Act like a software engineer specialized in software architecture and codeigniter technology, heading a team packed with developers and expertise required for this task. Hire new experts per required to add to your team as the task require.

In documentation/Perfex crm plugin document.docx file in the github repo, you will find main references about how to develop a module for PerfexCMS platform.  
  
  
In /documentation/ folder in the github repo, you will find 1) documentation/tree\_directory.txt: and that is the files directory tree of PerfexCRM, and 2) two files documentation/codebase-1.txt and documentation/codebase-2.txt together represent the full codebase of PerfexCRM starting with each file’s location in the codebase then the full code of the file (except some binary files).  
  
In documentation/help.perfexcrm.com you will find the full system manual of Perfex CRM for furthur information that you can find as needed. The name of the folder indicates what you can expect to find inside it. A good place to start is the help.perfexcrm.com’s files tree you finf here: From

First task required is for you to fully learn everything that has to do with modules architecture and features in the codebase so you become fully equipped to develop highly complex new modules that I upload into the system and test. Start with documentation/Perfex crm plugin document.docx file for a basic idea. documentation/Perfex crm plugin document.docx file select the relevant files/pages to modules in the elp.perfexcrm.com so you can read it and learn from it.  
  
Keeping in mind that the codebase follows codeigniter architecture, from the documentation/tree\_directory.txt pick the relevant core files in PerfixCRM you need to know as you develop modules for the system. Then study the existing modules to master modules architecture, how it is developed to add tables and so in the database, how to integrate module with core roles and permissions, themes, other modules and core features,custom files, etc...  
  
Forensically learn it and understand it asif you coded it character by character. Take all the time you need, this is a critical stage and no furthur work can be dine successfully if there was any flaws in the execution of this stage. Considering that it is a relatively big codebase, you will need to systematically work on it so you create a proper pyramid-grape-nodes-like understanding where at the end you have a condensed overview of the whole codebase that you keep in your context memory and document it in a file you create in a new folder you create called workspace, but it has the mapping and logical connections required to take you deep to different parts of the codebase based on what is required, what you are changing in the code, and where such changes may impact or demand connected changes.

Use the workspace/ folder as your working area. For example, add notes file (notes for yourself of things need to pay attention to at some point or get back to to do something, just so you dont forget it later), to-do lists, functional mapping file (where you map each of the platform features to the bits of the code relevant to it in what files and what code snippets in that file), bugs list (where you can see bugs require fixing), optimization file, enhancement file, suggested features file, platform flows document, and whatever tooling you need to have to help you accomplishing the work beyond your current limitations.   
  
but most importantly create a module architecture document where you put all what you learn about modules structure, commands, features, calls, etc..., how are features are developed in different modules, notes to keep in mind when you develop your own module, etc... So when we start developing new modules, you have all the know-how, guidelines, and references you need to get the job done perfectly.

After you take all the time you need and all the thinking power you can have to reach the full understanding point, we can move to another task. But take your time to forensically get this first task done and I will be checking your work regularly. Make sure to push all the work you do into the repo.

Impress me!  
  
I will push this prompt into the repo root as well so you get back to to whenever needed, as a task reference and reminder.