**Software Project Management Project Requirements**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Task Name | Duration | Predecessors | Resource Names |
| **1** | **Scope** |  |  |  |
| 2 | Determine project scope | 3 day |  | Management |
| 3 | Define preliminary resources | 4 day | 2 | Project Manager |
| **4** | **Analysis/Software Requirements** |  |  |  |
| 5 | Conduct needs analysis | 5 days | 3 | Analyst |
| 6 | Develop preliminary budget | 2 days | 5 | Project Manager |
| 7 | Review software specifications/budget with team | 2 day | 6 | Project Manager,Analyst |
| 8 | Develop delivery timeline | 1 day | 7 | Project Manager |
| **9** | **Design** |  |  |  |
| 10 | Develop functional specifications | 5 days | 8 | Analyst |
| 11 | Develop prototype based on functional specifications | 4 days | 10 | Analyst |
| 12 | Incorporate feedback into functional specifications | 1 day | 11 | Management |
| **13** | **Development** |  |  |  |
| 14 | Assign development staff | 1 day | 12 | Developer |
| 15 | Develop code | 15 days | 14 | Developer |
| 16 | Developer testing (primary debugging) | 15 days | 15FS-75% | Developer |
| **17** | **Testing** |  |  |  |
| 18 | Develop unit test plans using product specifications | 4 days | 16 | Testers |
| 19 | Develop integration test plans using product specifications | 4 days | 18 | Testers |
| **20** | **Unit Testing** |  |  |  |
| 21 | Test component modules to product specifications | 1 wk | 16, 18 | Testers |
| 22 | Modify code | 3 days | 21 | Testers |
| 23 | Re-test modified code | 2 days | 22 | Testers |
| **24** | **Integration Testing** |  |  |  |
| 25 | Test module integration | 5 days | 23 | Testers |
| 26 | Modify code | 3 days | 25 | Testers |
| 27 | Re-test modified code | 2 days | 26 | Testers |
| **28** | **Documentation** |  |  |  |
| 29 | Develop Help system | 3 wks | 27FS-50% | Technical Communicators |
| 30 | Develop user manuals specifications | 2 days | 12 | Technical Communicators |
| **31** | **Pilot** |  |  |  |
| 32 | Develop software delivery mechanism | 1 day | 30 |  |
| 33 | Obtain user feedback | 1 wk | 32 | Deployment Team |
| **34** | **Deployment** |  |  |  |
| 35 | Develop deployment methodology | 1 day | 12 | Deployment Team |
| 36 | Deploy software | 1 day | 35 | Deployment Team |
| **37** | **Post Implementation Review** |  |  |  |
| 38 | Document lessons learned | 1 day | 36 | Project Manager |
| 39 | Distribute to team members | 1 day | 38 | Project Manager |
| 40 | Create software maintenance team | 1 day | 39 | Project Manager |

|  |  |
| --- | --- |
| **Staff** | **Salary per hour** |
| Management | 30 |
| Analyst | 20 |
| Project Manager | 22 |
| Developer | 15 |
| Testers | 15 |
| Technical Communicators | 12 |
| Deployment Team | 12 |
| Trainers | 10 |
| Key Executive Committee | 25 |
| Investment Banker | 20 |

**Project Requirements:**

1- Mapping of resources and tasks from the tables provided and make any necessary changes.

2- Add Milestones to check the finalization for each block of work packages.

3- Use all possible types of relations between dependent tasks.

4- Show the actual cost and work breakdown structure columns and make any necessary changes.

5- Apply a project calendar with work from 8 am till 2 pm for 6 days per week and add 31st of each month is a non-working day.

6- Add Photoshop License, Laptops and Catering Breakfast as resources (adjust the type of each of them). Add estimated cost from your own.

7- Create Calendar for each group of resources (for example developers have a specific calendar).

8- Assign the Photoshop resource to the design task and Laptops to Developers’ Tasks.

9- Apply the important “weekly visiting” task which lasts for two months (Recurrent Event).

10- Design of tasks and summary tasks in Gantt chart, so that the beginning, end, pattern and color of each task and summary tasks are displayed and the width of taskbar is modified.

11- In the graphical part of the Gantt Chart, add the resource name on the left of the task.

12- Re-design of the critical path (adjust the design from your own).

13- Add all tasks to timeline and call them out.

14- Deactivate the integration testing summary task and its working packages.

15- View the overallocated resources.

**Notes:**

* Assume Project start date: 1/1/2023.
* Apply the working hours to the project to calculate time correctly.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Instructions:**

* We will send delivery times to each team (please comply with these times to prevent non-receipt).