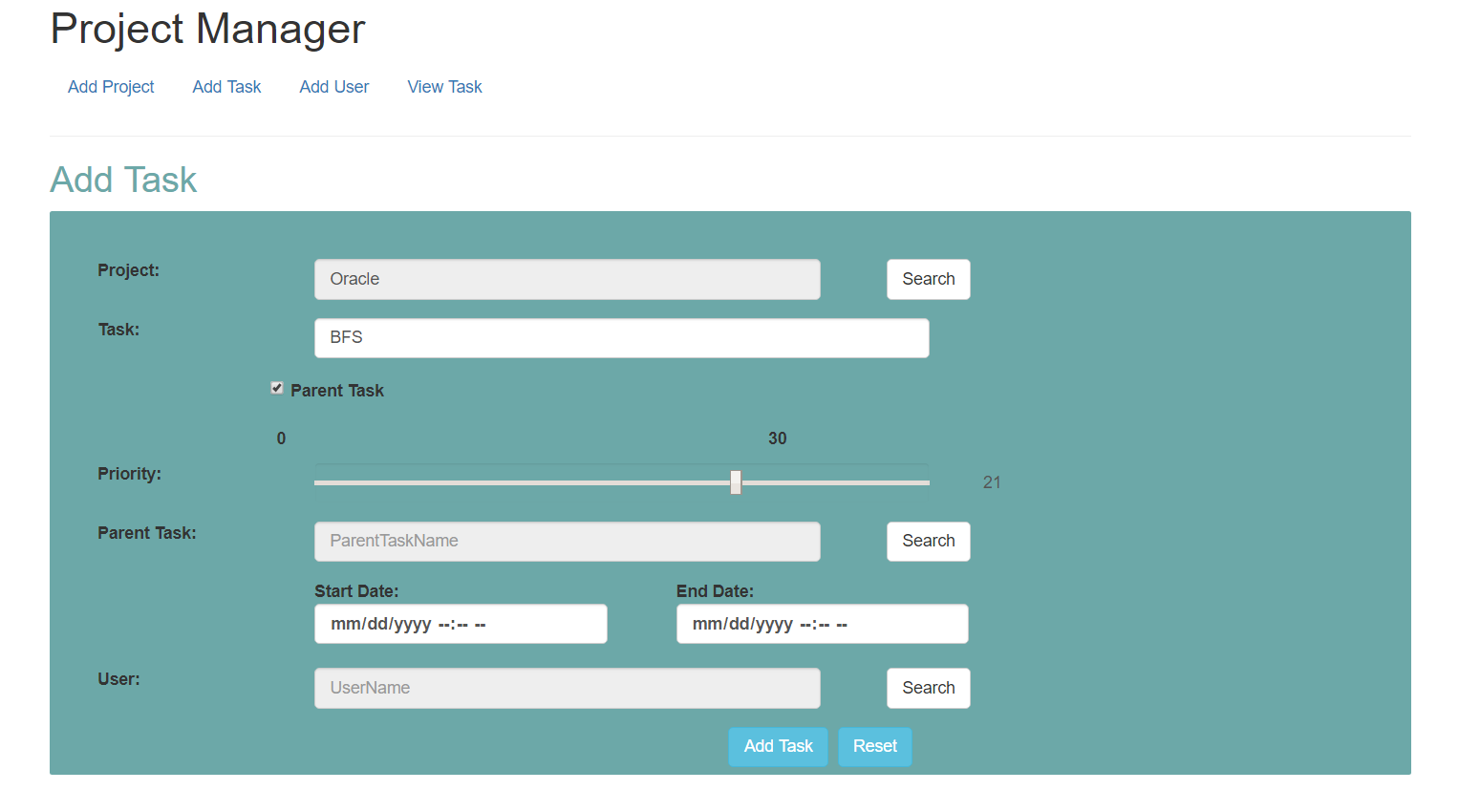


UI with task being added:

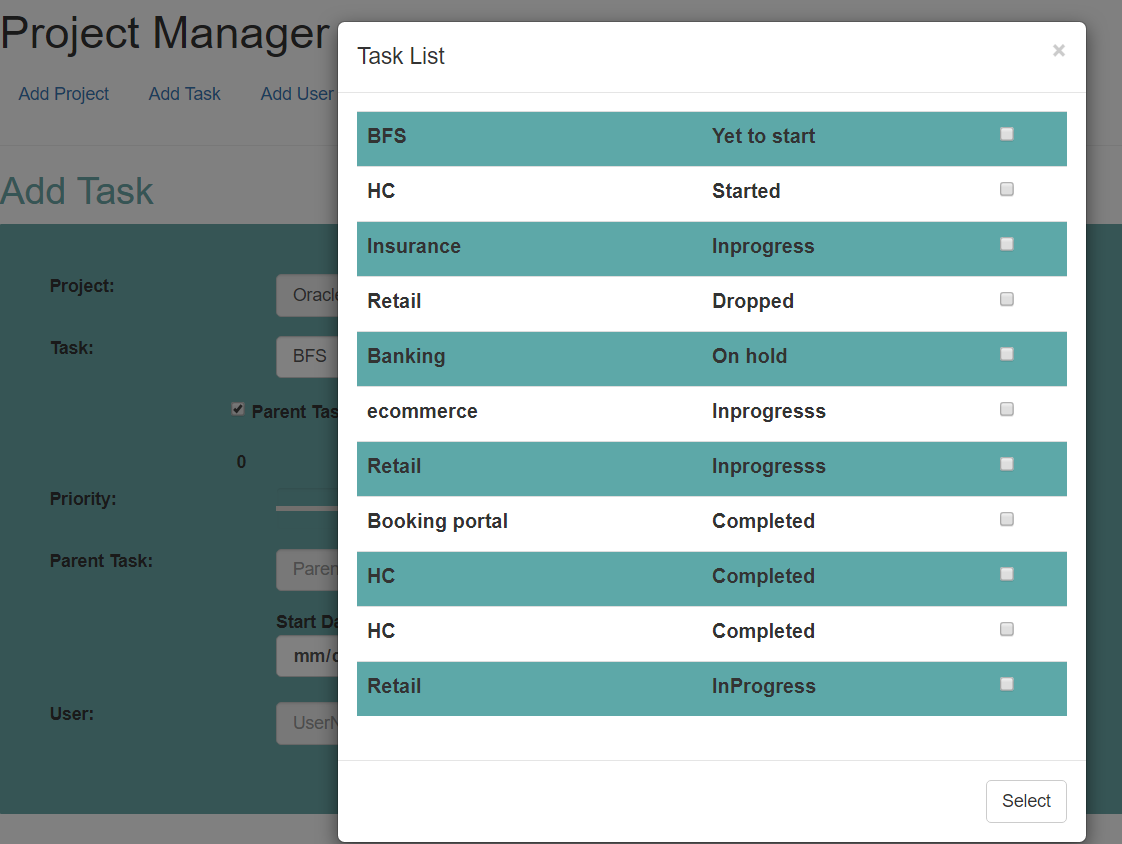
Search button allows the user to select the existing project.



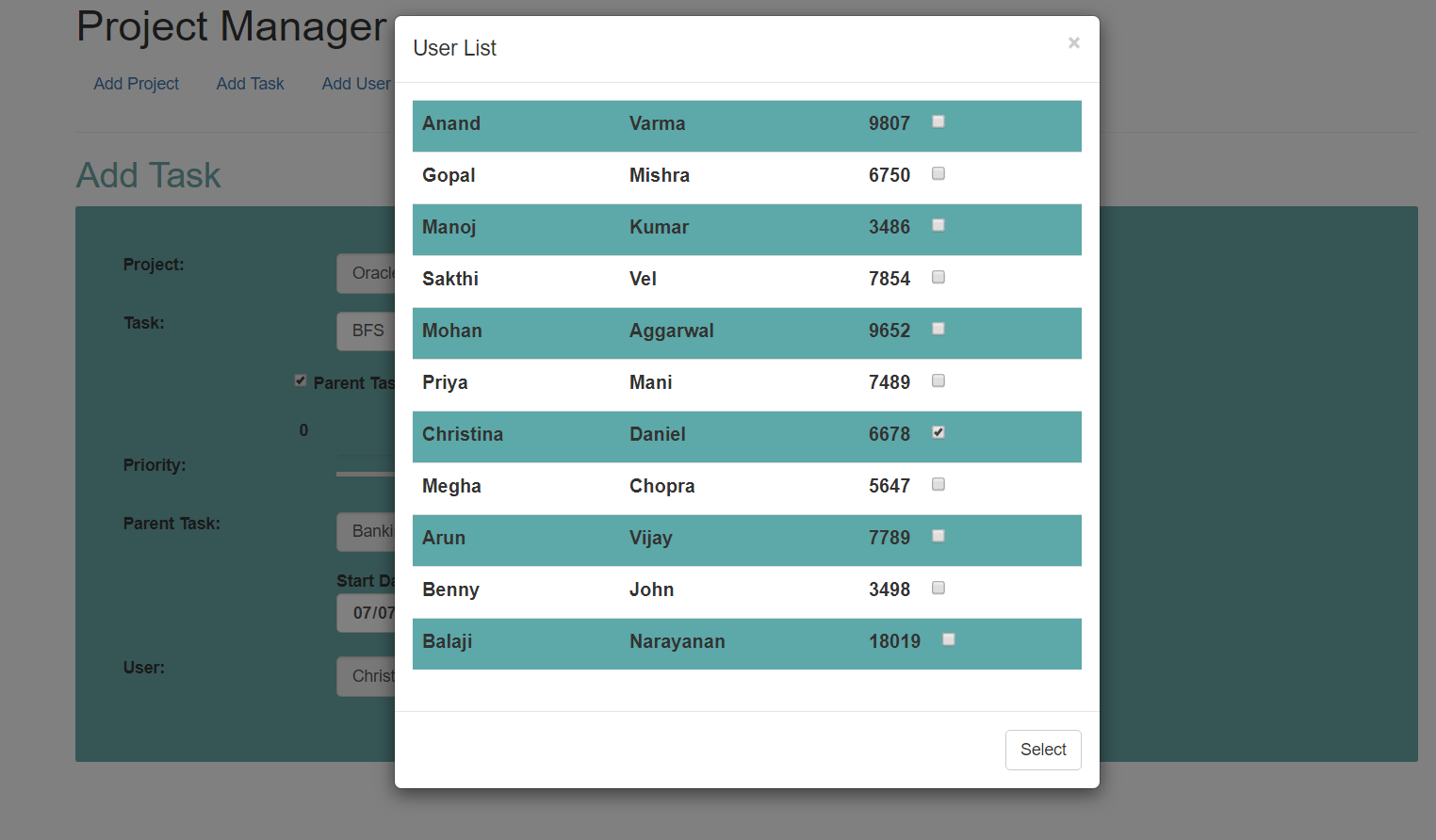
With the task,dates,priority being added:



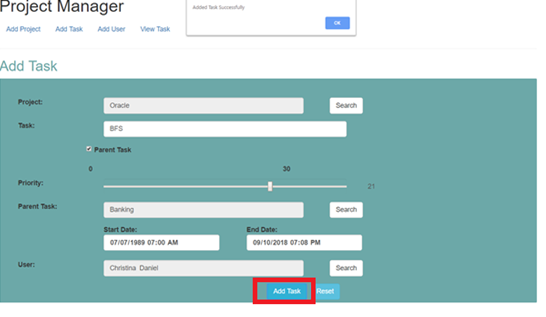
Search item to select parent task:



Search item to select the username:



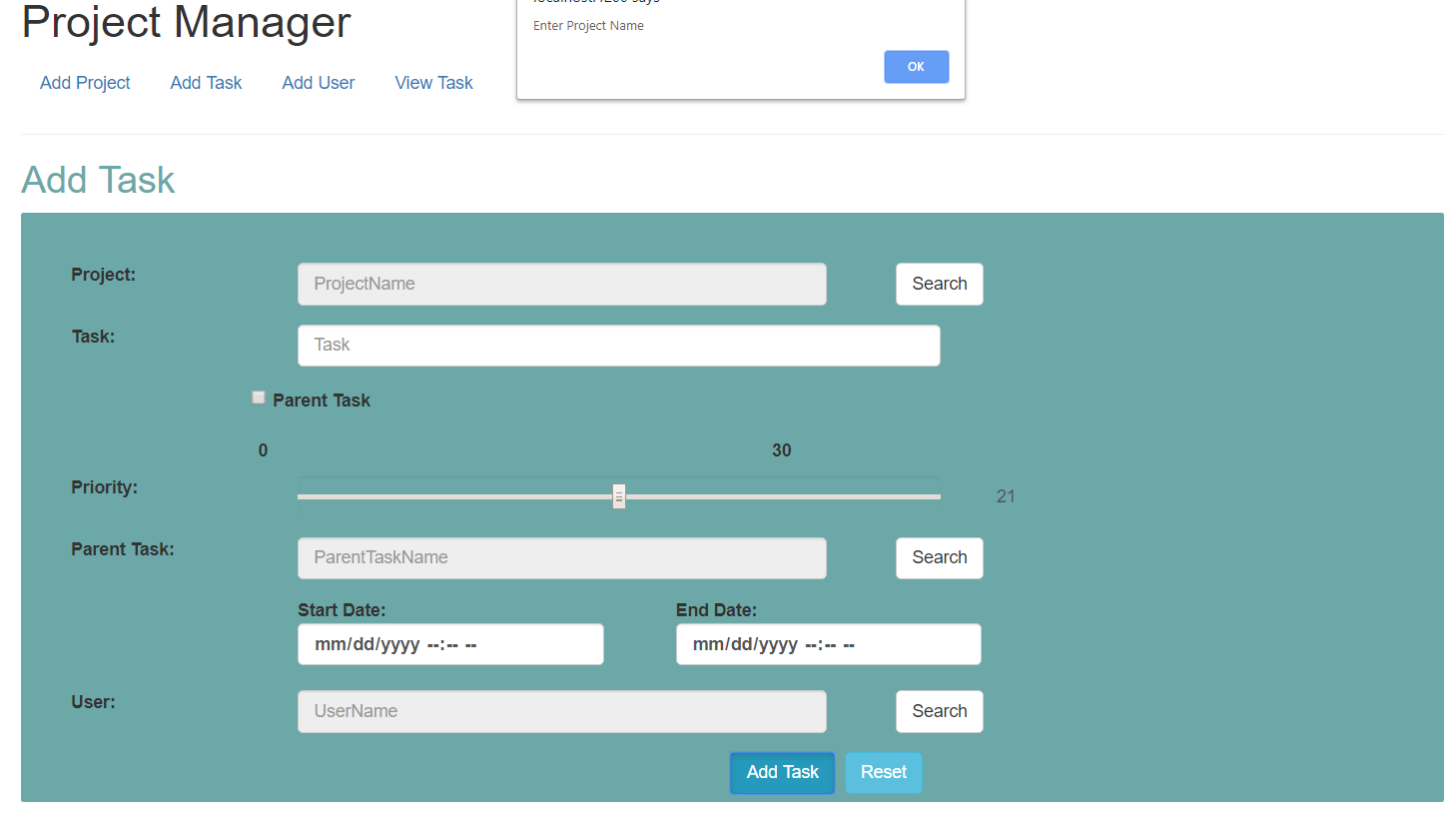
On clicking the submit button:



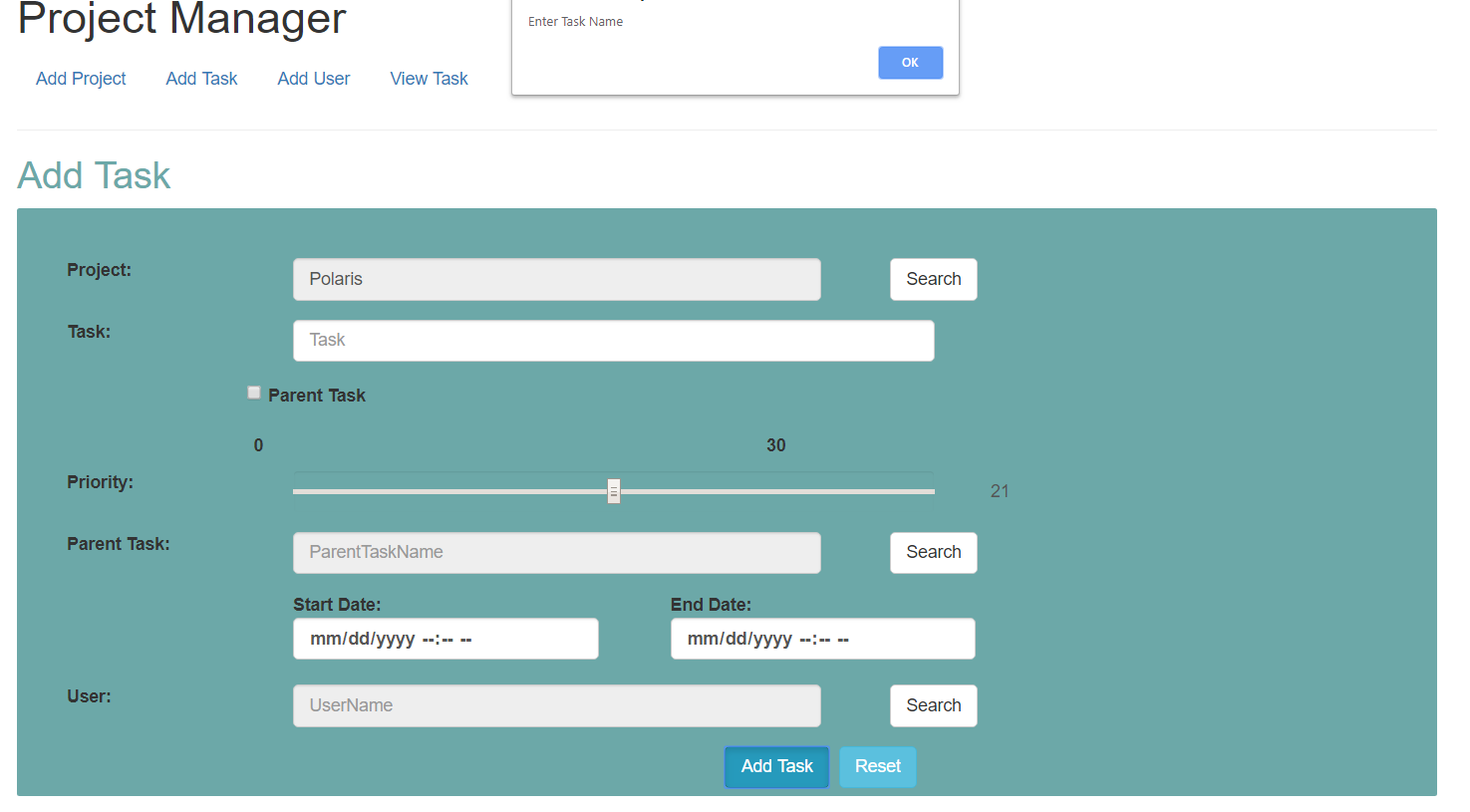
On Negative cases:

If user has not entered the project name, task name,has not selected the dates(start date/end date) ,has not selected the user name,and has not selected the parent task then the error pops up and will not allow the user to submit the project.

**When No project name:**



**When No Task Name:**



**When No Date:**



**When No User Name:**



**USER**

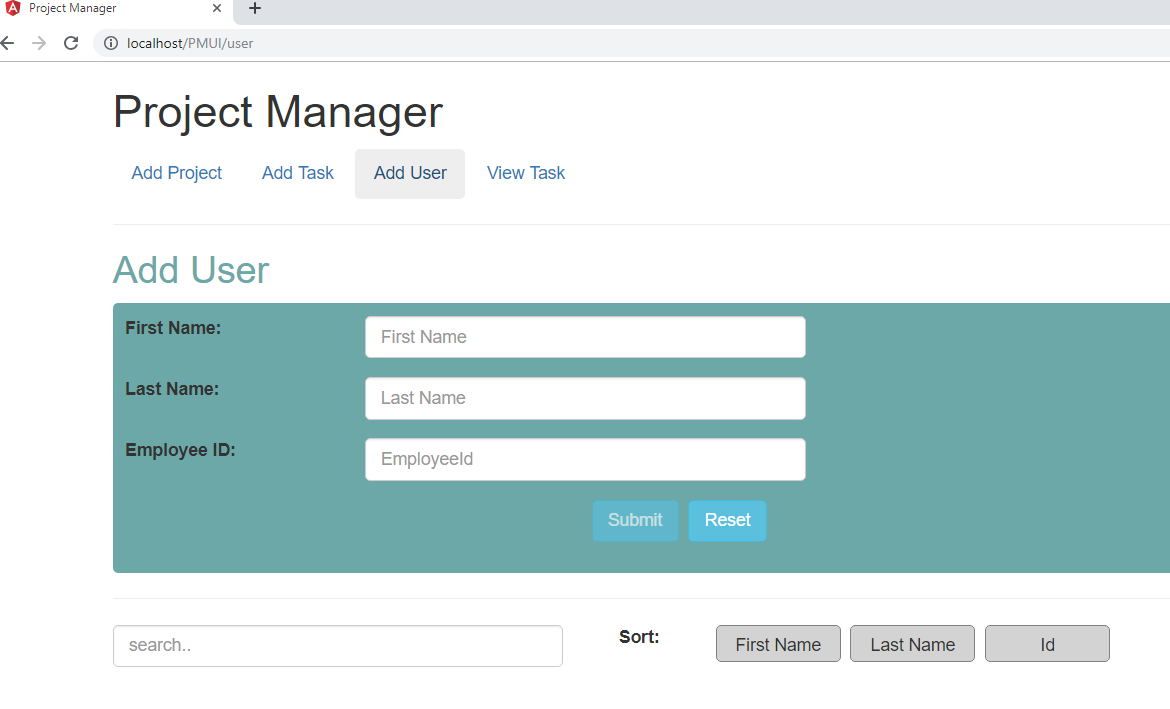
In user page, user can add their firstname, lastname and their employee id. Here there is no negative validation like popping of the error when any field is missed. Only on the presence of all the fields, submit button will be enabled. Submit button is used to insert the record and reset button is used to clear the record.

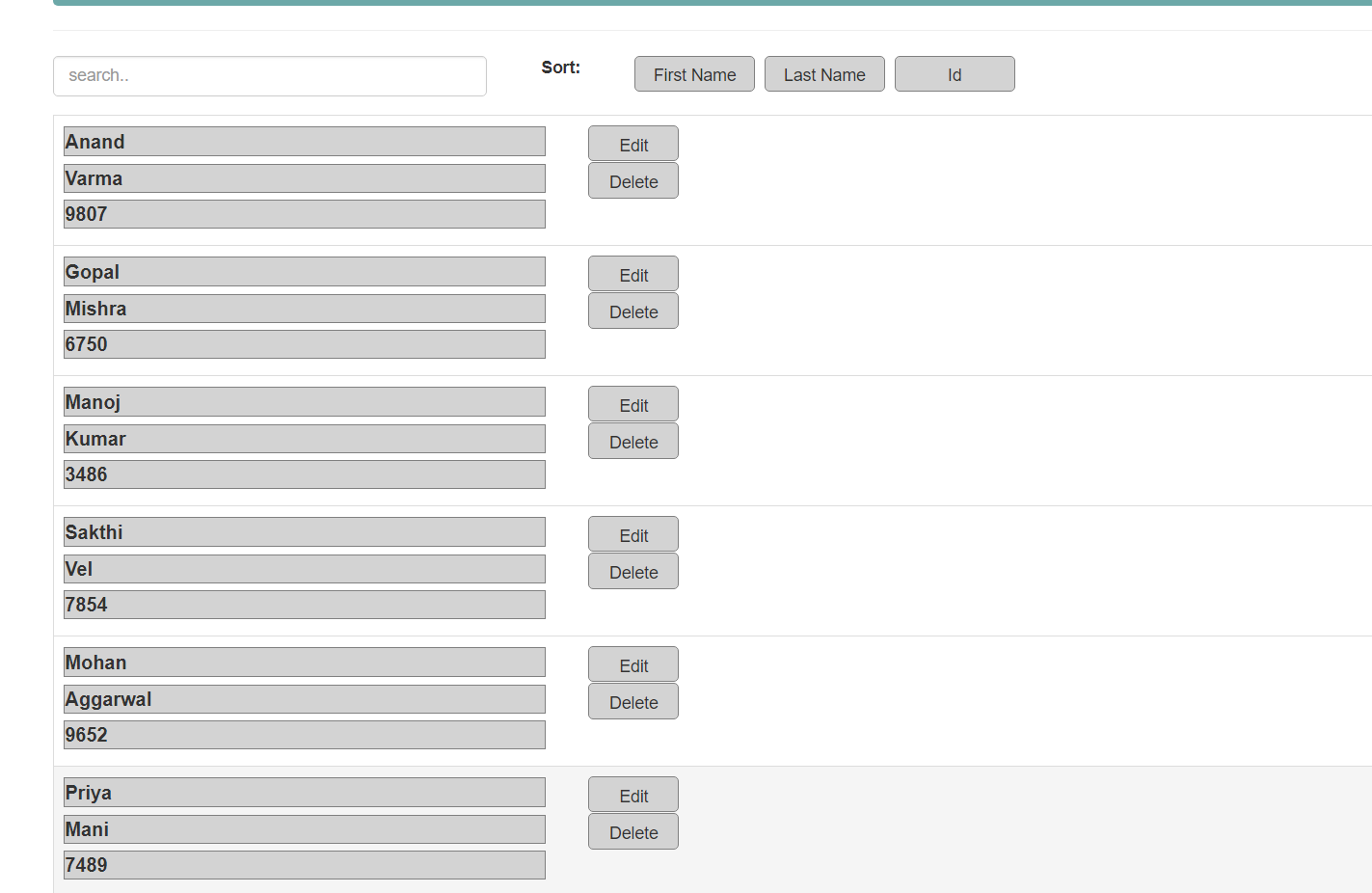
Apart from this, this page also allows the user to edit and delete the records. On clicking the edit button , the details will be listed in the add new user fields and user can edit them and submit it.

Delete button will allow the user to delete the record.

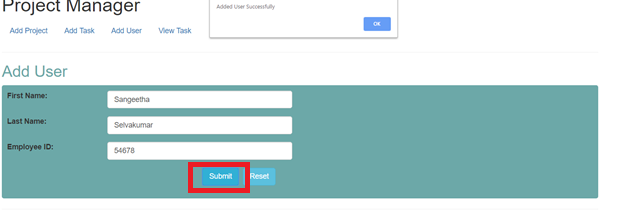
This page also allows the user to search for the user and sort the user details with the firstname,lastname and their ID.

User Page UI:

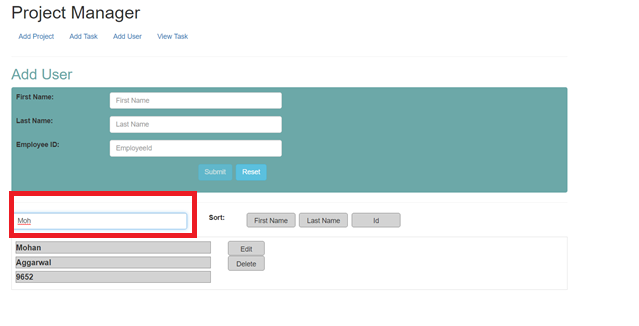




User details being entered:

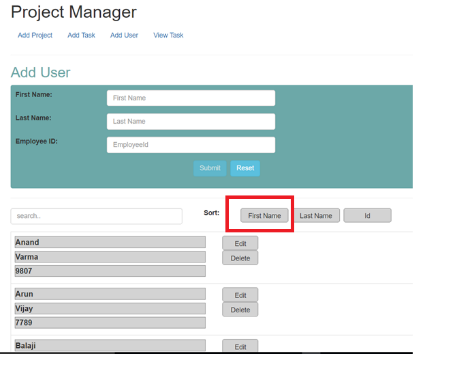


Search click:

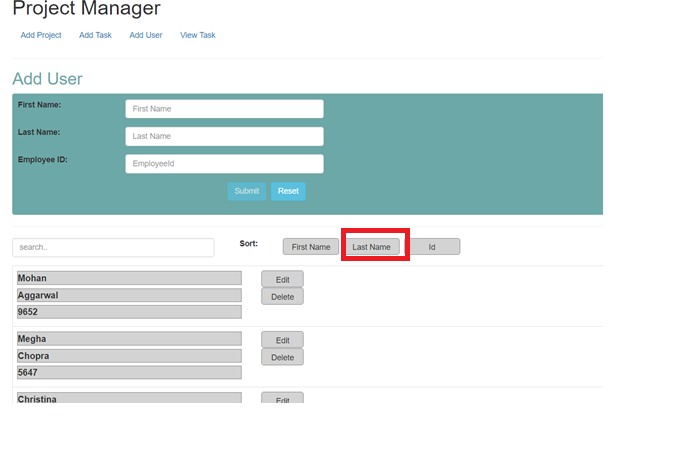


Sort:

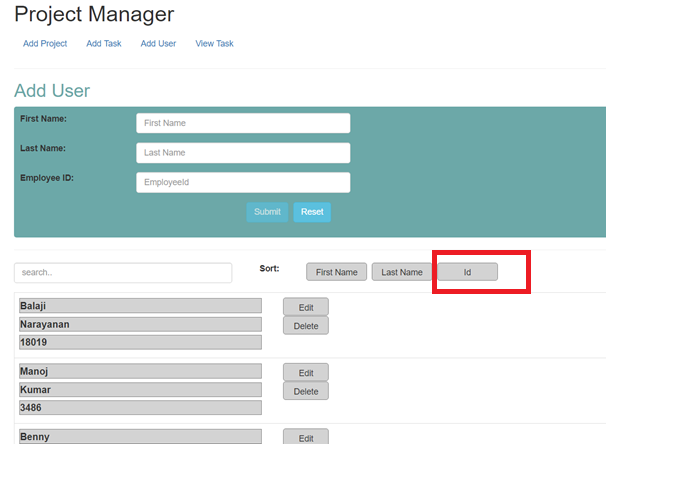
**Sort with firstname:**



**Sort with LastName:**

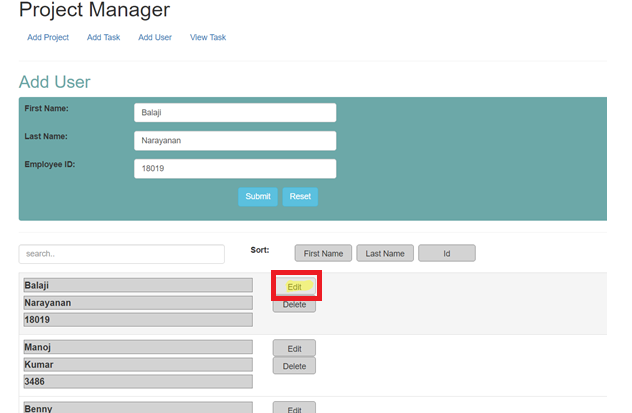


**Sort with Id:**

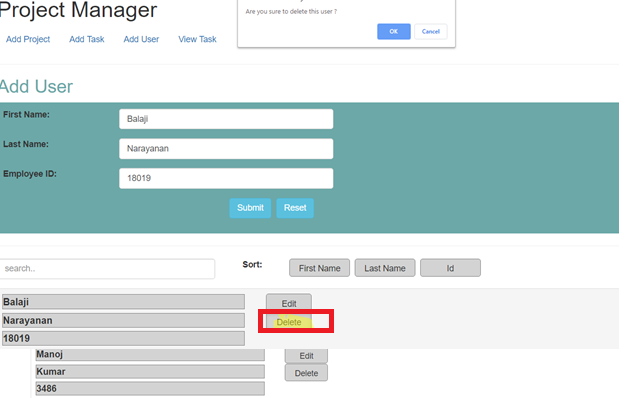


On Edit and Delete Click:

Edit will get the details of the user to the add user field and then user can able to edit them and submit again.



On clicking delete button, user can delete the details. Confirm pop up would ask for the confirmation.



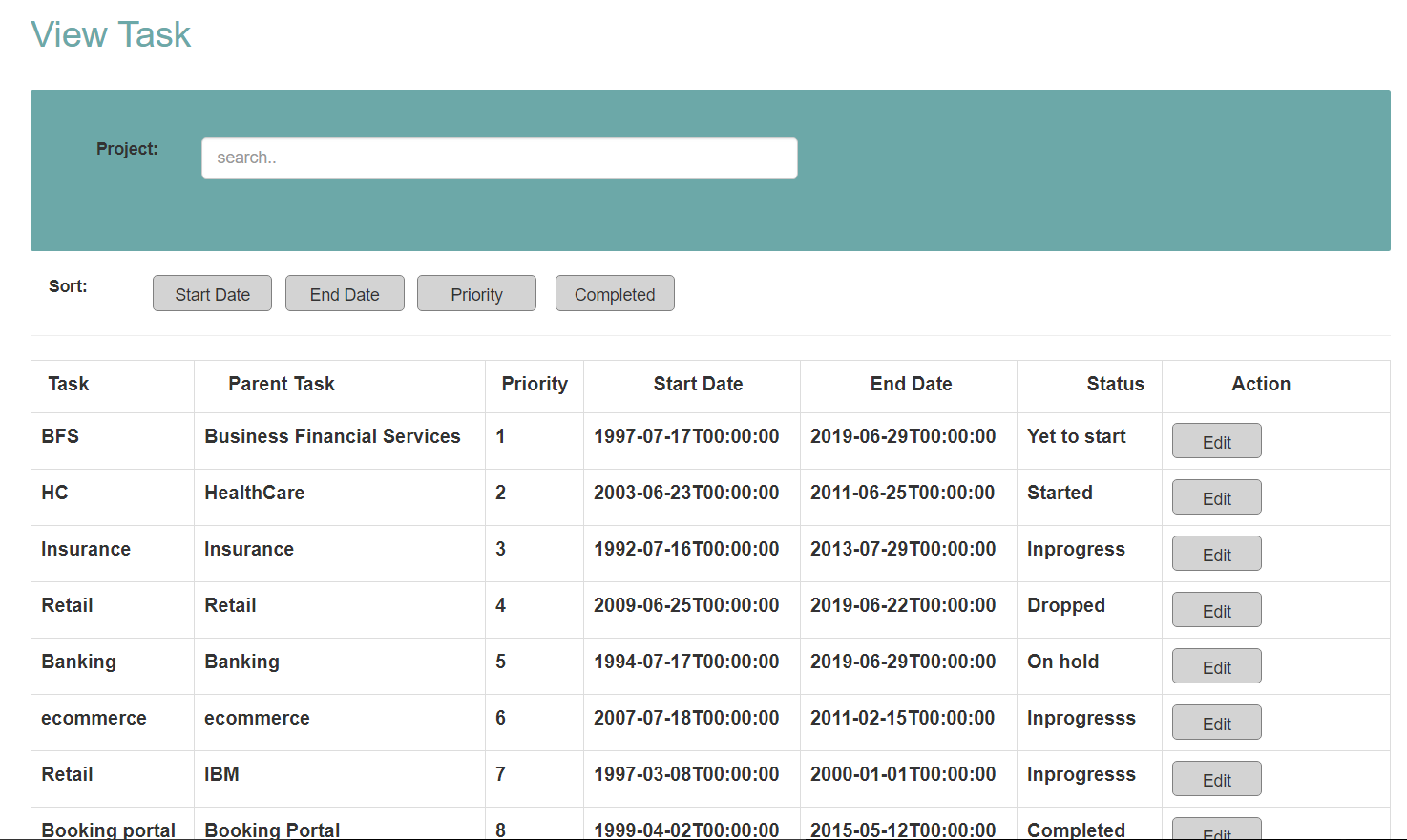
**VIEW TASK**

View page allows the user to view already existing and do some action items on the same. If the task has been suspended with the help of end task button, user can end the task to completed status.

Similar to the project page view task page also has the search option to view the existing task by clicking on the search button. Sort option is provided to get the details of task based on start/end date, priority and completed status.

On Clicking to the edit button listed in the task , it will open the update page, where user can update the required details(page very similar to add task).

View page UI:

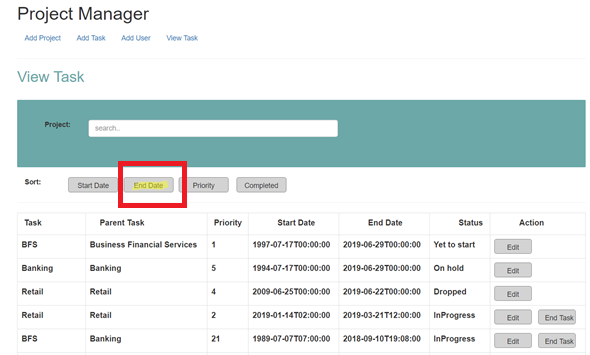


Search the task:

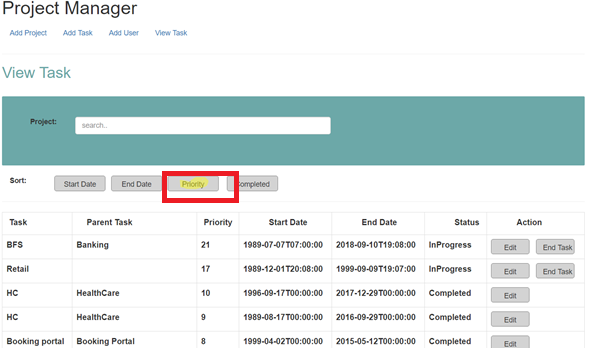
Sort :

Sort with dates:

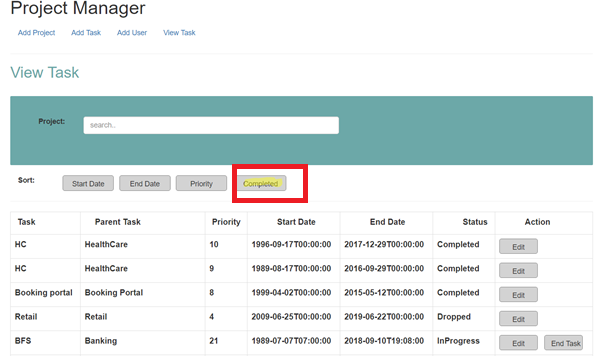




Sort with Priority:

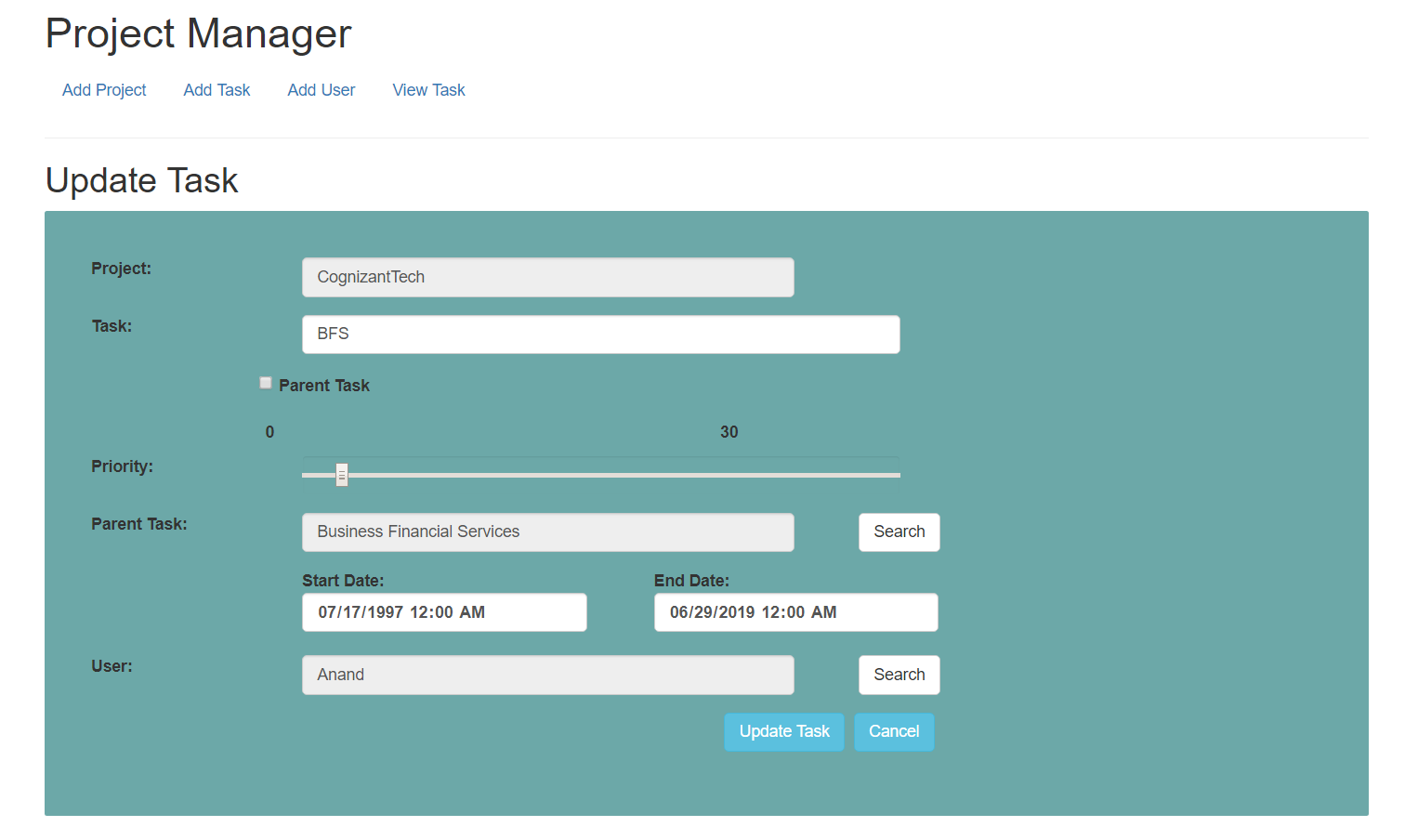


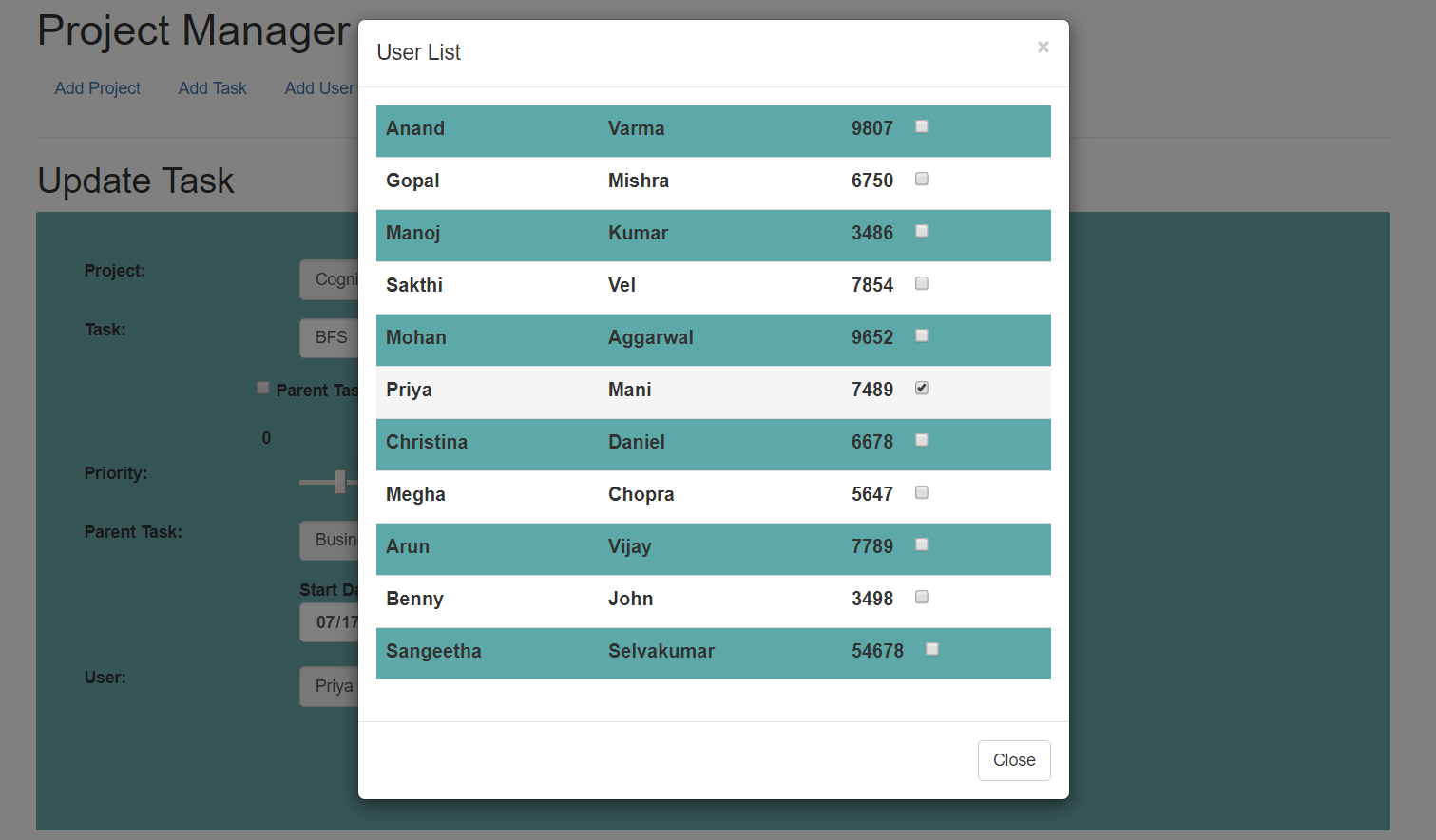
Sort with Completed status:

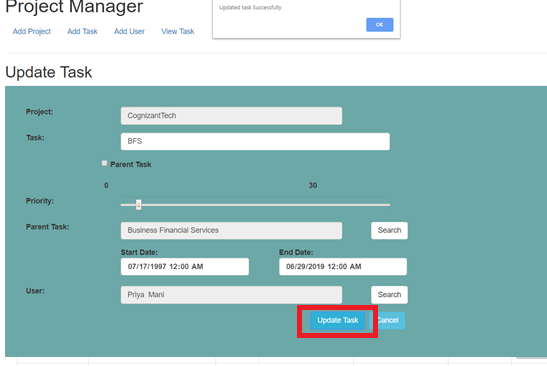


On Edit:

On clicking edit button, page will navigate to update task( the similar page as add task), where user can update the required details and submit the data.

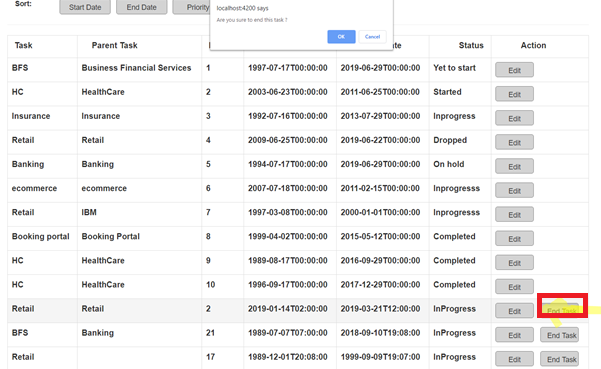


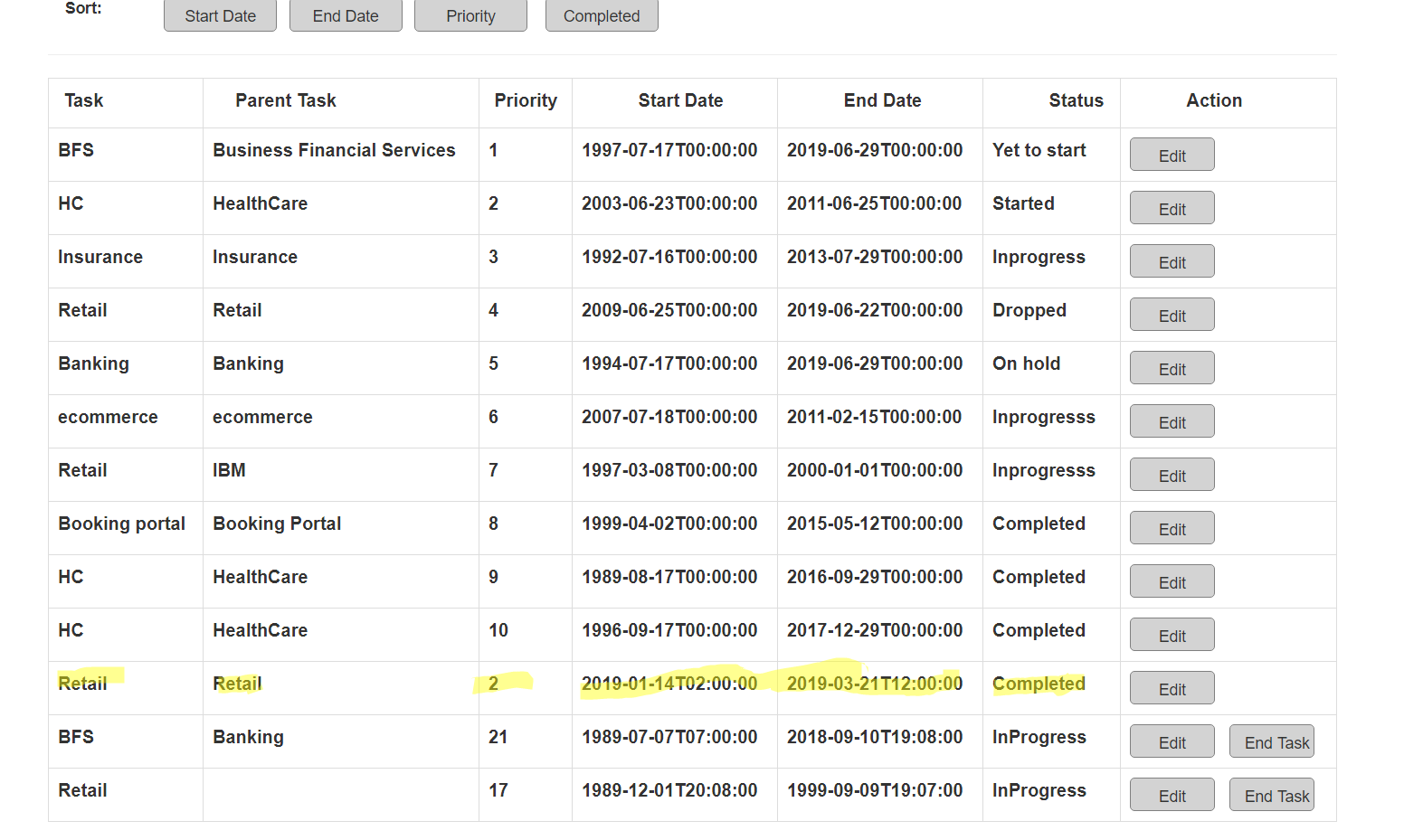




On End Task:

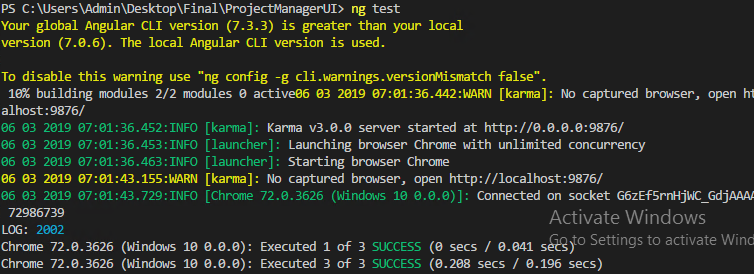
End task would end the task which is in progress to completed status.

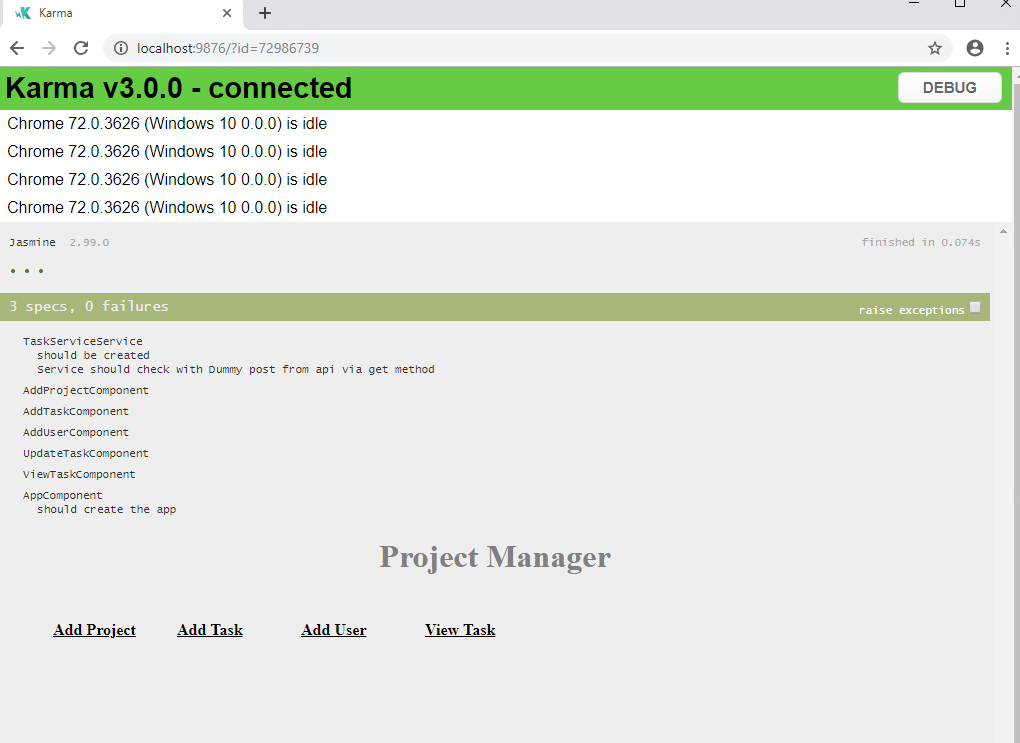




**ANGULAR TESTING**

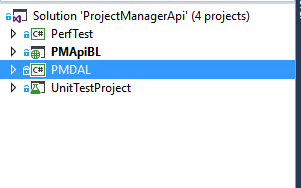
**Testing:**





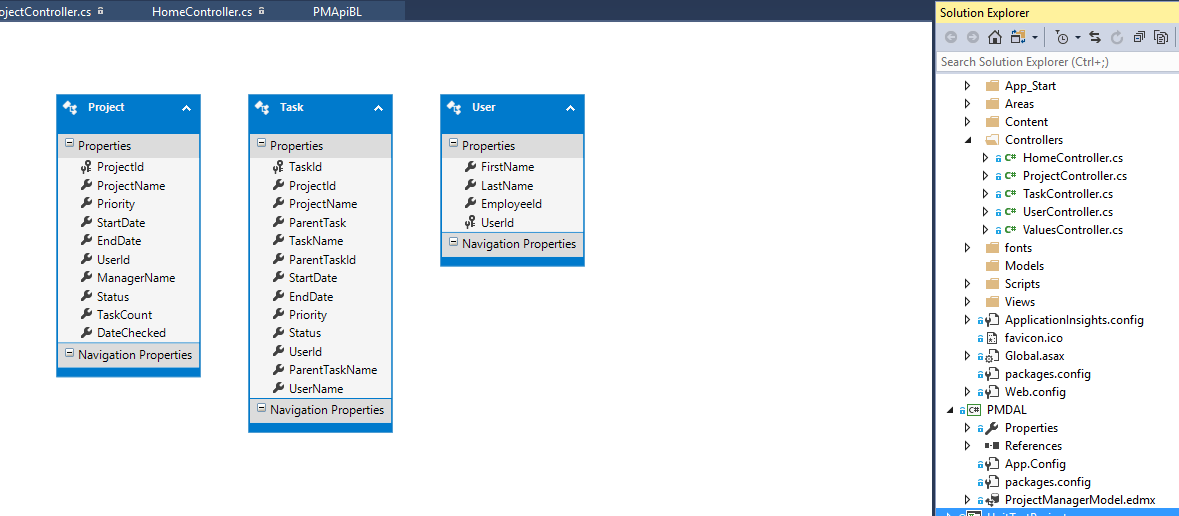
**WEB API APPLICATION**

WebAPI application is made of 2 layers here, one is database layer and other is business layer.



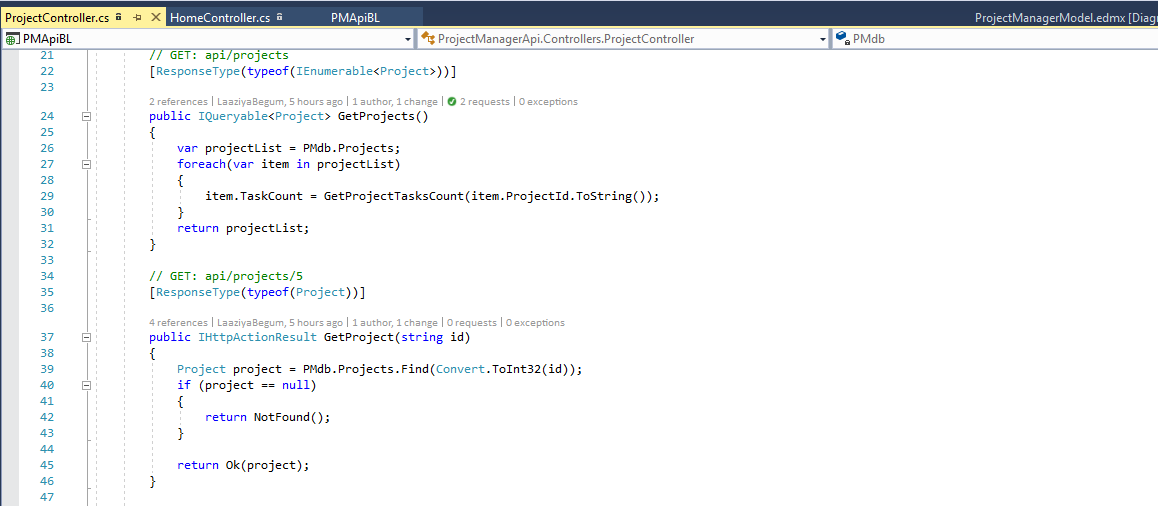
ProjectTaskManagerAPI is connected with the database via entity frame designer framework via database. Below are some snapshots of the code .

.edmx:

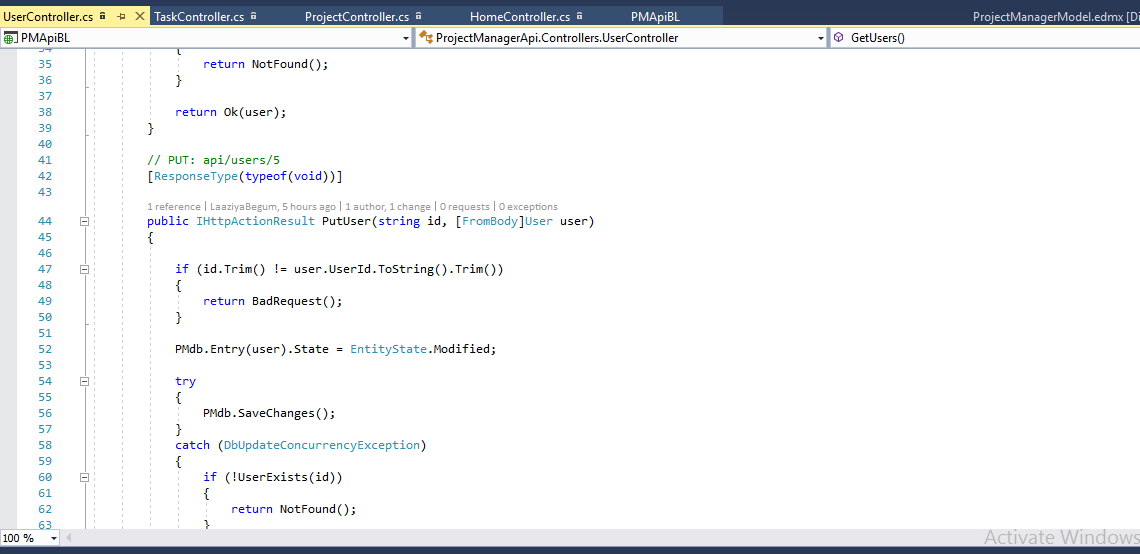


**Controllers:**

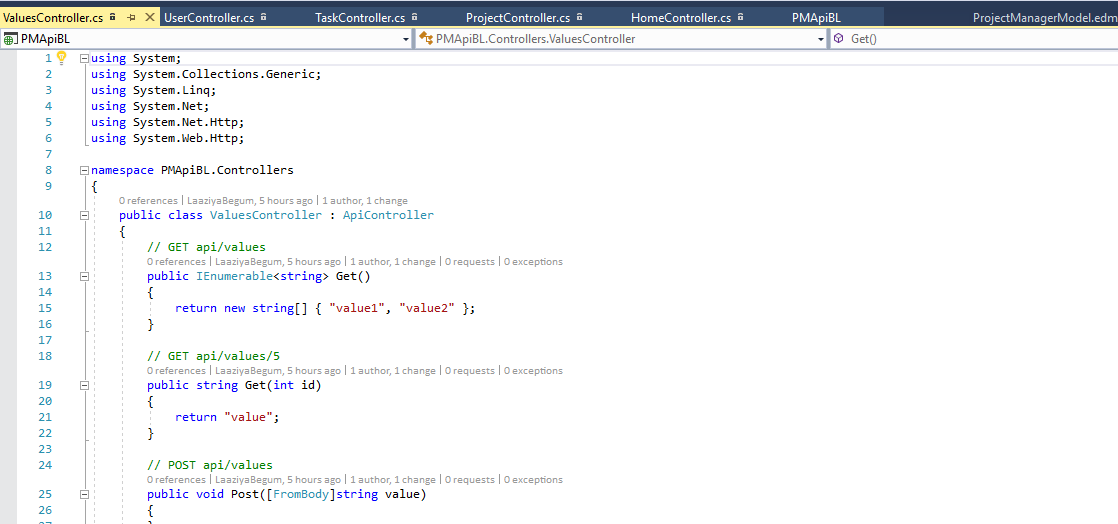
Project Controller:



User Controllers:

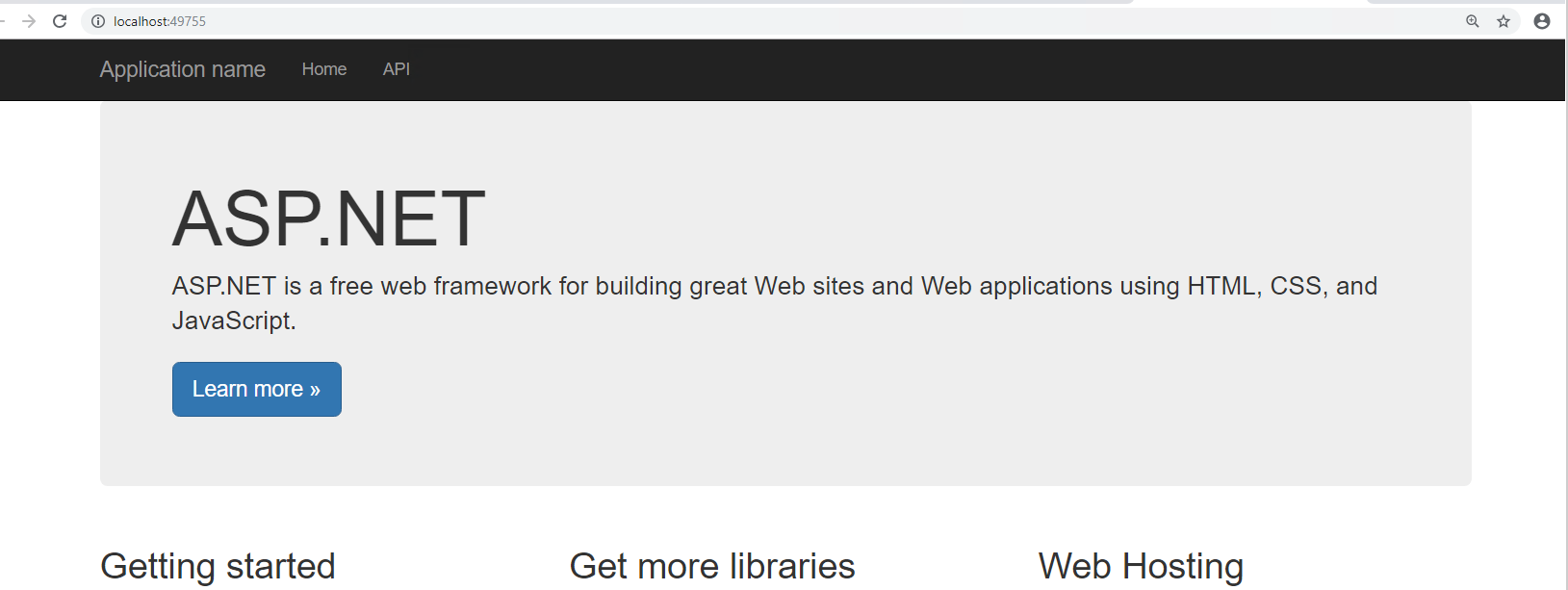


Value Controllers:



WebApi in local server:

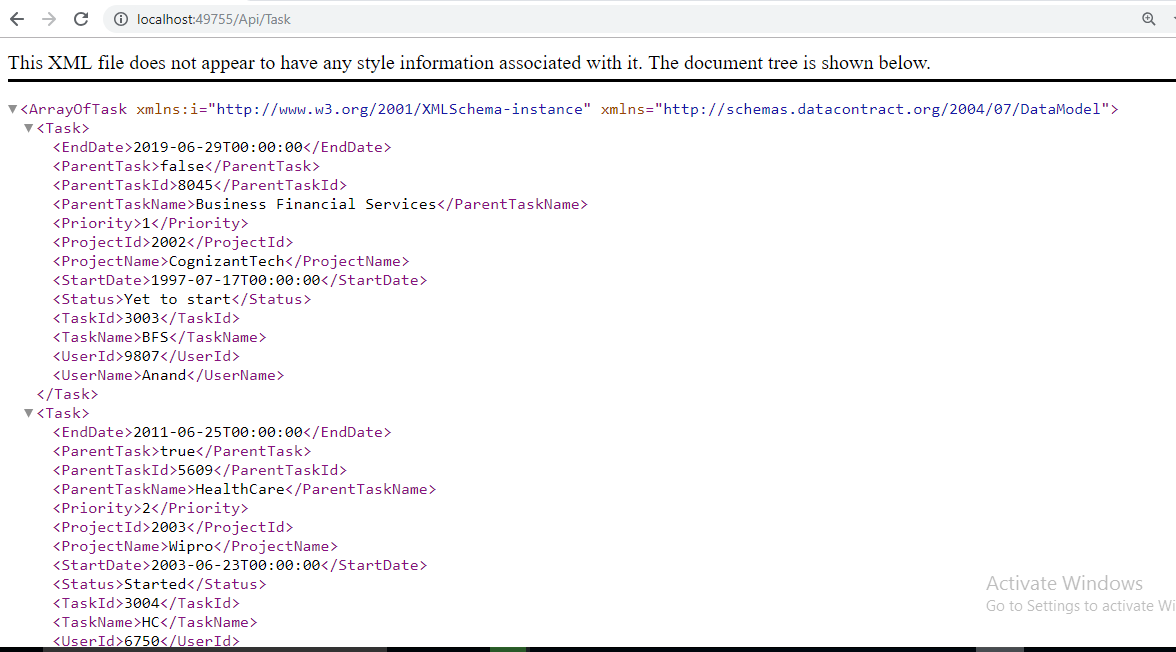
**Home page:**



**User:**



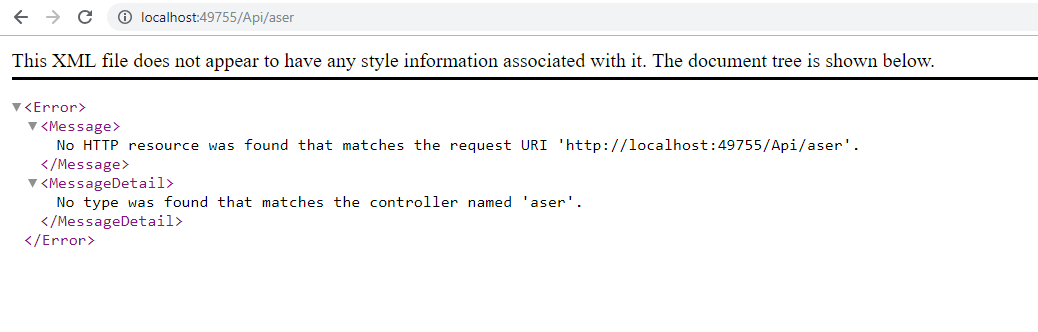
**Task:**



**Project:**



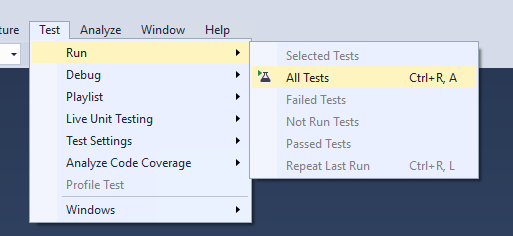
**Incorrect url:**

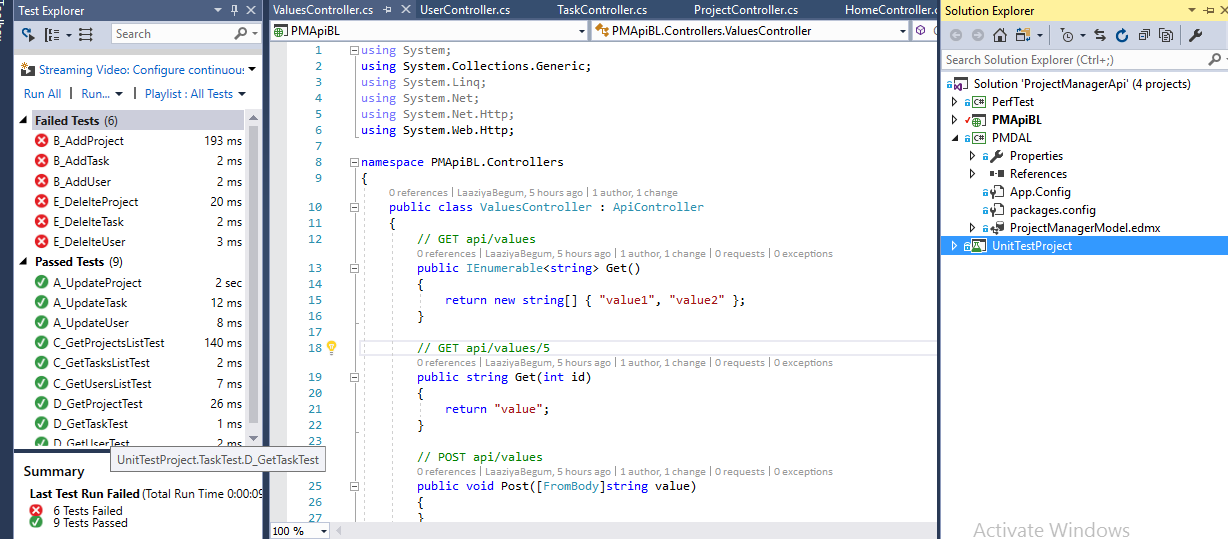


**NUNIT TESTING**

Unit test project has around 15 test cases, Have provided some values in the UT which is ran against the running code.

Out of 15 testcases , 6 failed (negative test results) and 9 passed (positive test result)



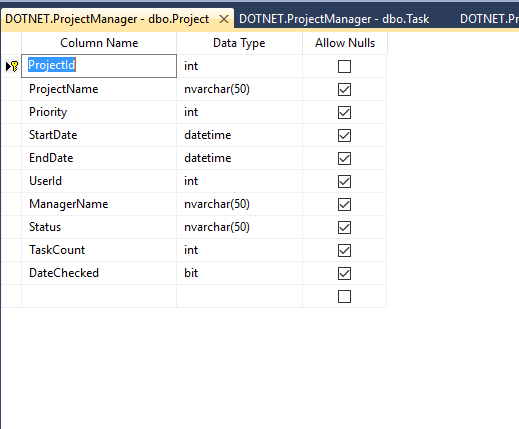


**SQL DATABASE**

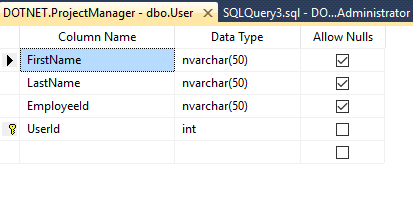
For ProjectTaskManagerAPI, have created Database named Projectmanger with the DOTNET user name and windows authentication. Same Details have been used up in the ProjectTaskManagerAPI.

Three tables are Project,User,Task. Below are the column and constraint snippets of the table.

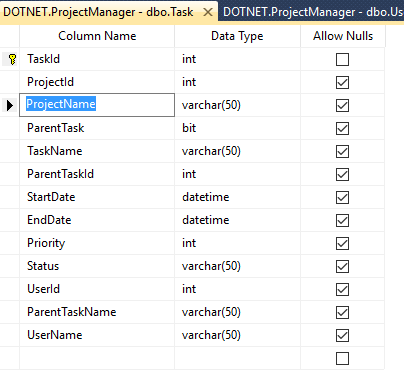
PROJECT:



USER:



TASK:

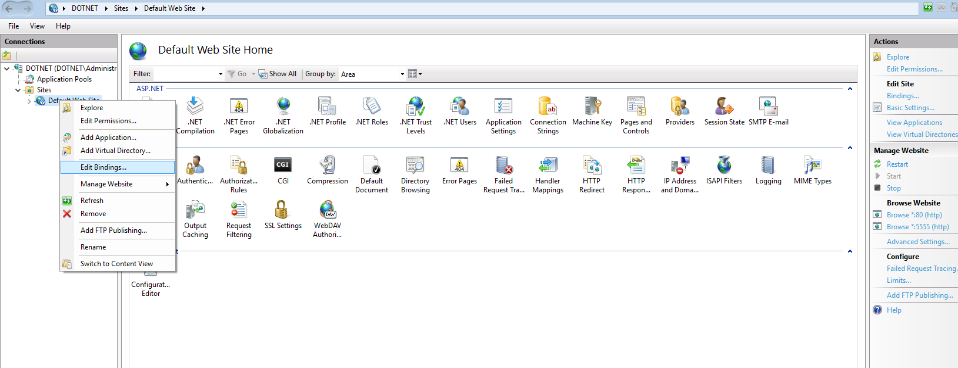


**IIS HOSTING**

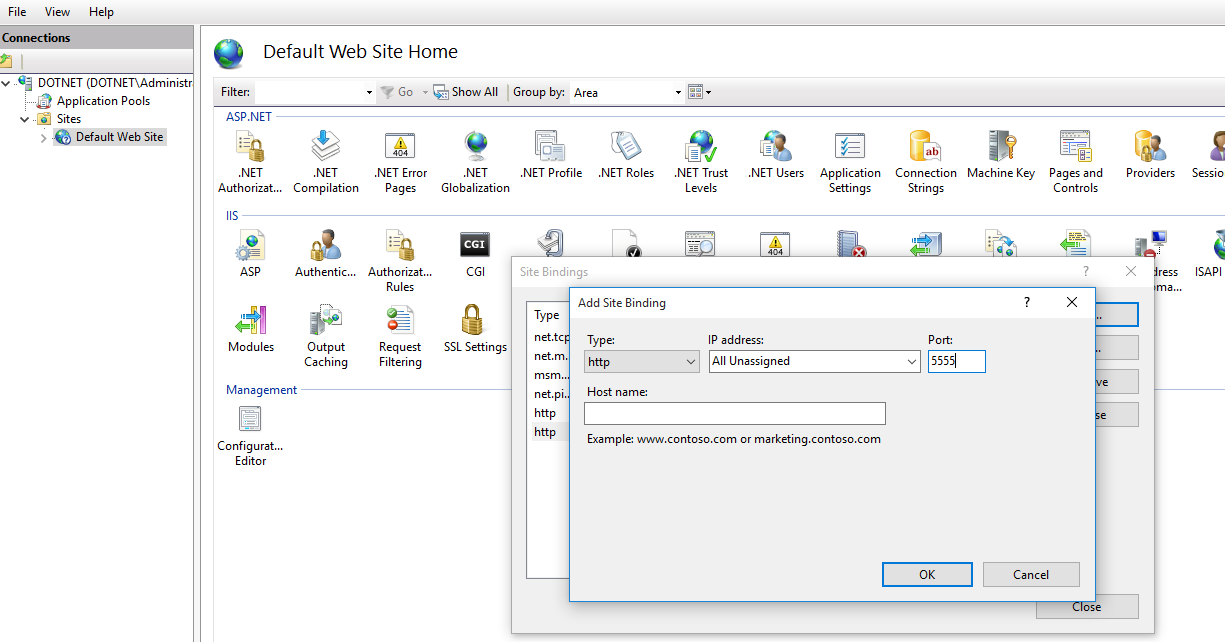
Both ProjectTaskManagerAPI and ProjectTaskManagerUI has to be hosted in IIS below are the steps followed and they have been hosted successfully in IIS in the port 5555.

Creating new port:

Open Inetmgr in the run , once the below mentioned window opened , right click on the default website and click on edit binding.



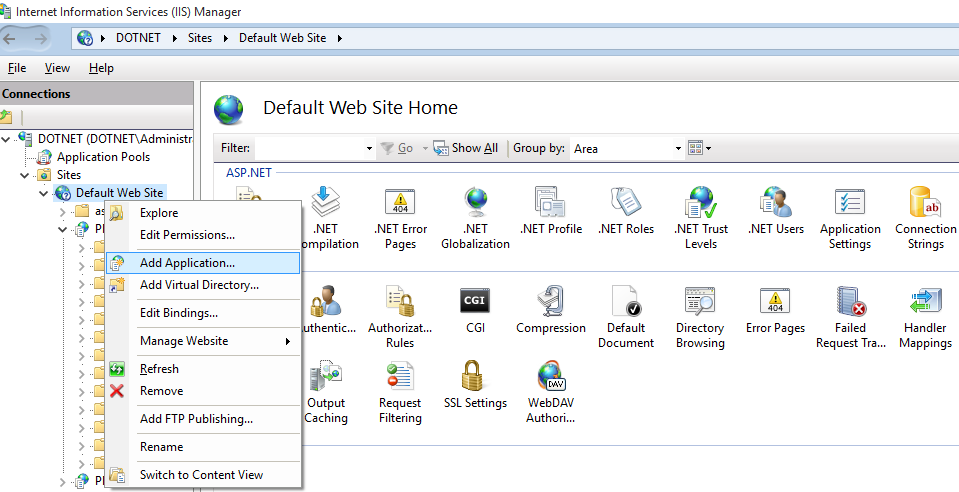
In the binding screen, click on add new binding. In that add the new port (i.e 5555) and then click ok . Now new port has been created.



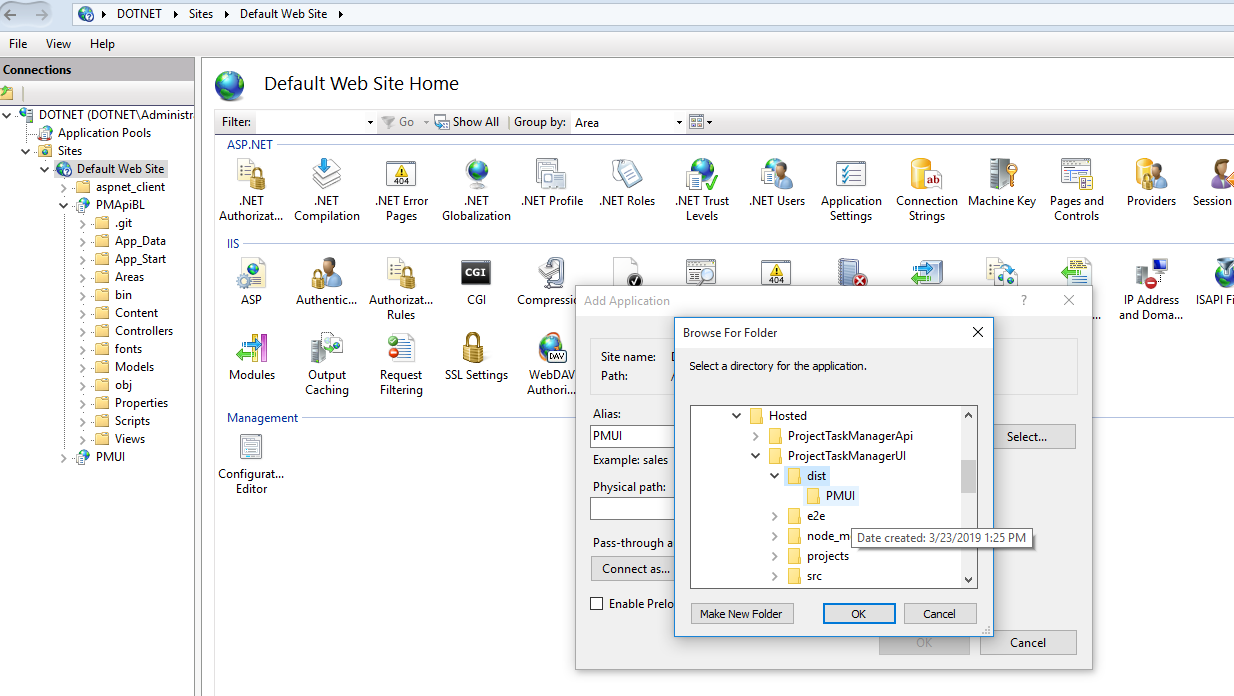
To host Angular UI:

* In the default website, right click->AddApplication
* Give the Alias name and then open the folder, make sure that angular project when build with the ng build –prod , creates the dist folder with the needed files.
* Open that file ProjectTaskManagerUI->dist->PMUI.
* Click on the ok button
* Make sure new application with PMUI has been created, check the permission whether it has been assigned with the user name “Everyone” with the all access.
* Now ProjectTaskManagerUI (Angular UI) has been hosted).
* Right click on the browser 80 and check whether the page is getting opened as expected.

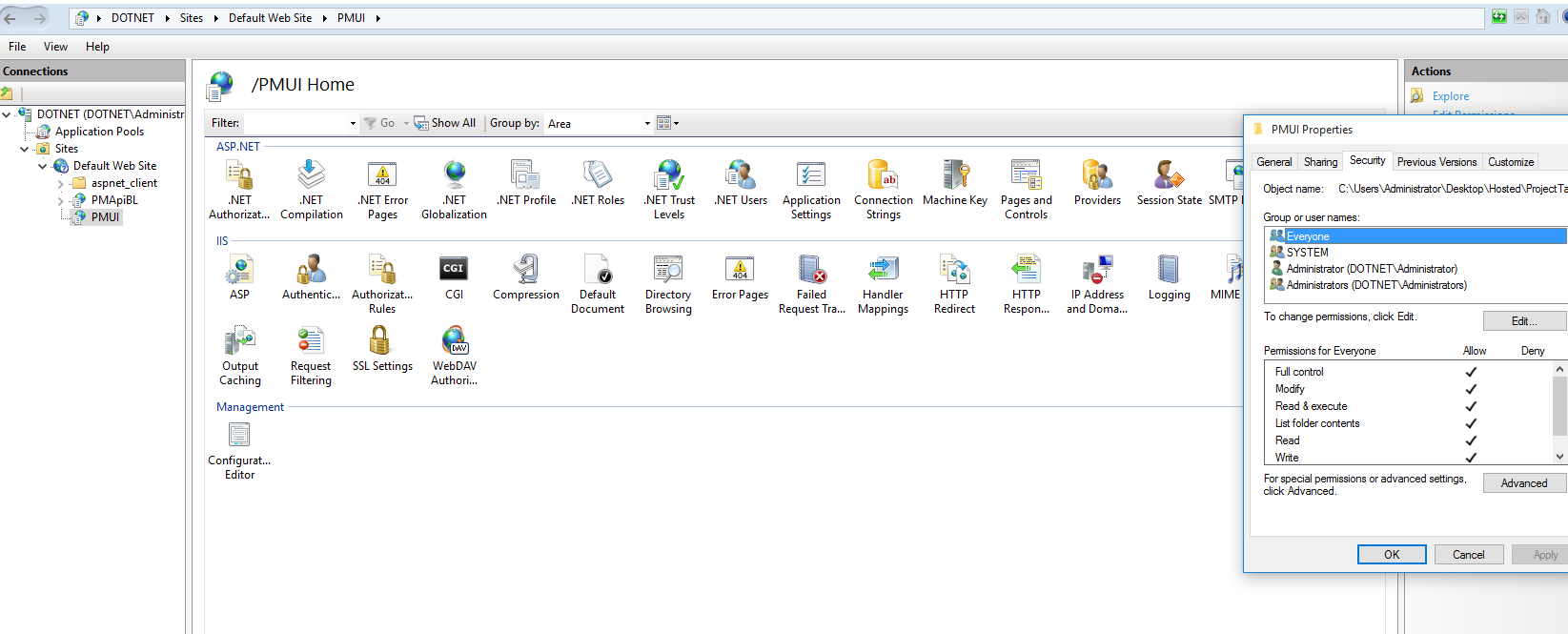
Creating Application:



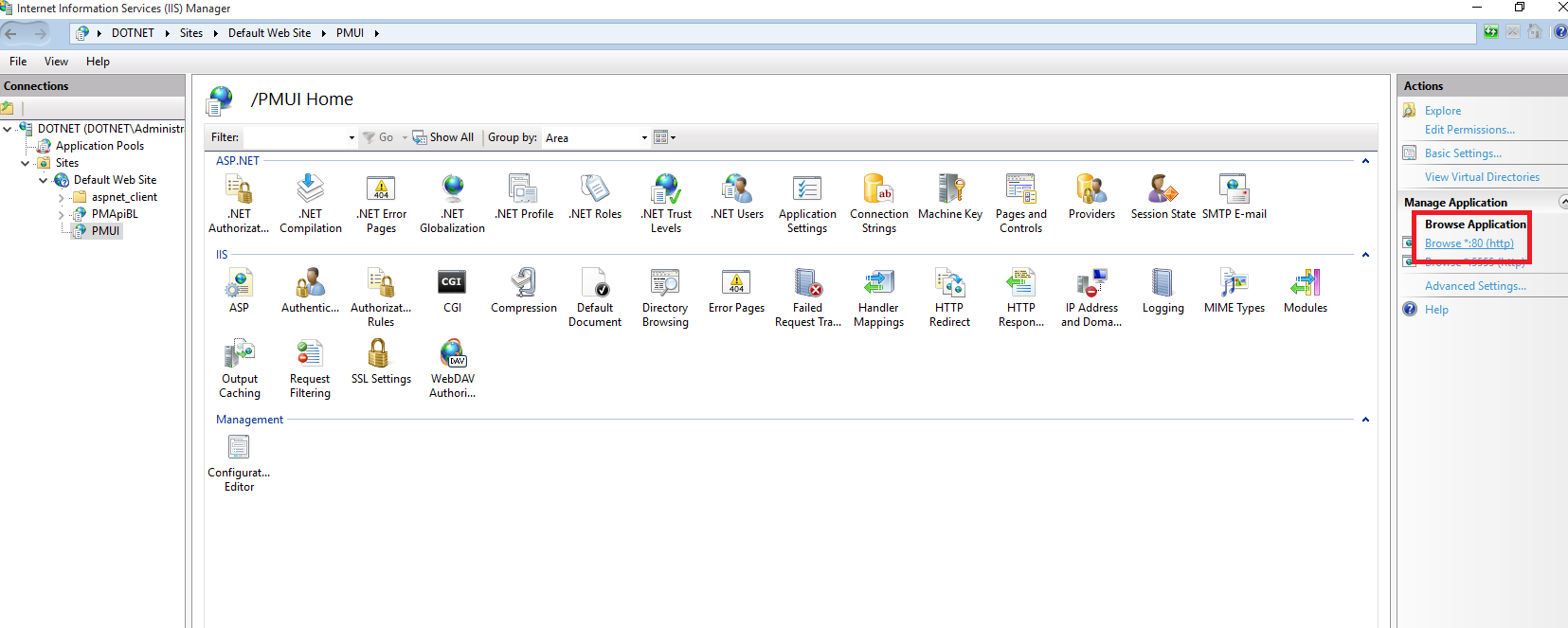
Choosing the “Dist” folder:



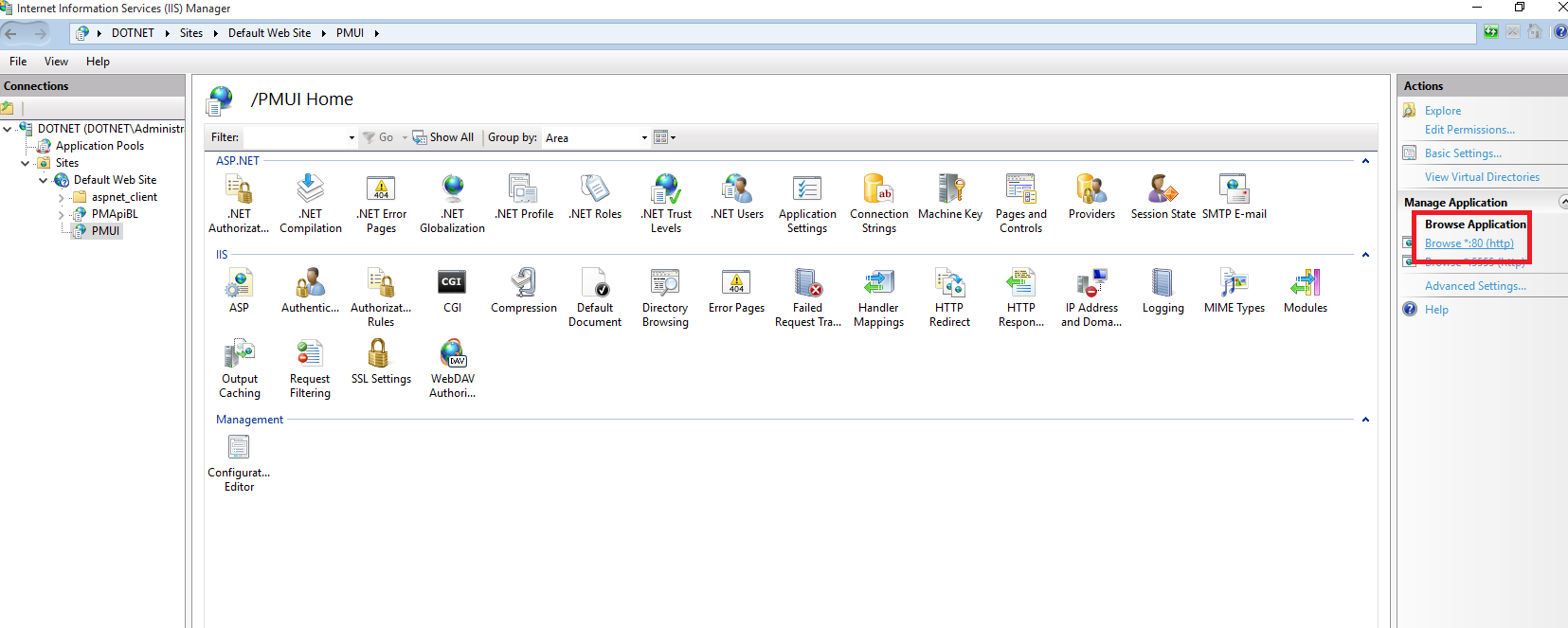
Editing the permission:



Browsing the site:



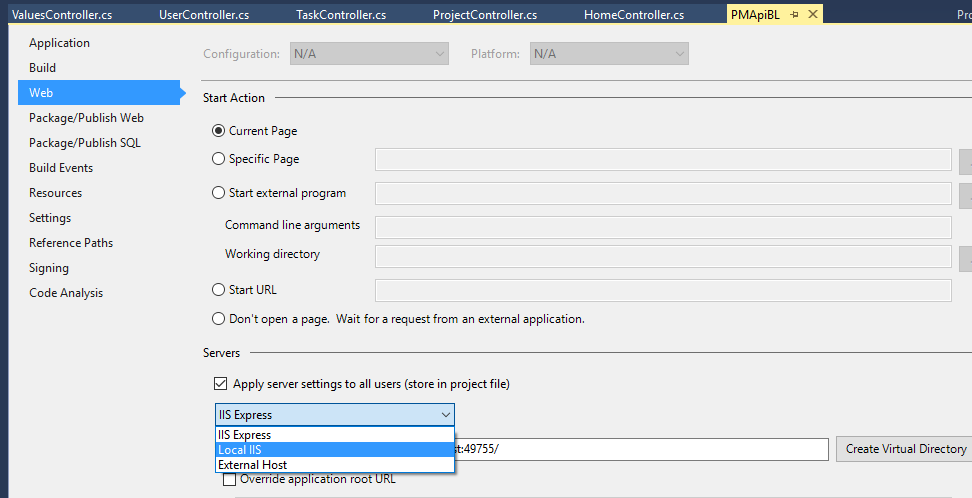
IIS hosted Angular UI:



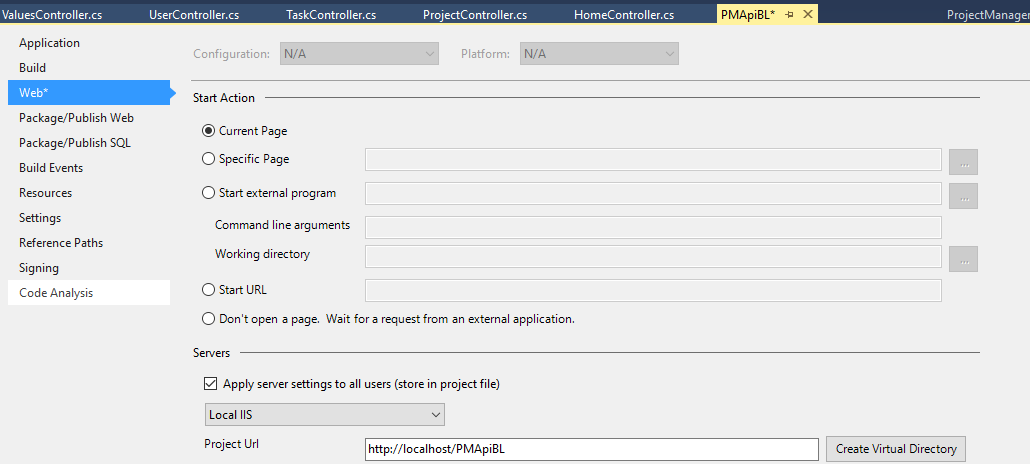
To host WebAPI:

* Open the WebAPI code in the visual studio
* Right click on the startup project
* Click on web->onservers,select local host and click on the create virtualdirectory.
* Now search for Inetmgr in the run, once IIs window is opened .
* Click on the DefaultWebsite-> then check whether the hosted named virtual application has been created , in my case it would be pmAPIBL.
* Once the application is created, check for the permission, provide the user access to “Everyone” with all granted permission.
* Now WebAPI has been published . browse with the port which has been created, it should open the page as expected.

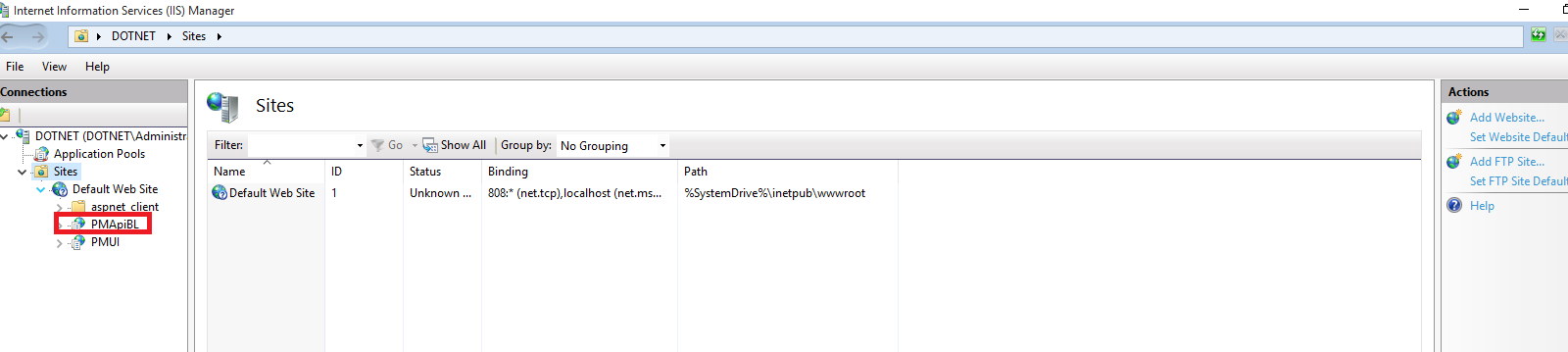
Selecting the Local IIS:



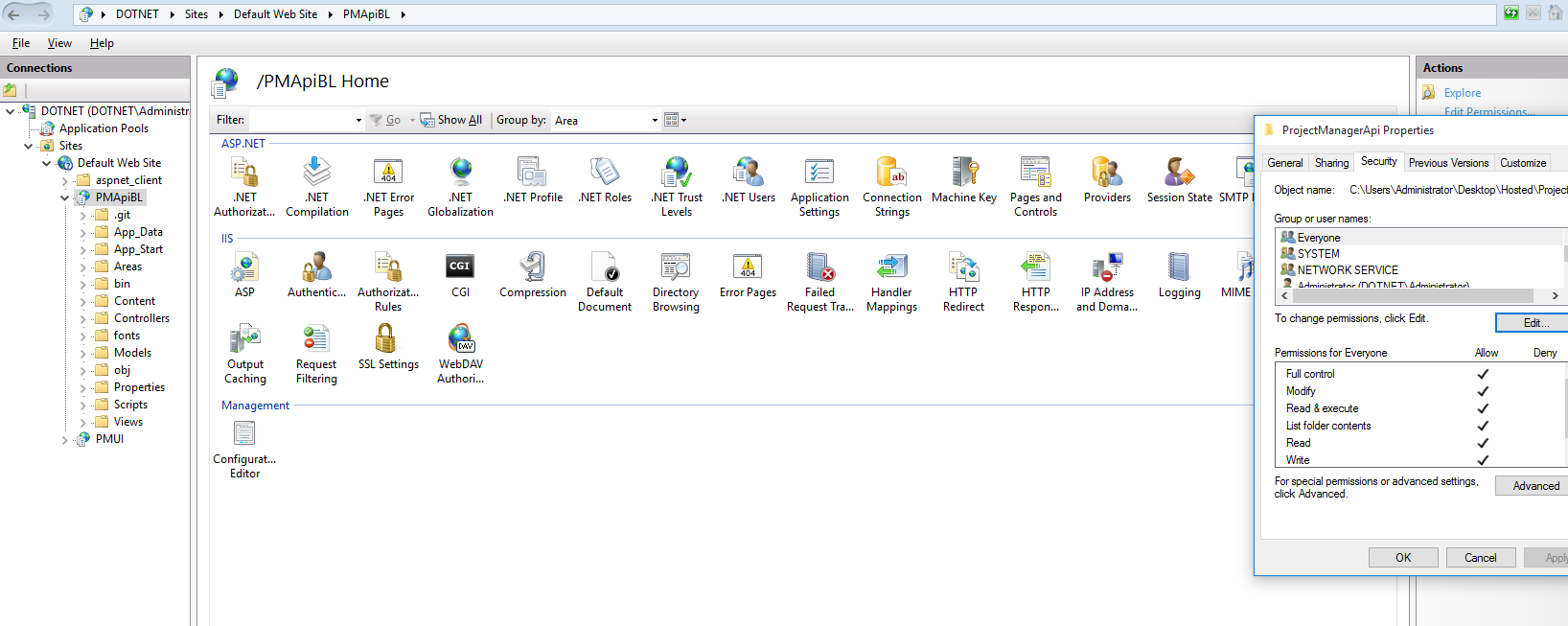
Creating the virtual Directory:



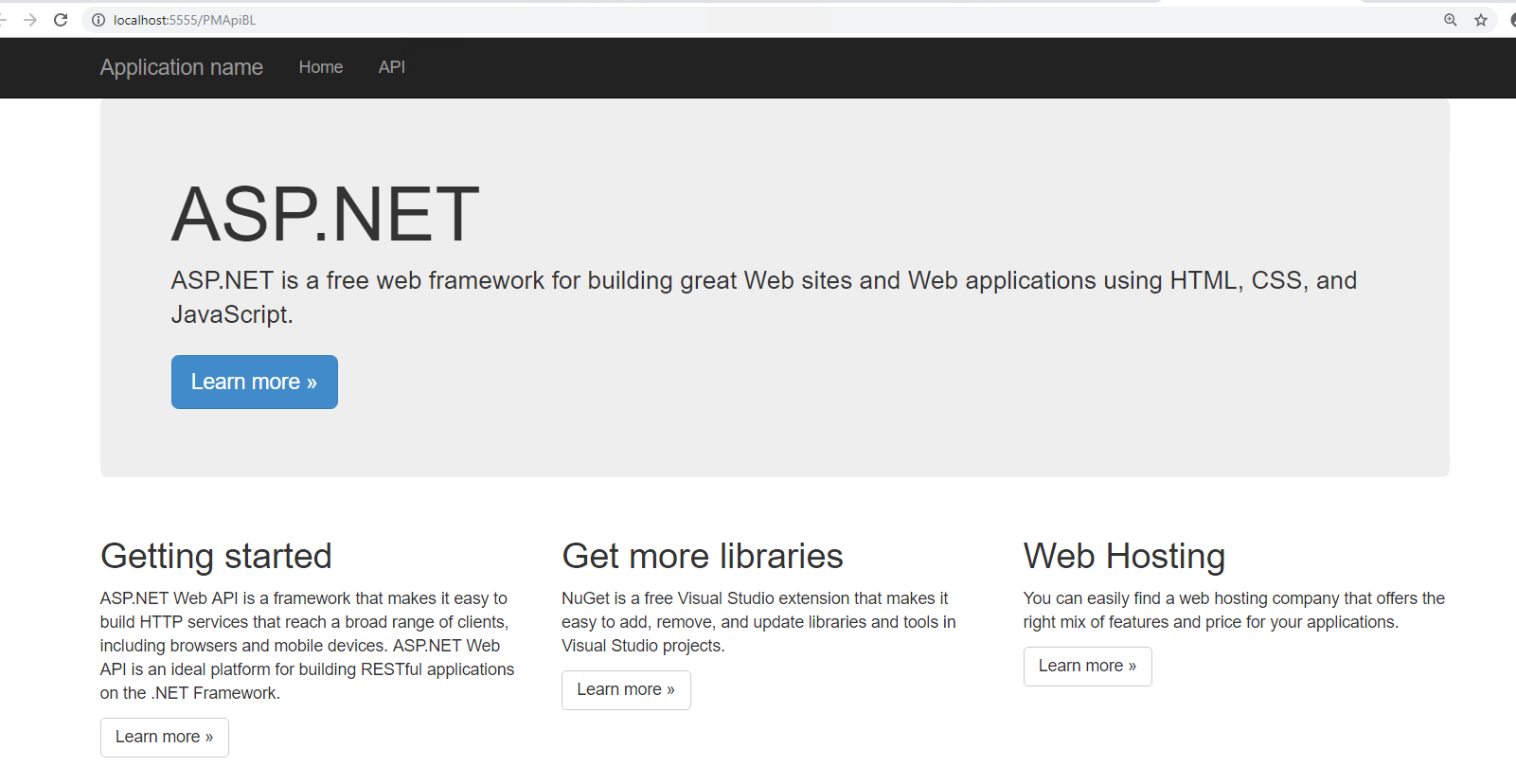
WebAPI hosted in IIS:



Checking for the permission:



Page successfully opened with the port:



**GIT HUB REPOSITORY**

ProjectTaskManagerUI , ProjectTaskManagerAPI and SQL scripts has been checked in the GIT Hub, for the review purpose they have been made to visible public.

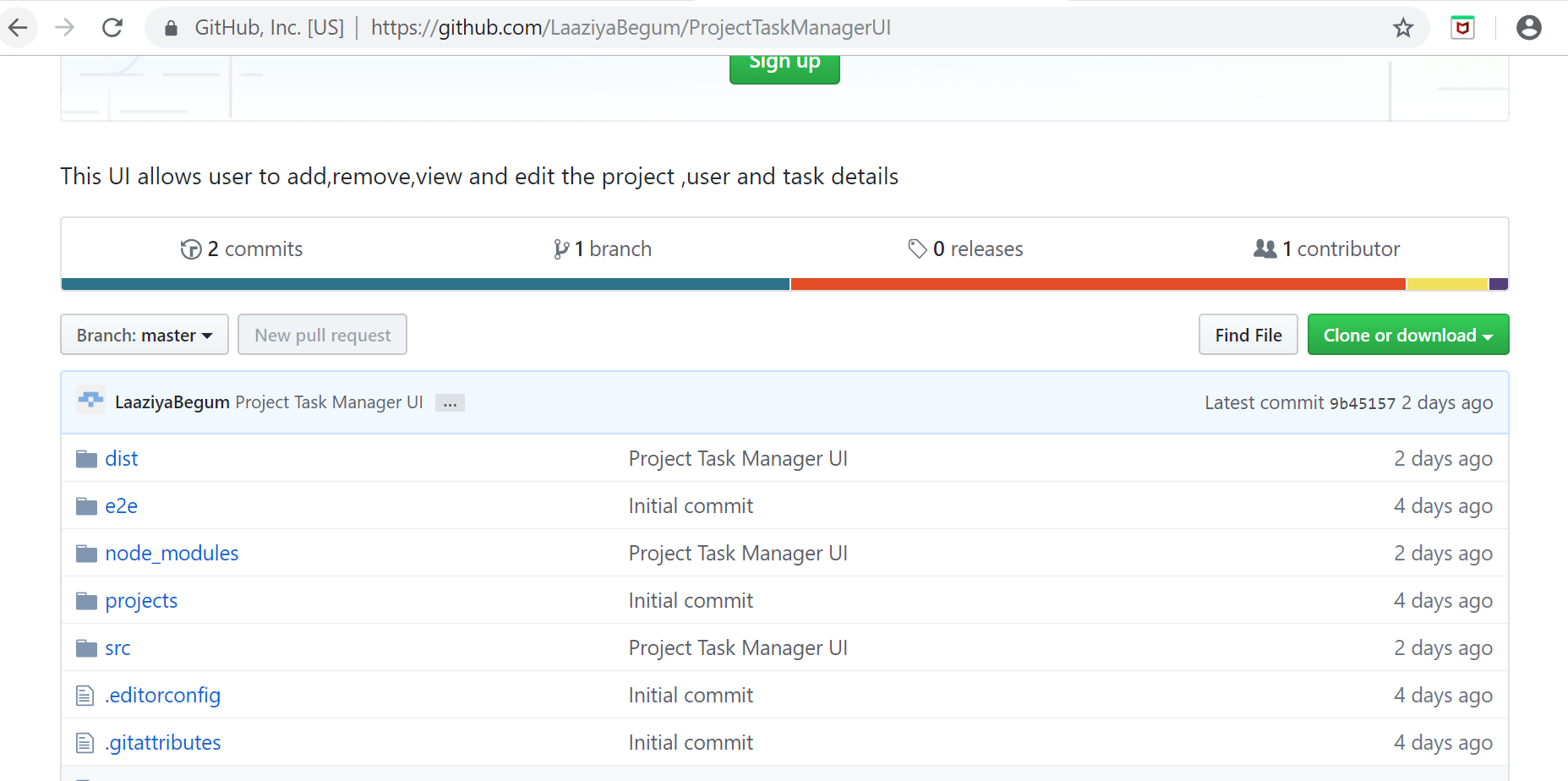
Git Hub Repositories :

<https://github.com/LaaziyaBegum/ProjectTaskManagerUI>

<https://github.com/LaaziyaBegum/ProjectTaskManagerApi>

<https://github.com/LaaziyaBegum/SQL_Scripts>

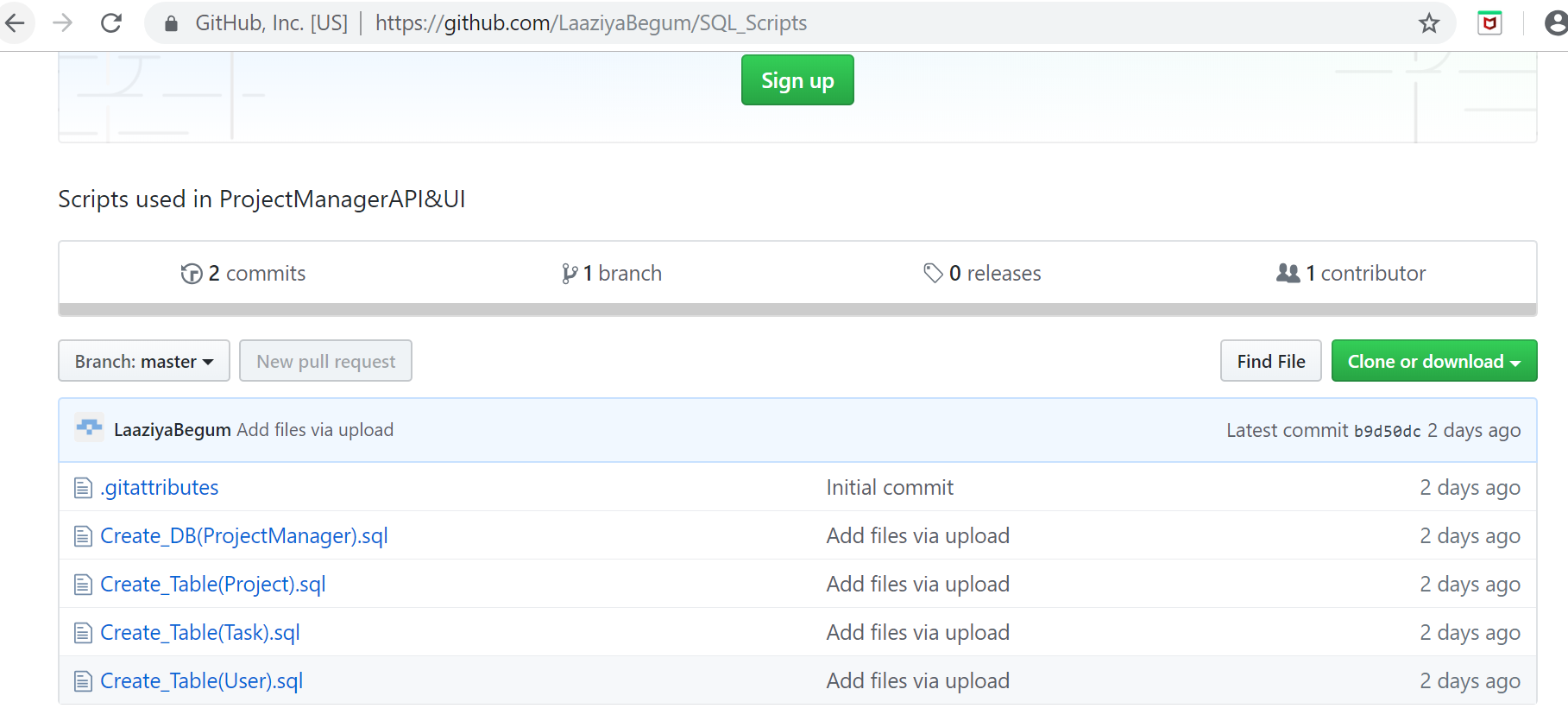
ProjectTaskManagerUI



ProjectManagerAPI



SQL SCRIPTS



**JENKINS**

Have created the Jenkins jobs by linking git path to the item, it has been build successfully . And corresponding workspace has been created successfully under Jenkins folder.

Have listed the below screenshots how jenkin job has been created and build.

* Install all necessary git library and materials from Jenkins manage plugin
* Go to Jenkins dashboard, create new item. Provide the item name, in my case it would be ProjectTaskManagerAPI and ProjectTaskManagerUI.
* Give the description as required , under the source code management select git and provide the url,

ProjectTaskManagerUI: <https://github.com/LaaziyaBegum/ProjectTaskManagerUI>

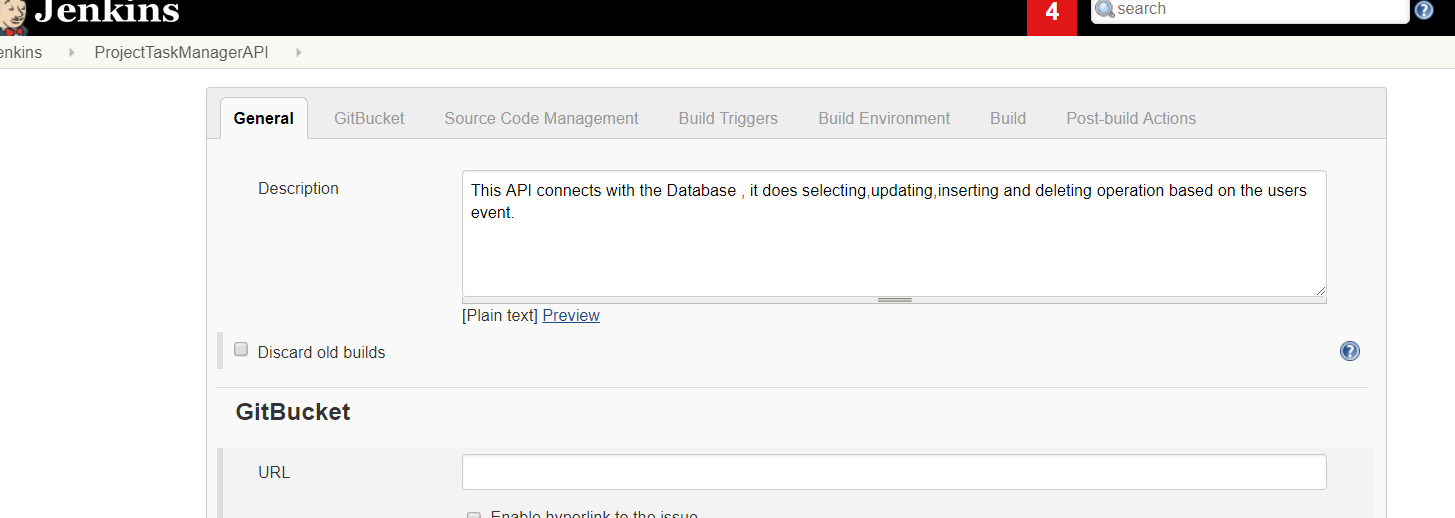
ProjectTaskManagerAPI:<https://github.com/LaaziyaBegum/ProjectTaskManagerApi>

* On build, select the build with the visual studio or solution using MS build and select the MS BUILD 2017 from drop down. Click save and run the build now. Make sure your git repositories are publicly accessible.
* After some time , build will be indicated as success. To make sure that build has been success got to C:\ProgramFiles86\Jenkins\Workspace\ProjectTaskManagerAPI

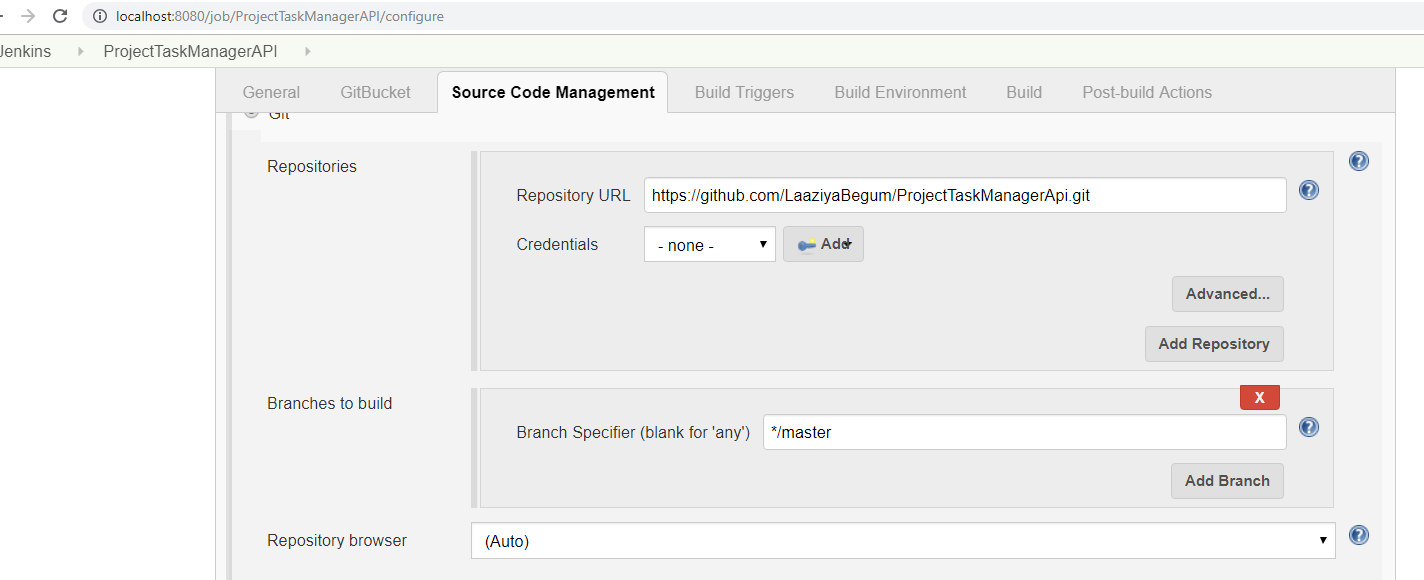
C:\ProgramFiles86\Jenkins\Workspace\ProjectTaskManagerUI

Below have provided screenshot for the ProjectTaskManagerAPI, in the same way have done for ProjectTaskManagerUI.

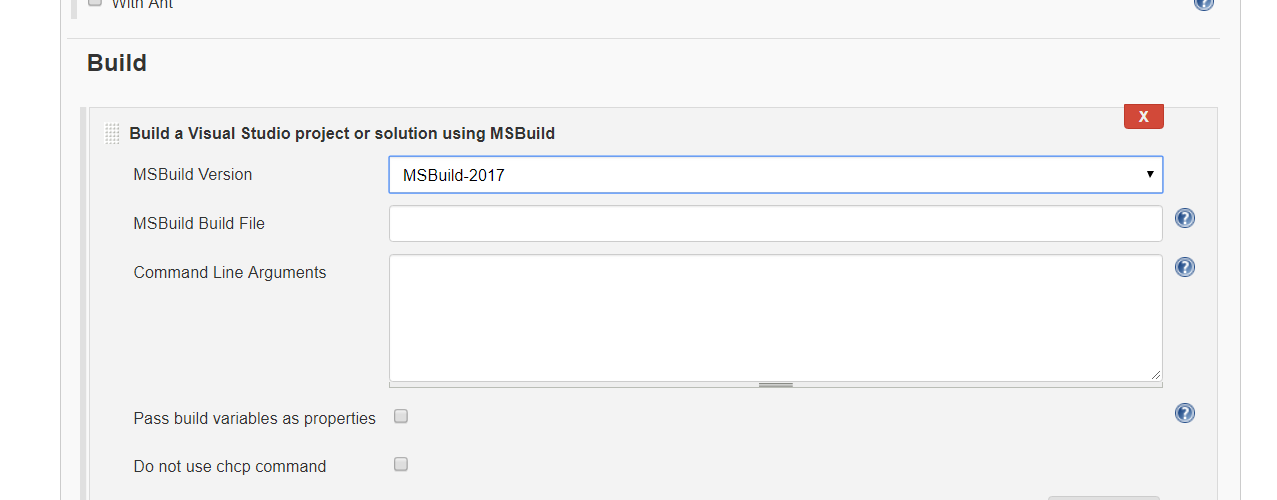
Creating the job:



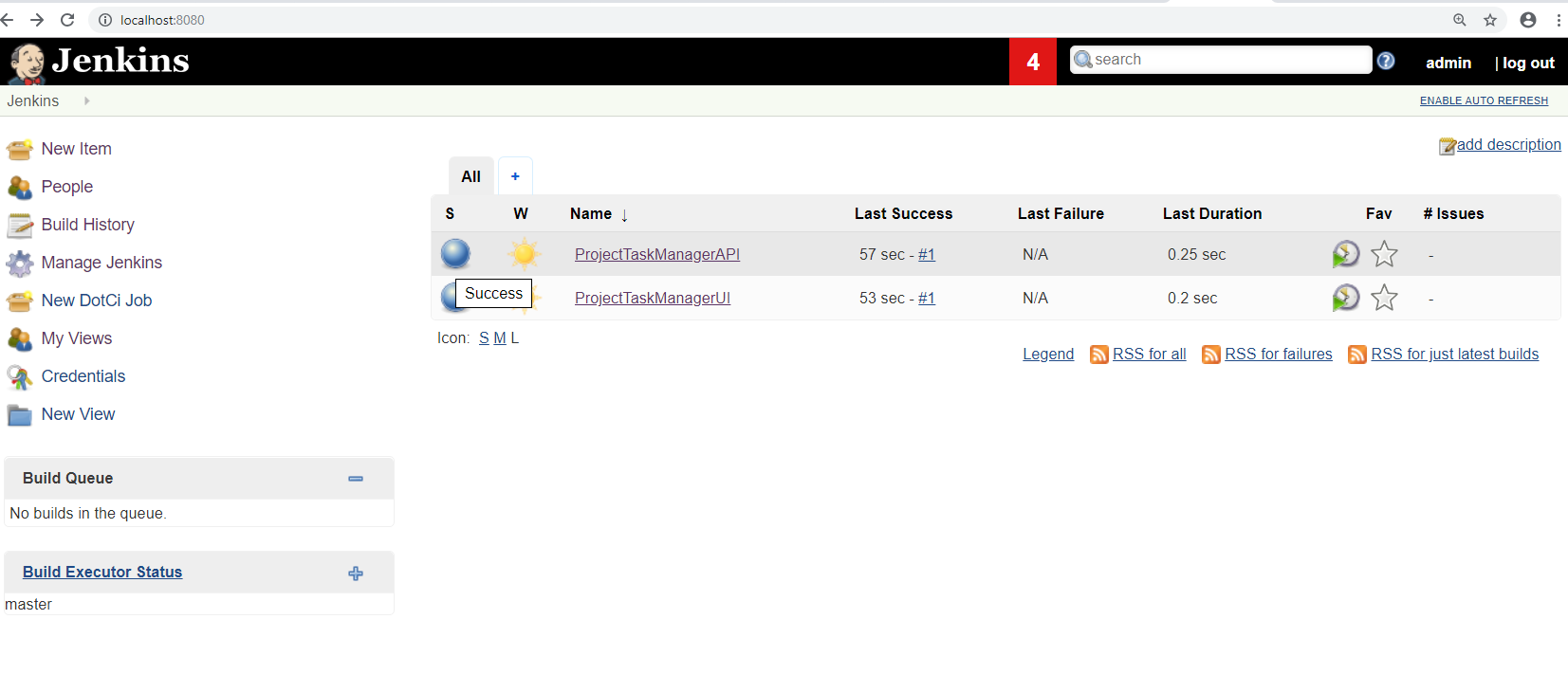
Providing the Git link:



Providing the build details:



Job Status:



Jenkins Workspace:

