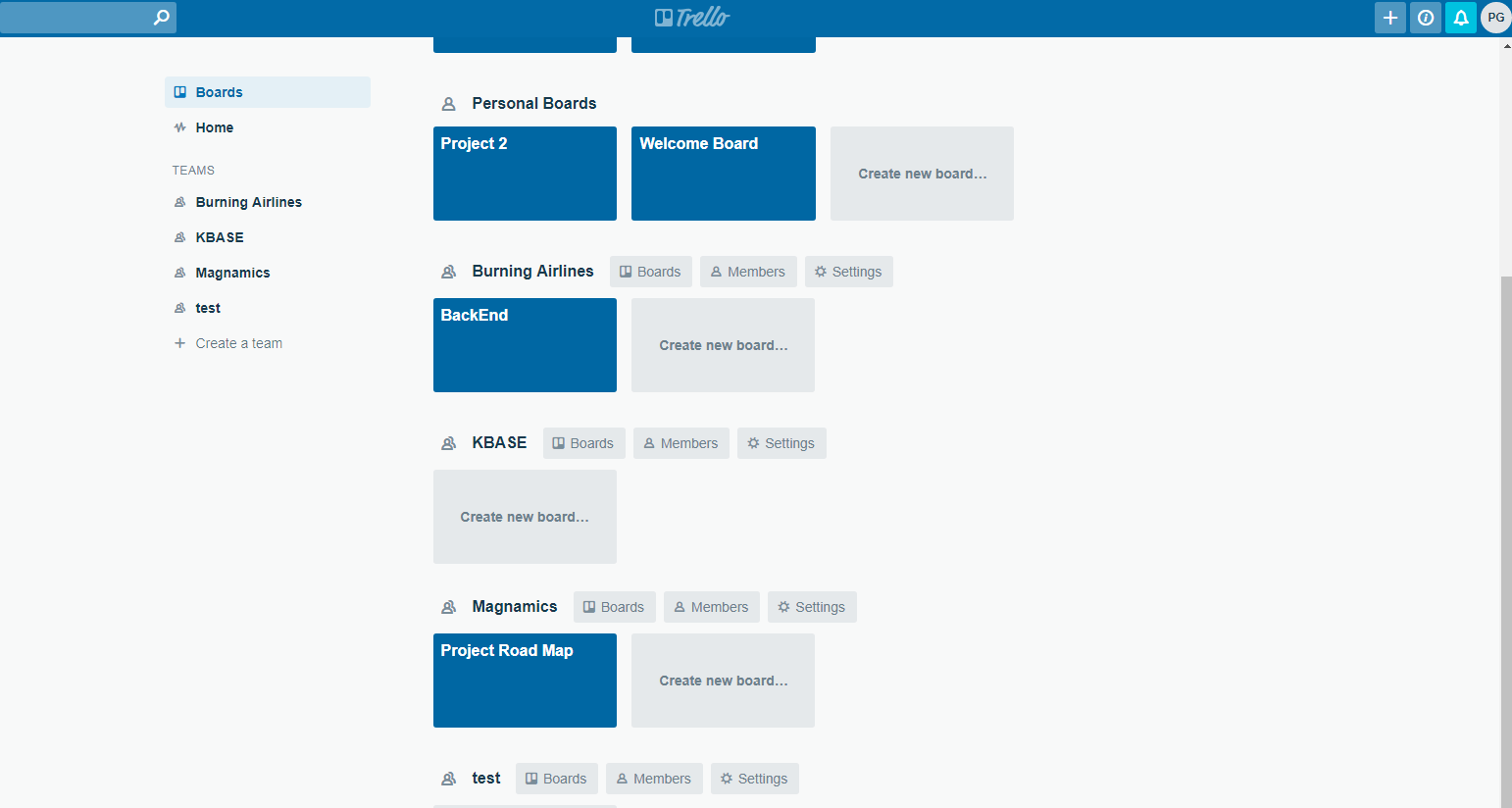
Project: Task Management (back-end and/or full stack)

**Background**

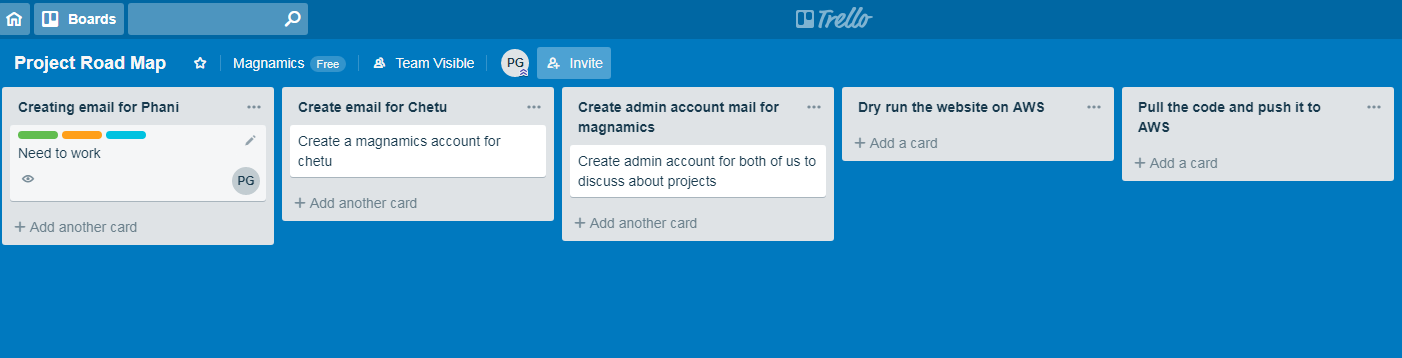
The project mirrors the capabilities of the Trello task management tool. For those unfamiliar with it, Trello is a free web-based service that lets teams structure tasks and assignments in a very flexible way, with an easy to use and intuitive front end written in HTML5.

Here is a screen capture of the Trello user interface with an example board set up:

Trello Home Page: (Boards)



Cards page:



Trello consists of the following constructs:

* **Boards**, which can contain zero to many **Lists**
* **Lists**, which can contain zero to many **Cards**
* **Cards**, which have a mandatory **Title** and can include a number of optional parameters, including: **Description, Due Date, Label** and **Members**
* **Members**, which can be assigned to a card
* Each Board can have up to six **Labels** which can be named on a per-board basis (e.g. a board created for tasks might use labels High Priority, Medium Priority, Low Priority, while one created for a shopping list might use labels Fruit & Veg, Bakery, Meat, Dairy, etc) and assigned to each Card

Actions which can be performed on each item include:

* Boards can be created, renamed and archived
* Lists can be created, archived, renamed and reordered
* Cards can be created, archived, renamed, reordered within a list, or moved to another list
* Members can be created, renamed, archived and assigned to cards
* Labels can be renamed and assigned to cards. Each card can only have one label

**Project Description**

We are not asking you to reproduce the user experience of using Trello, as this is clearly a very large task. We are however asking you to create a REST API using Python, Django and Django REST Framework with a minimal frontend UI which models the behaviour of the system.

**You should:**

* Assume you’re submitting this as production-quality code for review. Write useful comments, modular code, usage of PEP8, testable code etc.
* Use github for your code repository

**You may:**

* Keep the UI minimal to makes things easier for you
* Use any variant of NoSQL or SQL databases

**Bonus points:**

* Host your solution on AWS
* Unit tests in any of the popular Python/Django test framework

We are primarily concerned with the way you structure and organise your code.