UMS Documentation

Technical Doc & Guide

Abdul Mobeen
MOBEENDEV@GMAIIL.COM

UMS - Documentation

Contents

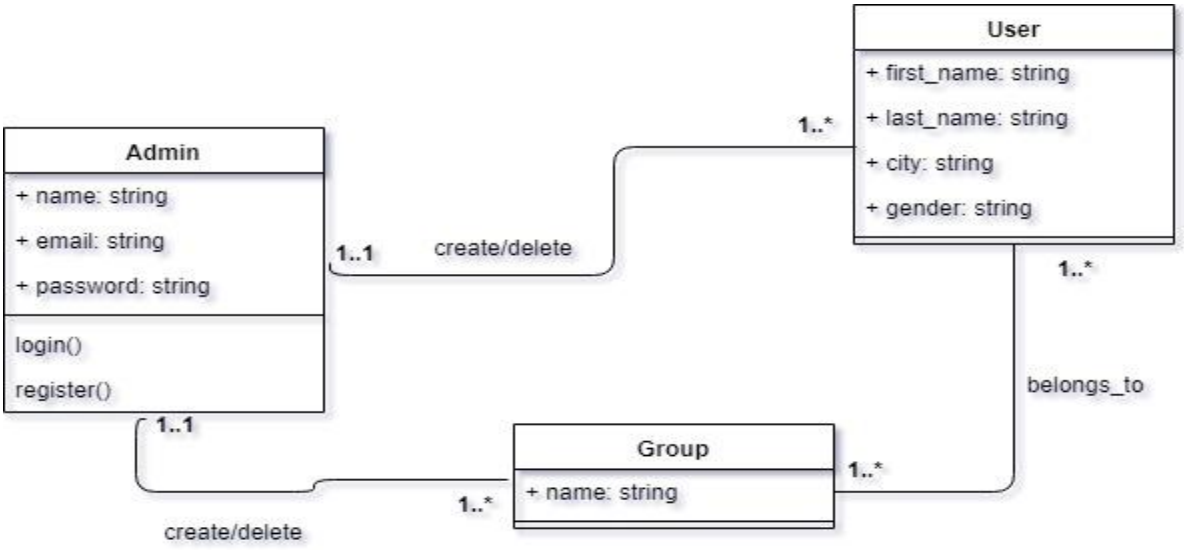
Introduction

This is a Web Application which is a **User Management System** with the following stories:

- A person can register as Admin
- Admin can login into the system
- Admin can add users
    - a user has following attributes
        - first_name.
        - last_name
        - gender
        - city
- Admin can view the User
- Admin can delete users from groups.
    - When admin deletes a user, his/her joined group will be detached but user will not be removed from the DB.
- Amin can attach users to a group they aren't already part of.
- Admin can remove users from a group.
- Admin can create groups.
    - A group has 'name' attribute
- Admin can view the groups with all the user's in that group on detail screen
- Admin can delete groups when they no longer have any users/members

UML

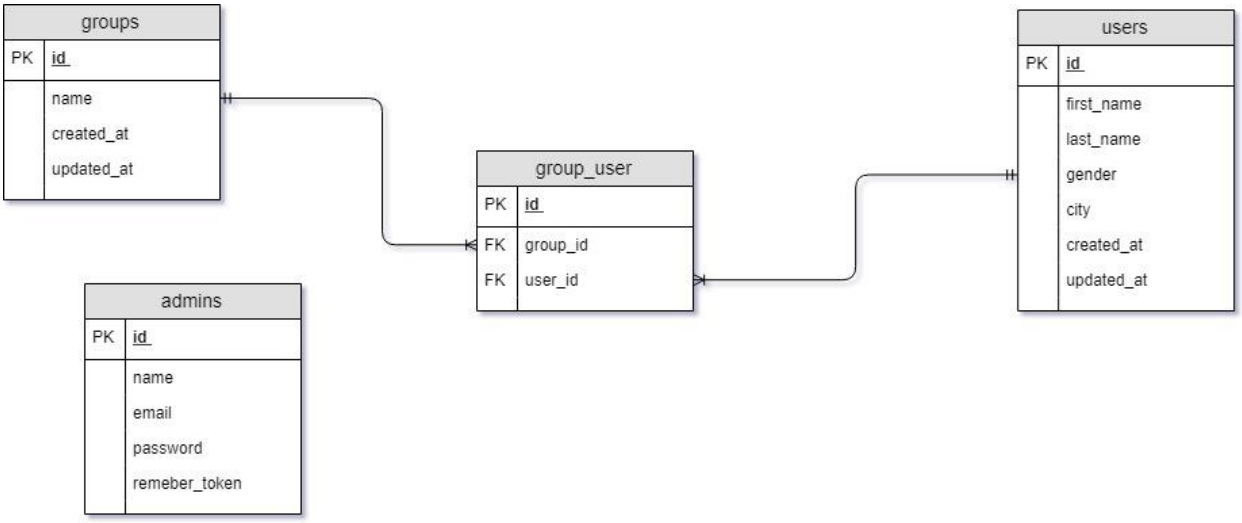
A domain model in problem solving and software engineering is a conceptual model of all the topics related to a specific problem. It describes various entities, their attributes, roles, and relationships, plus the constraints that govern the problem domain



ERD

As this web application uses **user** and **groups** data, so for this task I have used relational database, to store and retrieve the data, figure below shows tables and relationships among the tables which are used for this application.

I have assumed that there's **many to many** relationships between **users<->groups**. i.e. a user can be in multiple groups and a group can have many users.



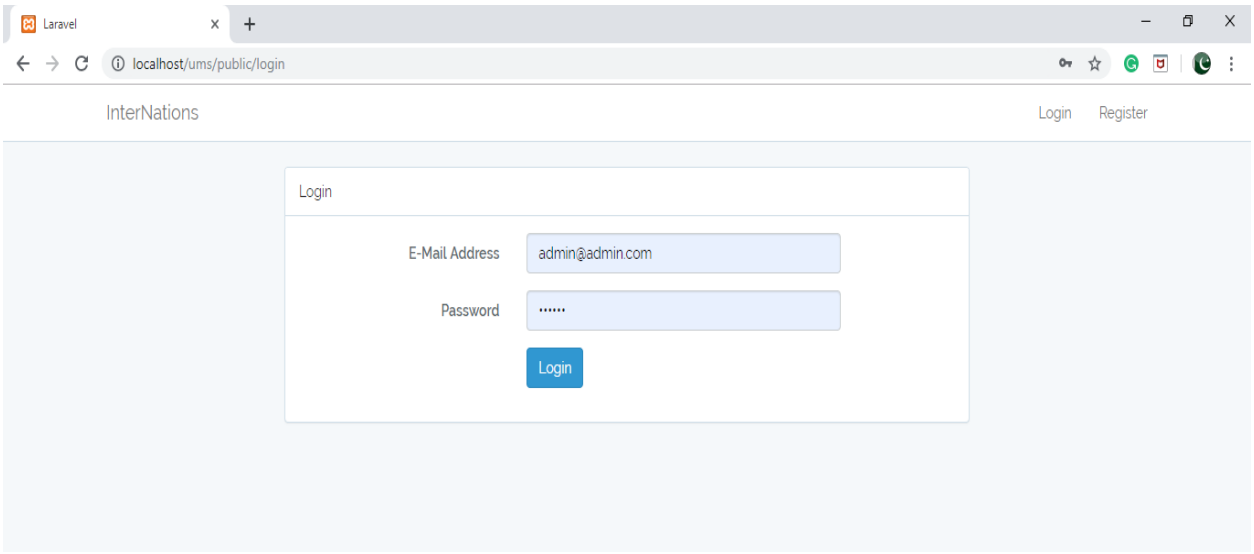
System Architecture

For this task, I have used three tier architecture i.e MVC or Model View Controller. I have also used the repository pattern where possible to create the abstraction for data access layer and to handle the domain objects (users, groups).

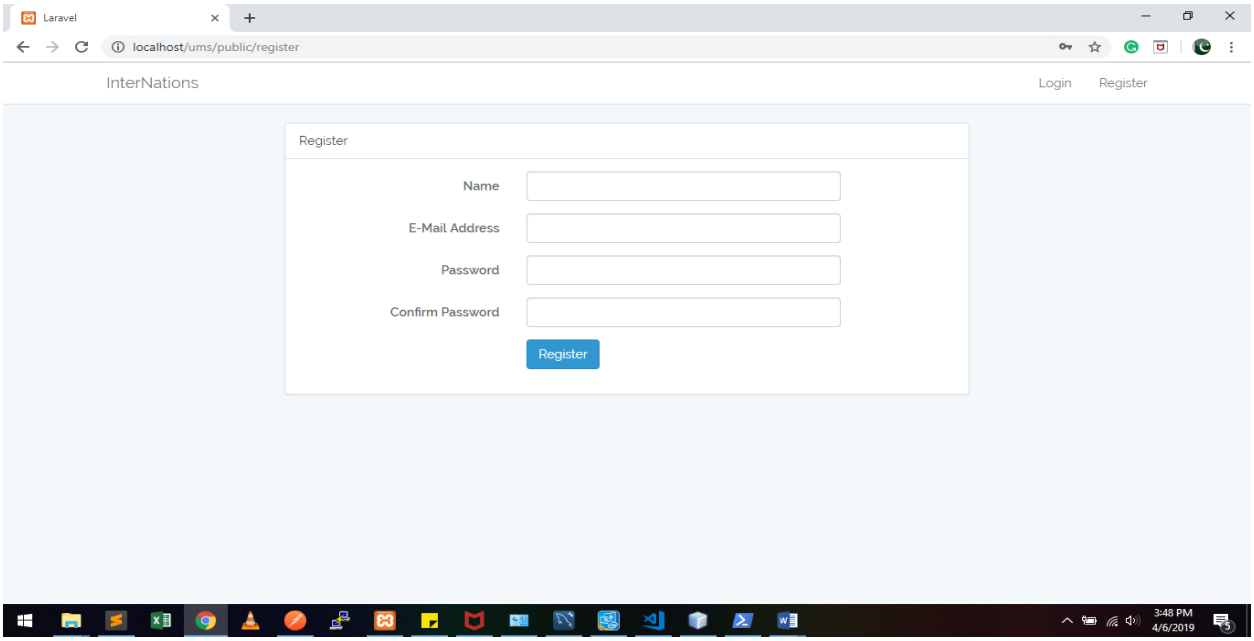
Next you can see the screen for above stories.

Web Application Screens

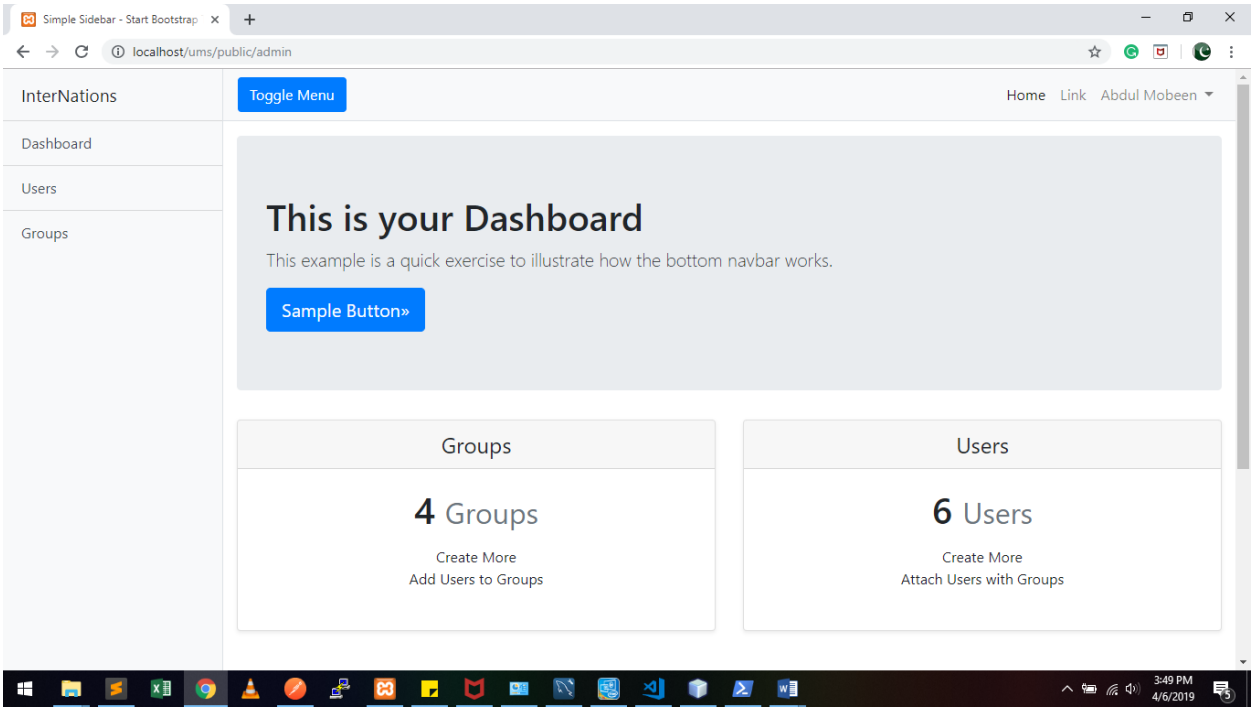
Login Screen:



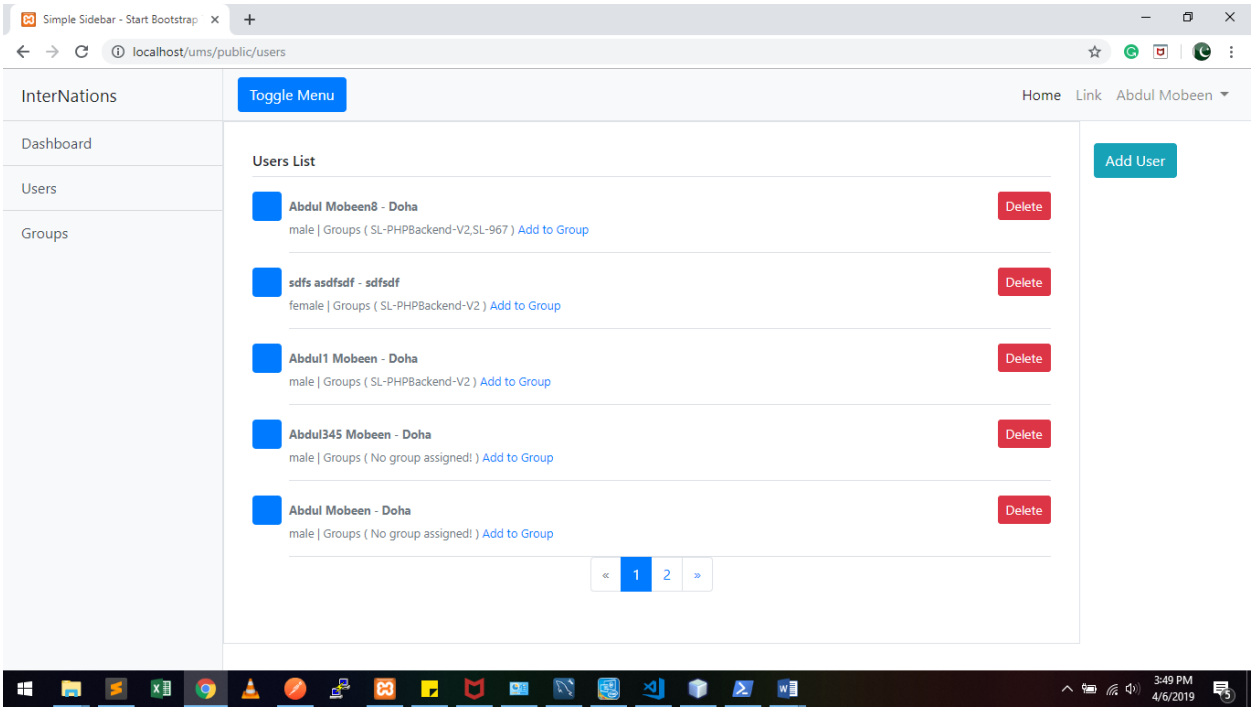
Register Screen



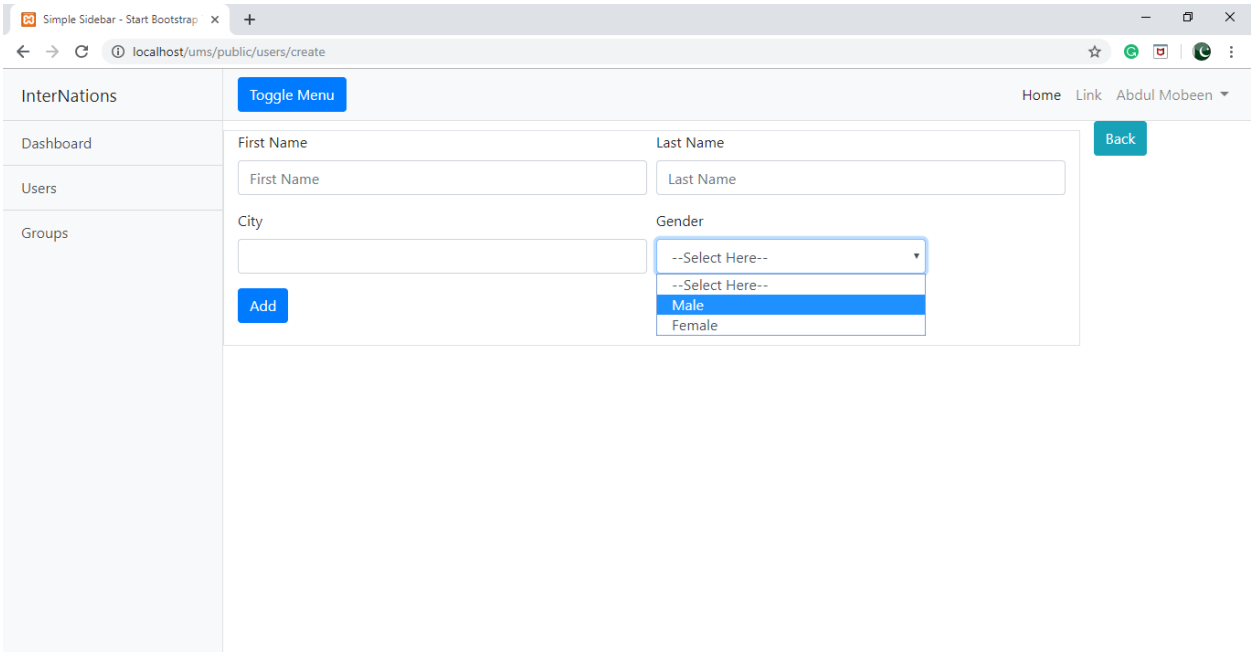
Dashboard



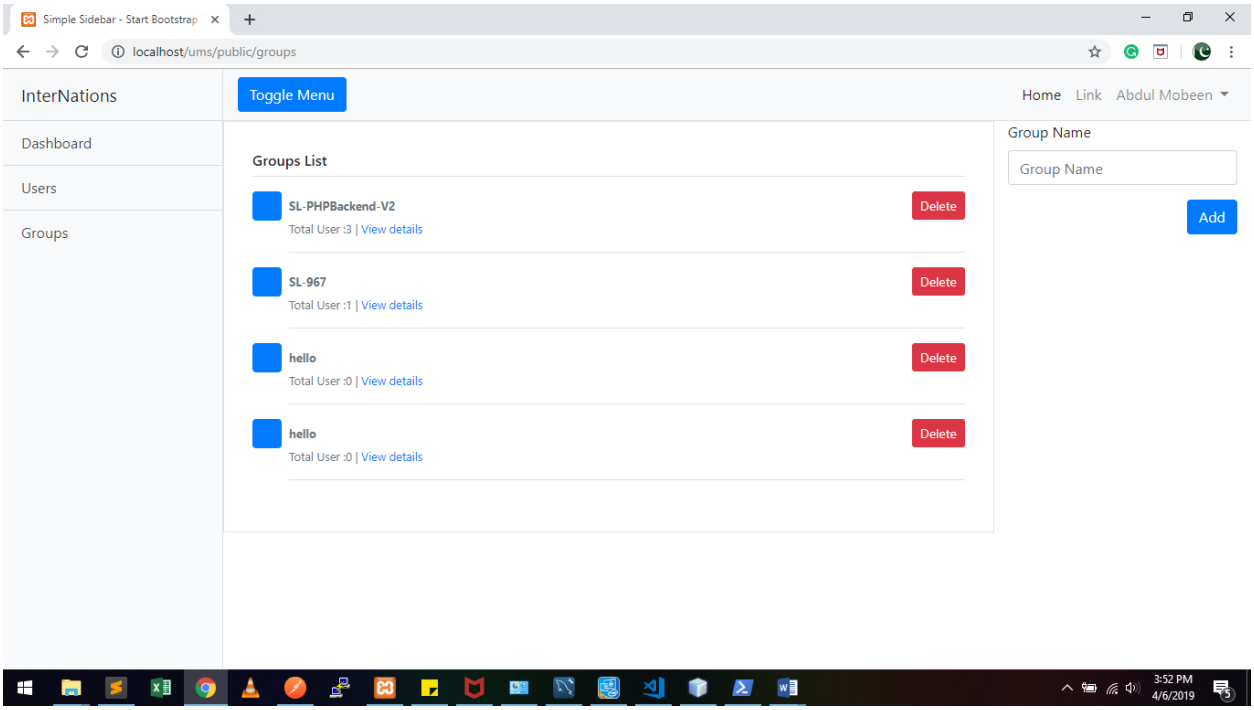
User Listing



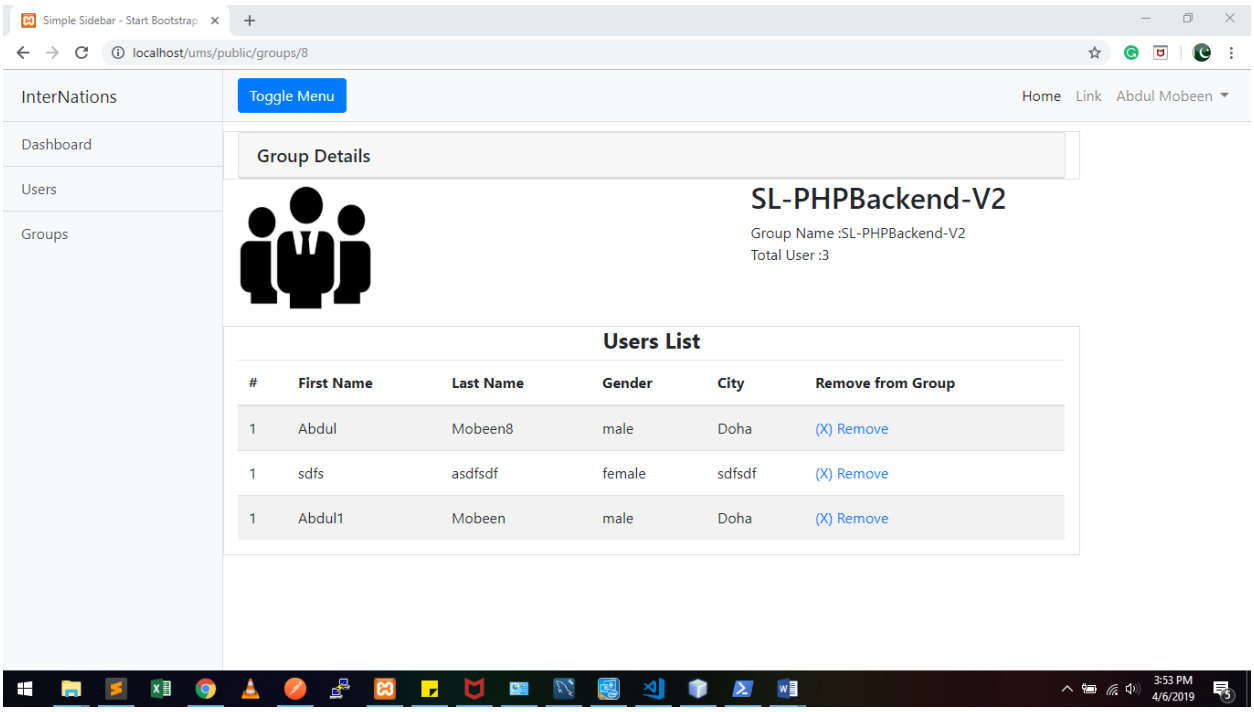
Add New User



Groups Listing and Add Group



View Group Detail



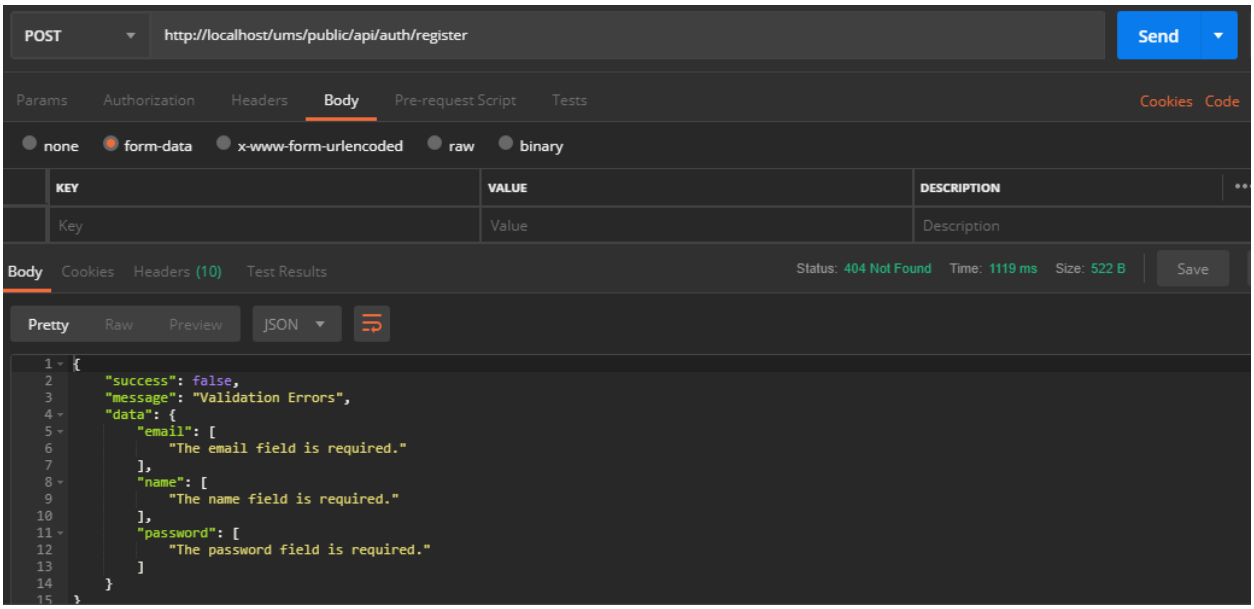
APIs

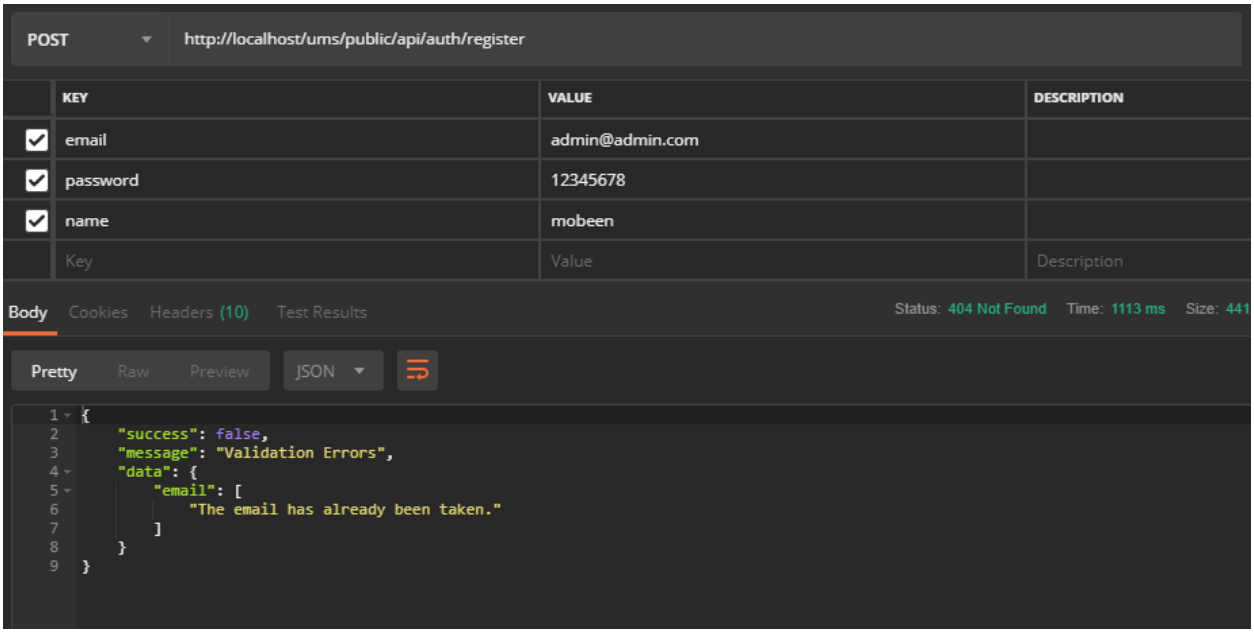
I have only created following APIs just for the demonstration

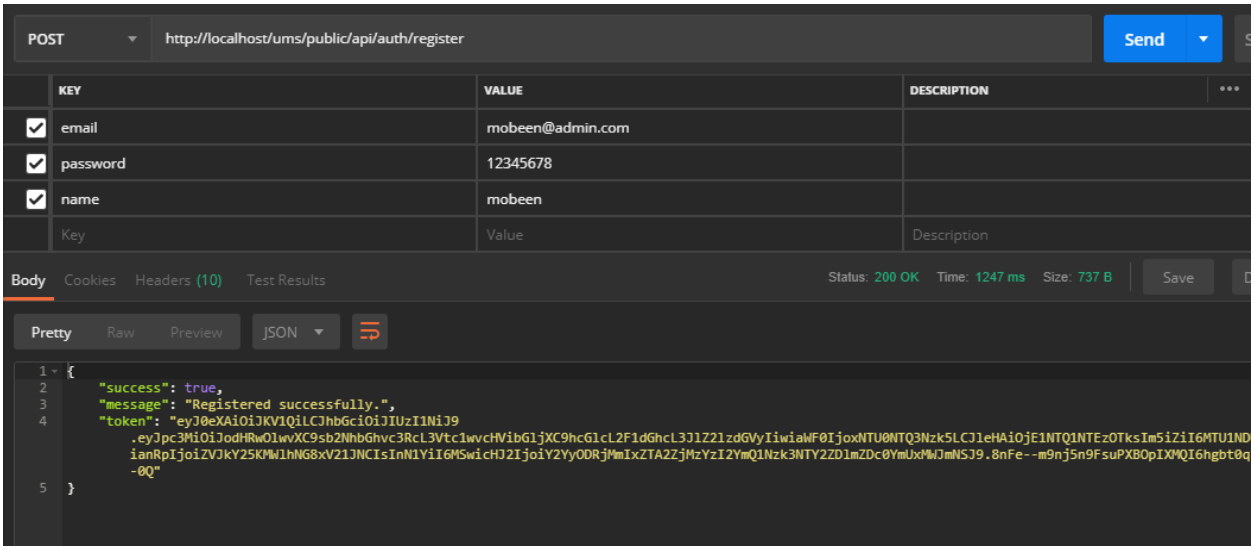
- Admin Login
- Admin Registration
- Get all users with their joined groups
- Get all groups with all the user's in that group

APIs snapshots

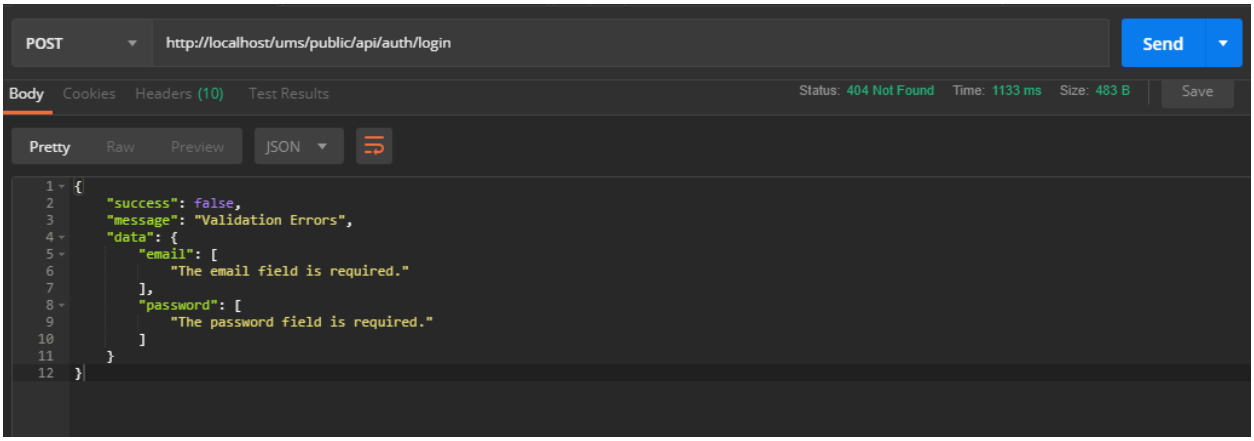
Admin Registration API

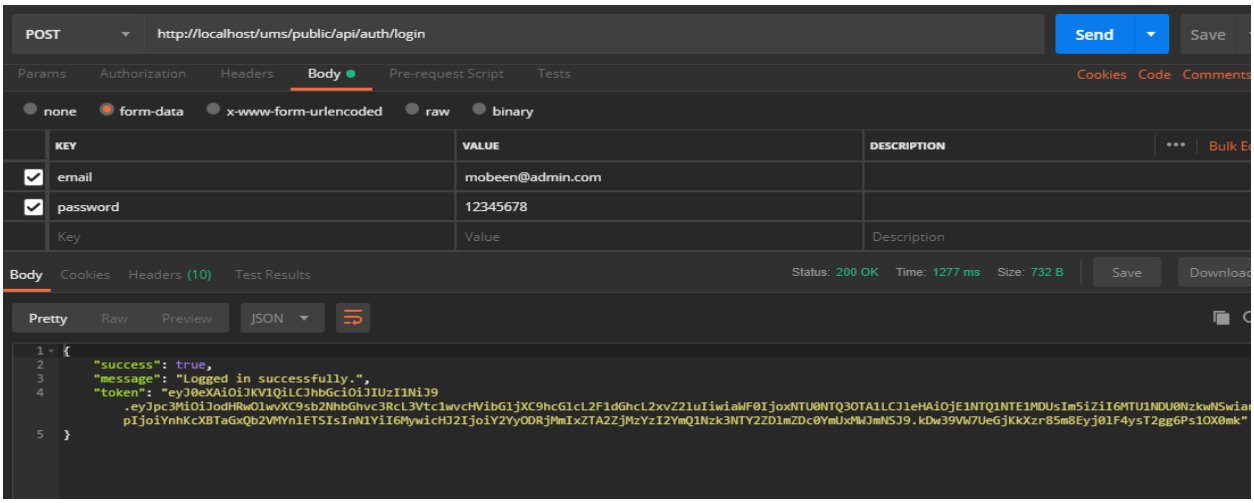




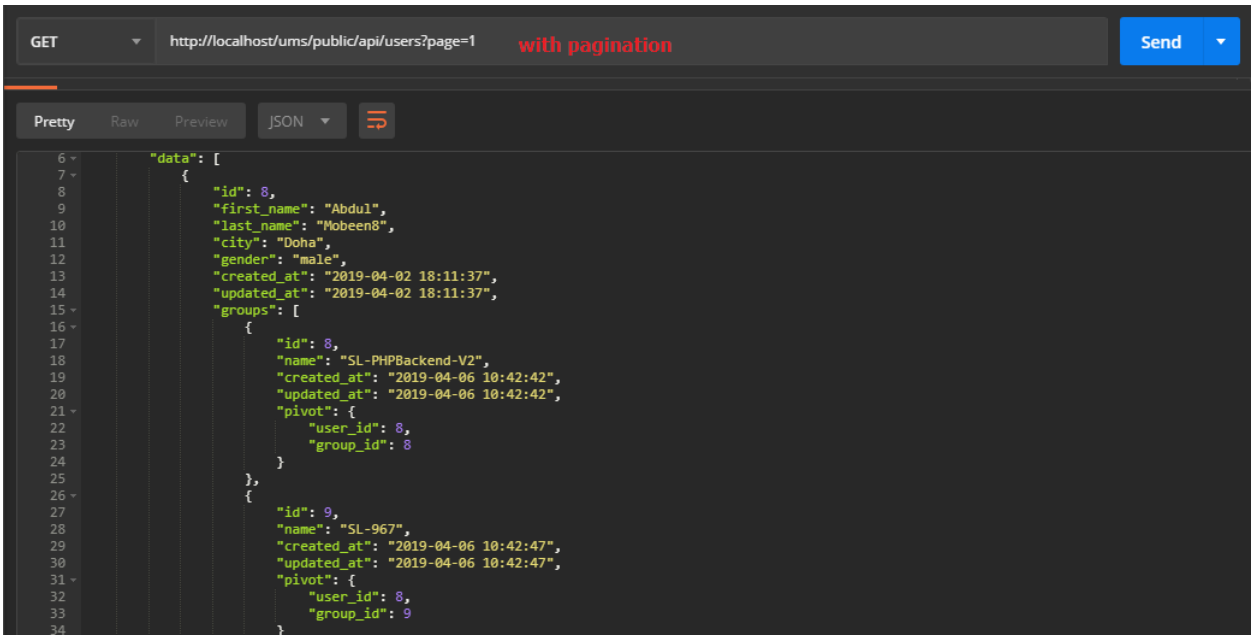


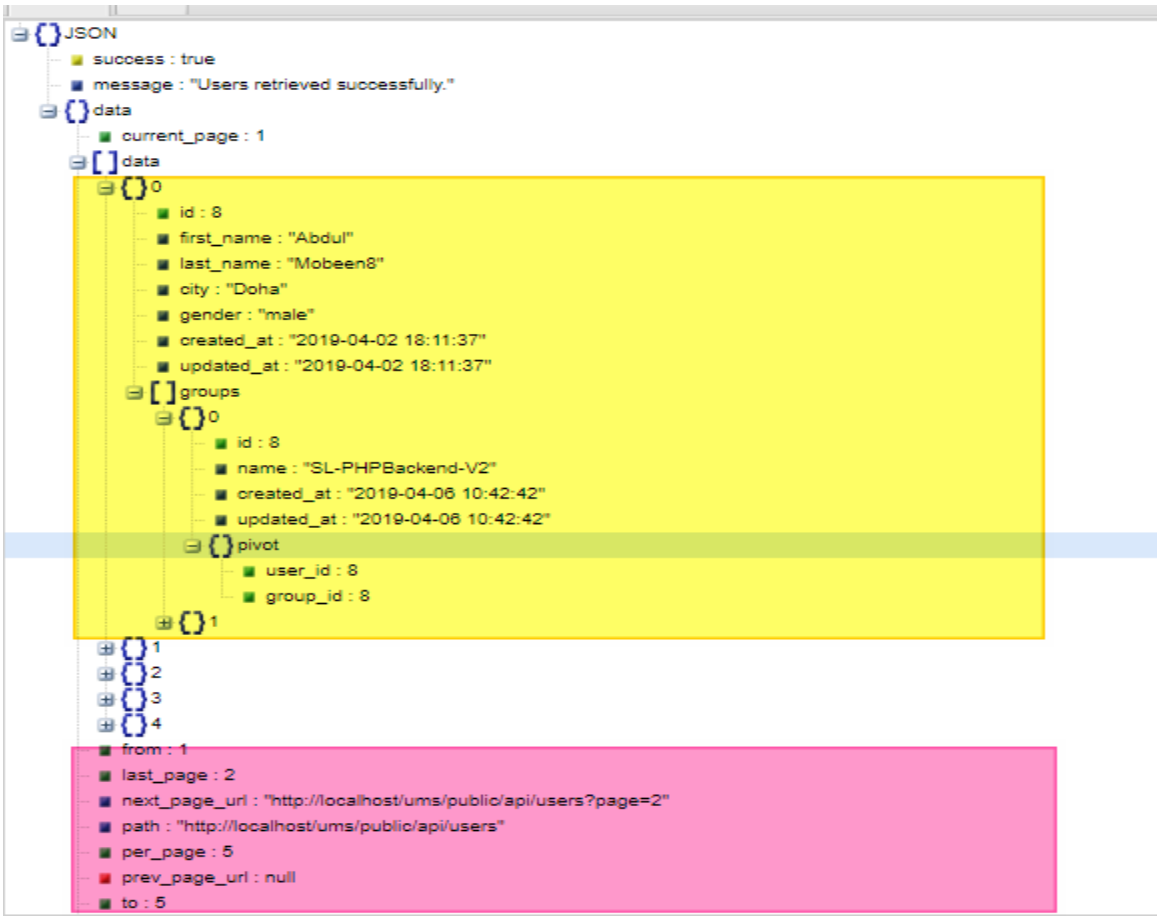
Admin Login API



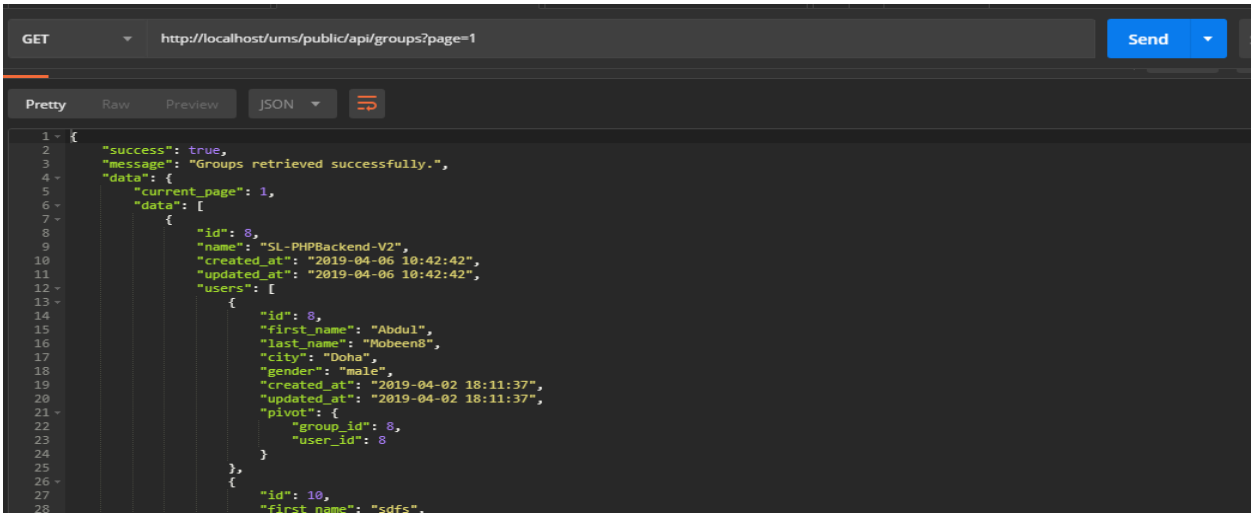


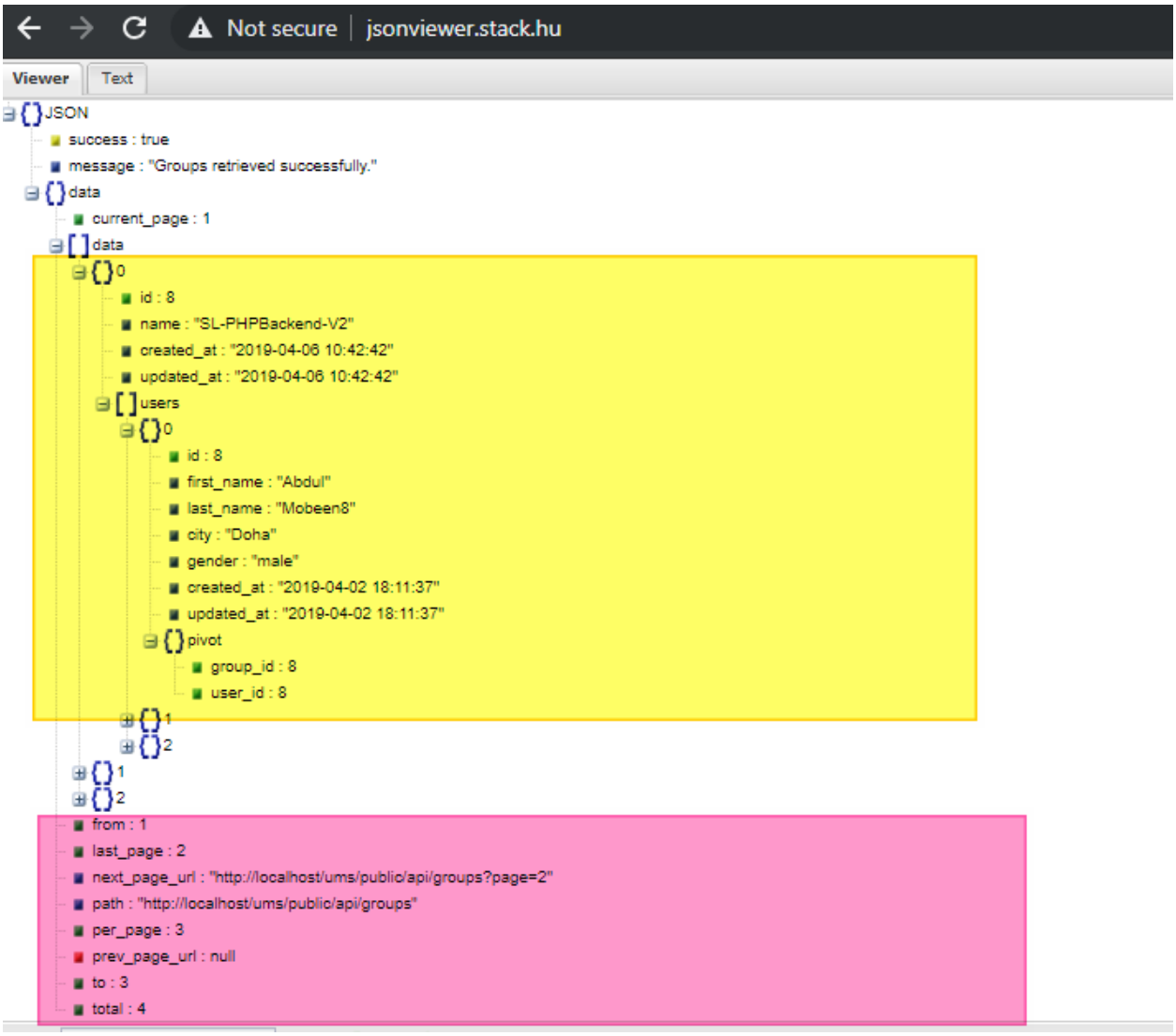
Get all users with their joined groups





Get all groups with all the user's in that group





Sample response objects for Users & Groups API are attached as well.

I have not applied the token check on the GET API calls for ease of testing the APIs.

System Implementation & Tools Technologies Uses

The system is developed using PHP, MySQL as main technologies and Apache is used as the web server to host this web application.

Other details are as follows

Laravel Framework, Bootstrap, Xdebug, Composer, XAMPP

Postman, Workbench, Chrome Browser

Setup Guide

Extract the Zip/Rar File.

Place the “ums” folder into XAMPP (htdocs) or WAMP (www) respective directory.

Open the Project in any code editor.

open the .env file and provide the DB details if yours are different e.g.

DB_CONNECTION=mysql

DB_HOST=127.0.0.1

DB_PORT=3306

DB_DATABASE=ums // this should remain same

DB_USERNAME=root

DB_PASSWORD=

Import the **ums_db.sql** file, it will create the database “ums” in your MySQL along with the required tables.

And hit this url: http://localhost/ums/public/

And you will see the login screen. As you are using it for the first time, just create an admin user using the Register Link and login.

If there is any issue in setting up the application, please let me know.

Thanks.