



# Technical Handbook

## For PIPL Development Department

The document will guide for technical guidelines within organization.



## **Overview:**

The document will guide to an individual for the “Technical Guidelines” to follow within organization. This version of the document will define the steps / guidelines to follow for technical stuff.

## **Points described in the document:**

The points covered in the document are as below. For each point, you can view brief description on point’s section (defined in the document itself)

- 1. Project Management Software**
- 2. Trouble Shooting**
- 3. Role of Technical Lead**
- 4. Role of Team Lead**
- 5. Process of completing a project**
- 6. Importance of deadlines**
- 7. Performance ratings from team leads:**
- 8. Task completion**
- 9. Daily updates via PM / Email**
- 10. Coding standards**

## ◆ Project Management Software

- **What is “Project Management Software”?**

“Project Management Software” (now onwards called as “PM Software”) is online website facilitated internal project related communication. This software gathers all project related information i.e. Project Requirement Document, Design Requirement Document, Client’s feedbacks, internal communications (asking queries, assigning tasks etc). This is the main source of information for all the projects.

- **How to use PM software?**

In order to use PM software, an individual must have login credentials. These login credentials are assigned to individual when he joins to PIPL. Using these credentials, an individual can login to his account’s Dashboard. Where an individual can view “My Projects”, “View Profile”, “Logout” menus. Each of these menus are described below,

### 1. My Projects

This menu will facilitate user to view “Projects” assigned to him / her. By Default project is displaying with his “Project Code” (an unique number associated with respective project). User can navigate to respective project by clicking on it. This will facilitate him / her below features.

#### 1.1 Dashboard

This page will display all recent activities related to respective project (i.e. A file is uploaded, Client’s Feedback etc). The overview section contains a short description for respective project.

#### 1.2 Messages

This section displays all messages posted on the respective project. A user can also post message by clicking “Add Message” link. “Add Message” link will redirect user to “Message” page, where he / she can enter Subject (must be unique) and message content. If desire, he can also upload a file (single file only, in case of more than one file, the user must create ZIP for it). The same page contains all other members for respective project, so the user can select them to send automatic email notification for respective message. On submit the message get recorded in the database and can be viewed in “Messages” section. The user can also place “Comments” on respective message which will have same features as “Add Message” section.

#### 1.3 Files

This section contains all the files of respective project. Revisions for a file can also placed for the same.

#### 1.4 Tickets

This section allows user to post a ticket for Technical issue. This ticket can then viewed by Team Lead / Tech Lead / Collogue and sorted out on the fly. It is important to close the tickets and also we should close the tickets in least possible time.

## 2. View Profile

This section allows user to change his profile information for his PM account (the password only)

## ◆ Trouble Shooting

This section defines guidelines for troubleshooting on technical difficulties occurred within project development.

Troubleshooting separated within two sections,

### 1. Technical problem

For technical problems such as issue in some coding snippets, solution on functional points, new technology / feature which is not handled by respective user previously, below are steps to follow,

- a. An individual must be clear for the issue raised / technical difficulty and must try to tackle it by himself / herself for a defined time (might be for 15 / 30 minutes)
- b. If the issue cannot be resolved within defined time, the user MUST “Open a Ticket” specifying the issue. By default the ticket will reviewed by respective Team Lead. If respective Team Lead cannot tackle the issue within defined timeline (i.e. 01 hour), the issue then forward to Tech Lead of PIPL.
- c. On completion / fixes on respective issue, the individual who open the ticket MUST close it by clicking “Close Ticket”.

### 2. Functional Problem

For functional problems such as “Functional Flow” of any module in requirement document, an individual must clarify it from respective “Team Lead”. If Team Lead cannot solve the problem, the respective issue should navigate to “Tech Lead” of PIPL. If Tech Lead cannot answer the query within defined time (i.e. 30 minutes from the point asked), the same will then forward to BA team for clarification.

## ◆ **Role of Technical Lead**

Technical Lead (Tech Lead) of PIPL, is the individual who hold responsibility to planning for projects, solution on technical and functional queries and tracking for project executions. The roles and responsibilities of Tech Lead described as below,

1. Meet project deliveries and take responsibility of fastening the development process
2. Retain Existing clients (by providing professional quality software)
3. Find out “Out of Scope” points and make them billable
4. Help in developing a great development team with positive spirit
5. Standardize the codes and documentations
6. Help transform projects into products
7. Conduct technical sessions on code improvements & other technical areas

## ◆ **Role of Team Lead**

“Team Lead” is the person, who holds responsibility to project execution within Team of involved employees. The team lead is responsible to understand project requirement, plan the tasks of involved team members and review the same. The roles and responsibilities of Team Lead is described as below,

1. Project Understanding and make understanding document for the same
2. Make Task Breakdown for modules involved in the project and get approval for the same
3. Review coding structure written by team members and correct wherever required
4. Manage Time schedule for the project and track for the same
5. Analyze work done by team members on daily basis
6. Give performance ratings for team members
7. Approval of Leave requests for team members
8. Arrange sessions for technical points
9. Solve Technical / functional issues faced by team members
10. Track on points asked by client (for in scope / out of scope differentiation)
11. Analyze the project requirement and client view
12. Create Task list for the project according to WBS
13. Control on the task being done properly
14. Allocate resources to the task and review for the task
15. Integrate the task into project

### ◆ Process of completing a project

This section defines a flow for project execution within organization. An individual MUST follow the guidelines as defined in below steps,

1. At initial stage, Marketing Team discusses the project requirement from client and defines timeline for respective project. On finalization, the Marketing Team hand over all requirement to BA team, who then creates “Functional Document” for scope of the project which involves all the agreed functional modules for respective project. On confirmation the functional document gets forwarded to Team Lead for further analysis. (with new project code for respective project). In this stage, the project has been started and “Analysis Phase” of respective project must be cover. For this stage respective Team Lead and Developer MUST analyze the functional flow document in deep and clarify the queries if any. The work should not start until all the functional modules are clear to everyone on respective project team. In this stage, the respective Team Lead must arrange a “Project Understanding Meeting” with “BA” team. The project can start with the functionality those are clear, holding the project for any specific functionality clarification makes us loose time as client count the start date from the day we accept the project.
2. On clarification and confirmation of involved functionality of the project, respective Team Lead creates “Task Breakdown” document for involved functionality. If respective project is similar to any of existing completed project, the Team Lead must make sure the functionalities to copy from reference code and indications for change according to need of the project requirement.
3. The database structure must create by Team Lead of respective team by taking care existing reference project’s structure. Once database structure defined, it should get confirmed by Tech Lead before proceeding. On final confirmation, the database tables cannot be altered unless confirmation from Team Lead / Tech Lead.
4. The Team Lead, then create Task list to be done on daily basis for respective project and assigns tasks for individual team member with defined date and time along with Test Rules defined by QA department for respective task.
5. A team member will work on Tasks assigned and make sure they are completed on defined time with confirmation from QA team. If confirmation not received from QA team, the respective tasks cannot treat as completed task and henceforth the Team Member will receive rating for respective task accordingly. The team member ensures before the delivery of the task, he checked the task properly according to Test Rule provided by Team Lead.
6. An individual MUST make sure he read and checked “Checklist Document for Starting Coding” before write coding. If he fails to follow that, respective Team Lead can take action on respective team member accordingly.
7. An individual MUST make sure he read and checked “Checklist Document for sending update to QA Team”. If he fails to do that and QA team detects redundant bugs, the individual will get rating from QA team which will effect on his Performance Criteria.
8. The individual MUST make sure his coding structure is following guidelines as defined in “Coding standards” section.

9. The individual MUST make sure he will raise tickets for unsolved functionalities / technical points, if he is not capable to do so.
10. The individual MUST make sure on completion of the project, he will keep the copy of local PC in a separate directory which can not be override (i.e. MUST be same with Live Version)

### ◆ **Importance of deadlines**

In every project “Deadline” is the most important factor. The deadline is the term on which date the project will get completed with all in scope points and with proper QA. If in any case the deadline is not meet by a team, it affects on several internal and external (client side) plans. The deadlines are the actual time frame which a project team should meet in any case, if the project do not complete in defined deadline, it directly effects on internal planning and costing of the project. If any team fails to meet the deadline, it will effect on entire team’s performance.



## ◆ Performance ratings from team leads

The team lead is having responsibility to review coding standards written by team member and task completion on daily basis. According to these criteria the Team Lead should report to Tech Lead, HR and Director of PIPL the performance ratings for his team members. According to ratings given by Team Leads, respective individuals performance will get counted which will have effect on his / her Appraisal Process.

The performance ratings can give according below criteria,

1. Task Completion
2. QA report for the task (for redundancy of bugs, level of bugs etc)
3. Coding Standards implemented in the code
4. Overall behavior within a Team
5. Technical compatibility of an individual in respective domain / tasks (advanced techniques used for better performance)

### ◆ **Daily updates via PM / Email**

Each employee in PIPL is bound to send “Daily Work” update to management team. The daily work update should be transparent and true. False updates for daily tasks may cause respective action from Team Lead / Management. The daily update must have in defined format and should contain brief description for completed task and not completed task must have appropriate reasons which can justify non completion of the task.

The daily update mail must go to below stated persons,

1. Team Lead of respective individual
2. Tech Lead of PIPL
3. HR of PIPL
4. Directors of PIPL

The daily updates can be sent via PM software (within respective project by Messaging Feature) or via email.

## ◆ Coding standards

This section of document defines the coding standards that MUST follow by individual.

Failure to follow, may cause appropriate actions. The responsibility to review the standard practices of coding will be Team Lead and Tech Lead.

Coding standard is required because there may be many developers working on different modules so if they will start inventing their own standards then source will become very un-manageable and it will become difficult to maintain that source code in future.

Below are reasons to follow coding standards,

1. Another Team mate must understand the code you write. A coding standard acts as the blueprint for all the team to decipher the code.
2. Simplicity and clarity achieved by consistent coding saves you from common mistakes.
3. If you revise your code after some time then it becomes easy to understand that code.

Below are guidelines to be followed by individual,

1. **Proper Naming Conventions**  
An individuals must give proper names to Files (php, js, css etc), variable he is using in his program. "ABC", "XYZ" etc variables are not allowed to be used in the code.
2. **Proper Folder Structure**  
An individual must create proper folder structure for his code, which can easily found and logically handled. For example, all JavaScript files must be kept in "JS" folder, the module specific files in respective module name's folder like "admin", "users", "products" etc.
3. **Indenting and Line Length**  
Use an indent of 4 spaces and don't use any tab because different computers use different setting for tab. It is recommended to keep lines at approximately 75-85 characters long for better code readability.
4. **Control Structures**  
These include if, for, while, switch, etc. Control statements should have one space between the control keyword and opening parenthesis, to distinguish them from function calls. You are strongly encouraged to always use curly braces even in situations where they are technically optional.
5. **Function Calls**  
Functions should be called with no spaces between the function name, the opening parenthesis, and the first parameter; spaces between commas and each parameter, and no space between the last parameter, the closing parenthesis, and the semicolon.

## 6. Comments

C style comments (`/* */`) and standard C++ comments (`//`) are both fine. Use of Perl/shell style comments (`#`) is discouraged.

## 7. PHP Code Tags

Always use `<?php ?>` to delimit PHP code, not the `<? ?>` shorthand. This is required for PHP compliance and is also the most portable way to include PHP code on differing operating systems and setups.

## 8. Variable Names

Use all lower case letters

Use `'_'` as the word separator.

Global variables should be prepended with a `'g'`.

Global constants should be all caps with `'_'` separators.

Static variables may be prepended with `'s'`.

## 9. Alignment of Declaration Blocks

Block of declarations should be aligned

## 10. One Statement Per Line

There should be only one statement per line unless the statements are very closely related.

## 11. Short Methods or Functions

Methods should limit themselves to a single page of code.

## 12. Function arguments

Since function arguments are just variables used in a specific context, they should follow the same guidelines as variable names.

It should be possible to tell the main purpose of a function just by looking at the first line, e.g.

`get_user_data($username)`. By examination, you can make a good guess that this function gets the user data of a user with the username passed in the `$username` argument.

Function arguments should be separated by spaces, both when the function is defined and when it is called.

However, there should not be any spaces between the arguments and the opening/closing brackets.

## 13. SQL code layout

When writing SQL queries, capitalise all SQL keywords (`SELECT`, `FROM`, `VALUES`, `AS` etc.) and leave everything else in the relevant case. If you are using `WHERE` clauses to return data corresponding to a set of conditions, enclose those conditions in brackets in the same way you would for PHP if blocks, e.g. `SELECT * FROM users WHERE ( (registered = 'y') AND ((user_level = 'administrator') OR (user_level = 'moderator')) )`.

## 14. Quoting strings

Strings in PHP can either be quoted with single quotes (') or double quotes ("). The difference between the two is that the parser will use variable-interpolation in double-quoted strings, but not with single-quoted strings. So if your string contains no variables, use single quotes and save the parser the trouble of attempting to interpolate the string for variables

## 15. Shortcut operators

The shortcut operators ++ and -- should always be used on a line of their own, with the exception of for loops. Failure to do this can cause obscure bugs that are incredibly difficult to track down.

## 16. Use constants where possible

If a value is not going to change throughout the execution of your script, then use a constant rather than a variable to define it. That way, if you do change the value by accident, the PHP parser will output an error and allow you to fix the problem, without it causing any unforeseen side effects.

Remember that constants should never be enclosed in strings, either single or double. You must always use concatenation if you wish to include a constant's value in a string.

## 17. Turn on all error reporting

If you write proper object orientated code and make sure that each class doesn't produce any errors, you should be able to make your program as a whole run without errors under normal circumstances.

Of course, on a production site you might want to turn off all errors, or at least redirect them to a file, to avoid admitting to users that your scripts are broken in some way. That's perfectly fine and in many cases the recommended action to take. So long as your scripts work with all error reporting turned on, it doesn't matter where they are deployed.

## 18. Proper Database Queries

You must make sure you will write proper queries to database which may have good results. Using "\*" may cause a lot of issues in speed of the database. When using classes of database, do not create several instances of connections, instead try to use "clones" of existing connections which will override "Maximum Connections" issues. The queries should be written in such a way to handle "Sql Injections" so each input from user should be parsed in proper database formats, by using `mysql_escape_string`

## 19. Using Views (front end pages)

The static labels (like username, password etc texts) should not directly written as a text, instead they should keep in a variables and then display those variable by echo / print. Static text on page is STRICTLY PROHIBITED.

## 20. JS and CSS files

A developer should keep ONLY NECESSARY JavaScript files, unwanted JS files may cause page speed issues.

## 21. Images and Thumbnails

All the dynamic uploaded images (like product photos, user's pictures) must not display as it is on the page, it should be make appropriate thumbnail (the size MUST be defined according to section of the page). The thumbnail MUST be in proportion and should not display messy.