**PM Tool Backend Technical Document**

It is a project management tool which currently using Kanban-like board view. Allows end-user to create projects, teams, lists, card, checklist etc. Backend is created using .Net Core 5. The backend uses entityframework core as its ORM and repository pattern (without Unit of Work) to access DB context. Also, .Net Core 5’s provided automapper library is used to map objects where possible. Lastly signalr is used for push notifications.

Follow the link below to open swagger documentation.  
[Swagger URL](http://sandbox.aimviz.com:6444/index.html)

**Checkout below links to get started.**

* .Net Core Web Api: <https://www.toptal.com/asp-dot-net/asp-net-web-api-tutorial>
* Entity Framework Core: <https://docs.microsoft.com/en-us/ef/core/>
* Repository Pattern: <https://www.c-sharpcorner.com/article/generic-repository-pattern-in-asp-net-core/>
* AutoMapper: <https://code-maze.com/automapper-net-core/>
* Signalr: <https://docs.microsoft.com/en-us/aspnet/core/signalr/introduction?view=aspnetcore-5.0>

**Project Structure**

It is a two-layered architecture i.e. its contains “Application Layer” and “Data Access Layer”.

1. *DAL:* Contains DB context, entity models, DB migrations and repository class.
2. *APPLICATION LAYER:* Contains all the business logic implementation.

***DAL***

*PMToolDbContext.cs:* Contains dbset and public-key mappings.

*[Models]:* Contains all the entity models.

*[Migrations]:* Contains database-migrations created by the ORM.

*[Repository]:* Contains generic repository interface and its implementation.***APPLICATION LAYER (For More Details Refer Code Documentation***

*[Authorization]:*

* ActiveUserAuthorizationHandler.cs: Handler for active-user authorization policy.
* ActiveUserAuthorizationRequirement.cs: To check active-user policy requirement.
* JwtClaimAccessor.cs: Used to get claim value generated after authorization.

*[Controllers]:*

* BaseController.cs: Has attributes applied needed for each controller and each controller is extended from this class.
* AuthController.cs: Contains all authorization endpoints and their implementation. [Check appendix](#AuthorizationEndpoints) to see which endpoints are included.
* CommonController.cs: Contains shared endpoints i.e. endpoints used in multiple locations.
* UserController.cs: Contains endpoints to get, edit or delete a user.
* TeamController.cs: Contains endpoints for list, add, edit, delete or filter a team.
* ProjectController.cs: Contains endpoints associated with following views (project-list, create-project, project-permissions).
* ChallengeController.cs: Contains endpoints for ‘project-detail’ view i.e. add/delete card, add/delete list, view’s subheader and move card.
* CardController.cs: Contains endpoints for all card-view functionalities e.g. add/remove member, attachment, checklist, comment, etc.
* NotificationController.cs: Contains endpoints for notification panel.
* FaqController.cs: Contains endpoints for faq page and contact us/feedback form.

[Hubs]:

* NotificationHub.cs: Hub for signalr used for notifications.

[Mappers]:

* AutoMapper.cs: Automapper profile class.

[Templates]: Contains templates and images used in emails.

[Models]:

* [DTOs]: Contains ‘Data Transfer Objects’ used for sending response data.
* [Enums]: Contains “Enum.cs” which contains all the enum classes.
* [General]:
  + Constants.cs: Contains strings to be used as messages for user.
  + CustomEmail.cs: Contains email methods specific to the project.
  + Email.cs: Contains generic email methods.
  + EncryptDecrypt.cs: String encryption/ decryption helper class.
  + Helper.cs: Contains general helper functions e.g. serialize/ deserialize objects, date conversion.
* [Request]: Contains request objects that are used as controller’s formal parameters.
* [Response]: Contains responses that are in some cases returned from controller instead of generalresponse.
* [Services]:
  + Contains services for each controller to implement business logic.

**APPENDIX**

* Active user policy: Verify in addition to token whether the user is valid or not.
* Authorization endpoints includes (login, logout, socialLogin, signUp, accountVerification, forgetPassword, changePassword.