

Modern Pandit - Final Report

Shudhatma Jain 130050024

Aditya Kusupati 130050054

Owais Khan 130050050

Introduction:

IIT-B community organises numerous events covering every possible genre in the world. However participating alone becomes very boring and one doesn't enjoy or learn to the fullest. Moreover, many events require pairing up and team formation. So what's the issue?? The problem one faces is that, every individual has his/her own interests and hence finding a compatible team member with similar interests at times becomes difficult. Consequently, participants are not able to make the most of these events or they just don't participate. We experienced this issue in our daily life and hence came the idea of building an app to fix this up.

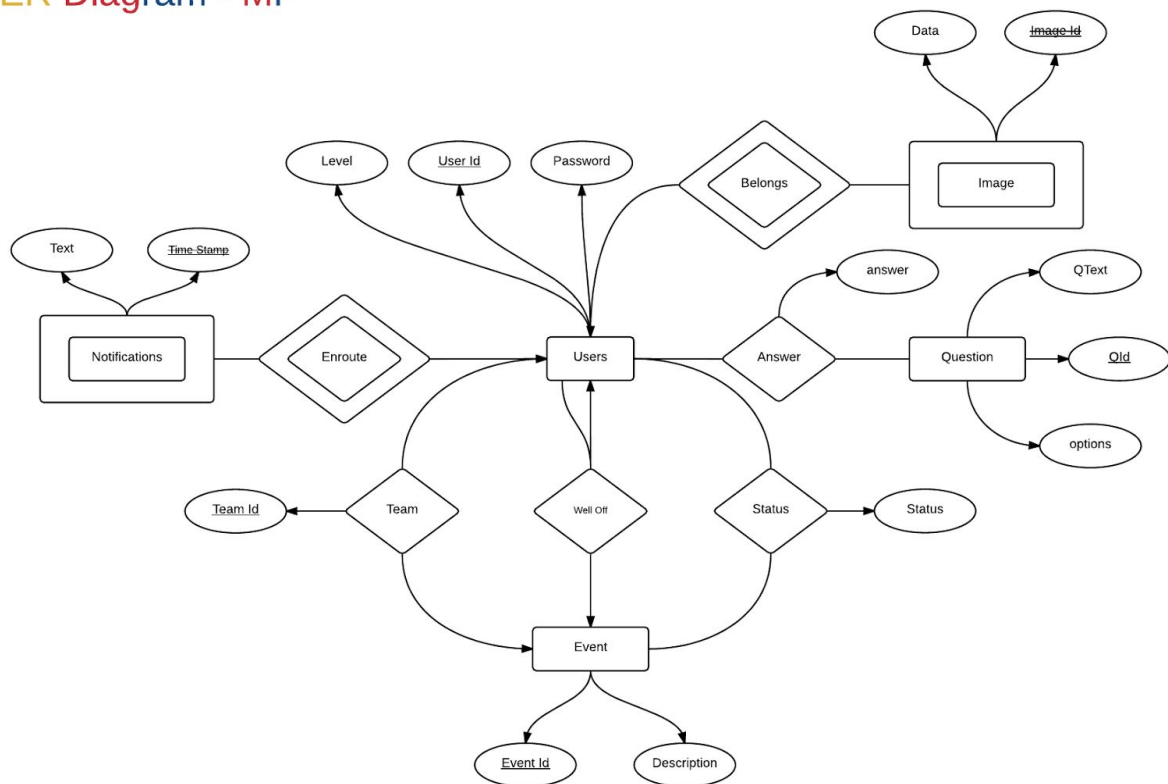
Approach:

Individuals interested in finding out potential pairings register on the app. A list of upcoming events is available to the users. Depending on their interests they can opt for events to participate in. Once they select 'Going' for an event, we provide suggestions to the users of potentially candidates for team formation. The app include feature of notifications.

Tools Used:

Java, JDBC, Javascripts(Ajax), jquery, GSON library, Apache tomcat v7.0., HTML, CSS, Bootstrap

ER Diagram - MP



*Struck off text are primary keys for weak entities

Filters:

AuthFilter

Since the application is user specific for many pages and does not allow any anonymous logins, we need to filter each request and make sure it comes from a user who is logged in. In order to achieve this when a user logs in we set an attribute corresponding to his/her session variable and then check whether that attribute is set in the later requests made by the user.

Modern Pandit x Aditya

localhost:8080/ModernPandit/auth

Apps Bookmarks fb CS G1 Messenger G0 FS Why Graph Data Other bookmarks

Login or Sign Up

New Account:

First Name

Last Name

Username

Email

Link to Fb profile

Password

☐ Male ☐ Female ☐ Other

mm/dd/yyyy

Create Account

Login:

Username

Password

Login

Servlets:

SignUpServlet

When a user is creating a new account this servlet checks if the username is already taken and if not it creates the account using the Member class and redirects to the login page. If the user name is taken or if the data entered is incorrect it responds with the corresponding error messages.

Connected JSP Files: login.jsp

Login or Sign Up

New Account:

☒ Male
 ☐ Female
 ☐ Other

Login:

Login

Create Account

AuthServlet

The main functionality of this servlet is to authenticate a user. If the user is authenticated the servlet redirects the user to the home page and ties his session key to a Member object corresponding to his account. Else he/she is redirected to login.jsp

Connected JSP Files : login.jsp

Login or Sign Up

New Account:

Account Successfully Created

☒ Male
 ☐ Female
 ☐ Other

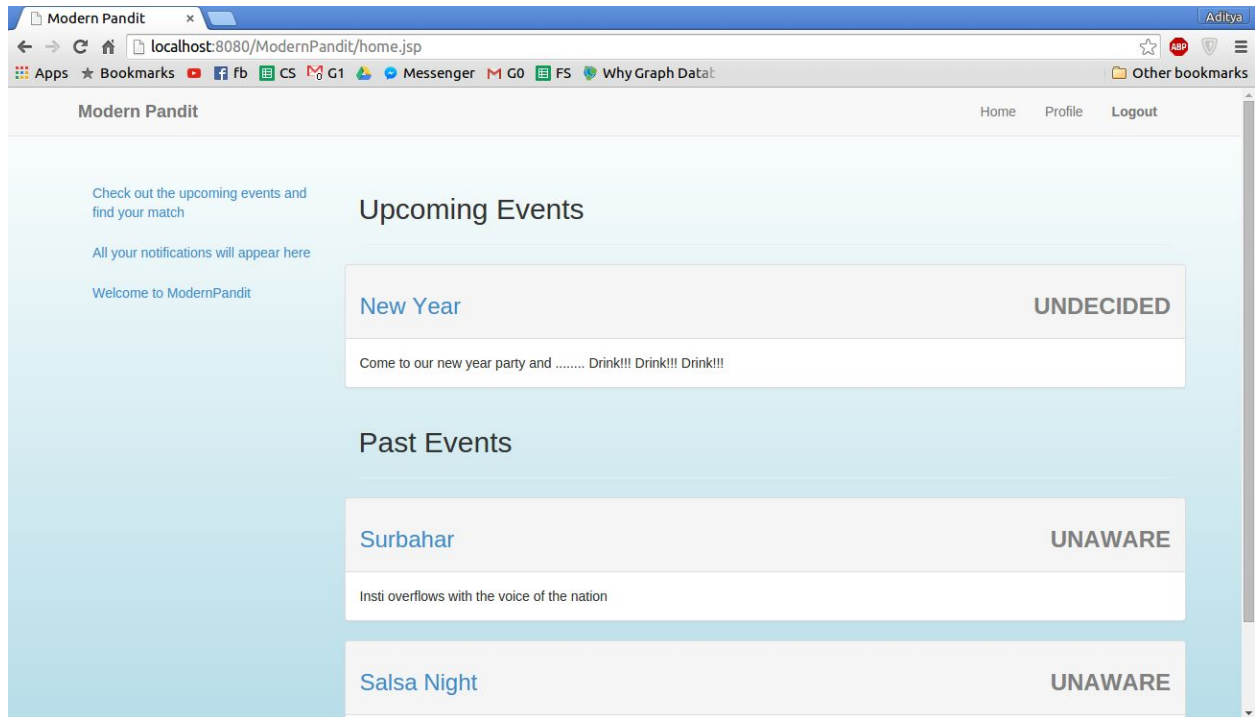
Login:

Login

Create Account

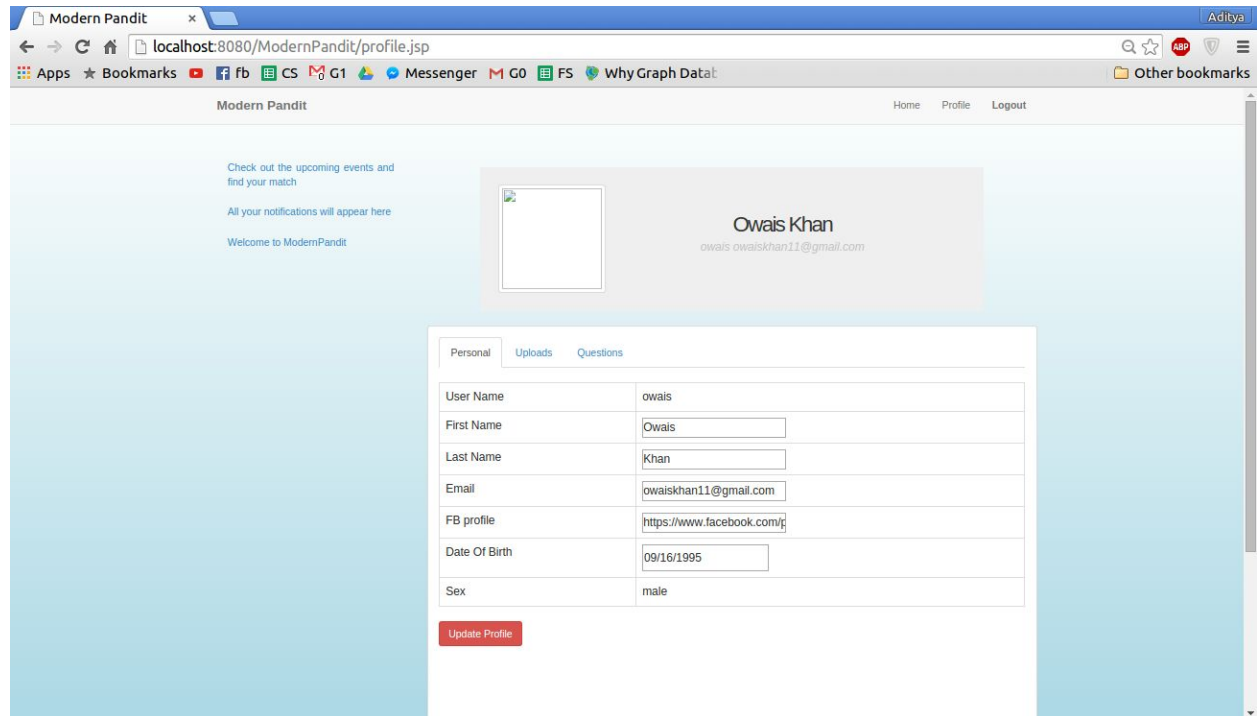
NotificationServlet

The users browsers continually polls the server for any new notifications. These requests are handled by this servlet. Since the polling would happen quite often, this servlet must be efficient. Connected JSP : profile.jsp, home.jsp, event.jsp



ProfileServlet

In case a registered user wants to update his/her profile, the requests for updating data done through the profile.jsp page is handled by this servlet.

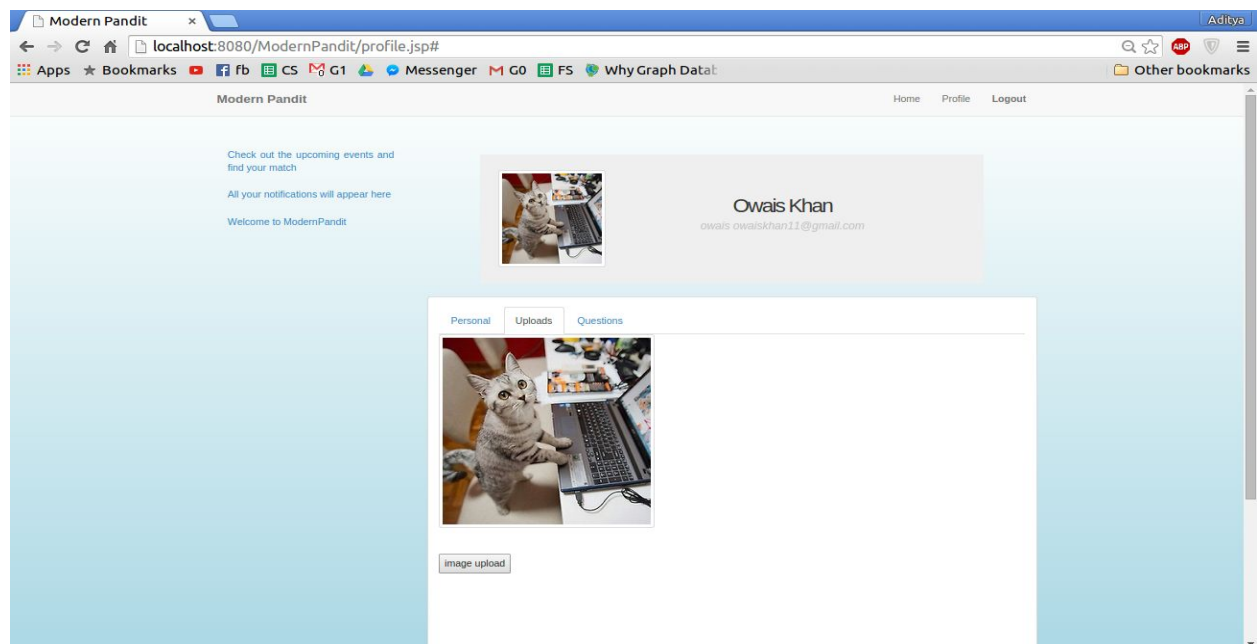


FileUploadServlet:

Servlet to handle the Photo uploads and storing them in database

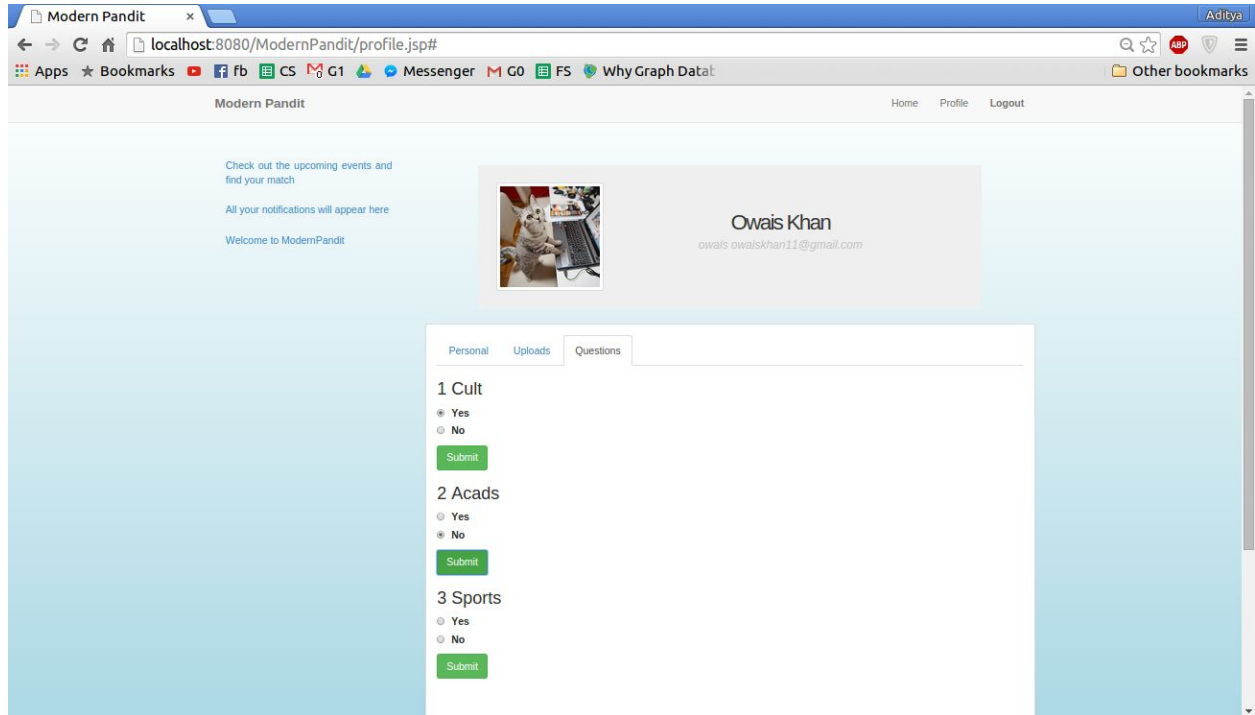
FileRetrieveServlet:

Servlet to retrieve the uploaded photos and save them as Profile picture based on the image id stored in user database



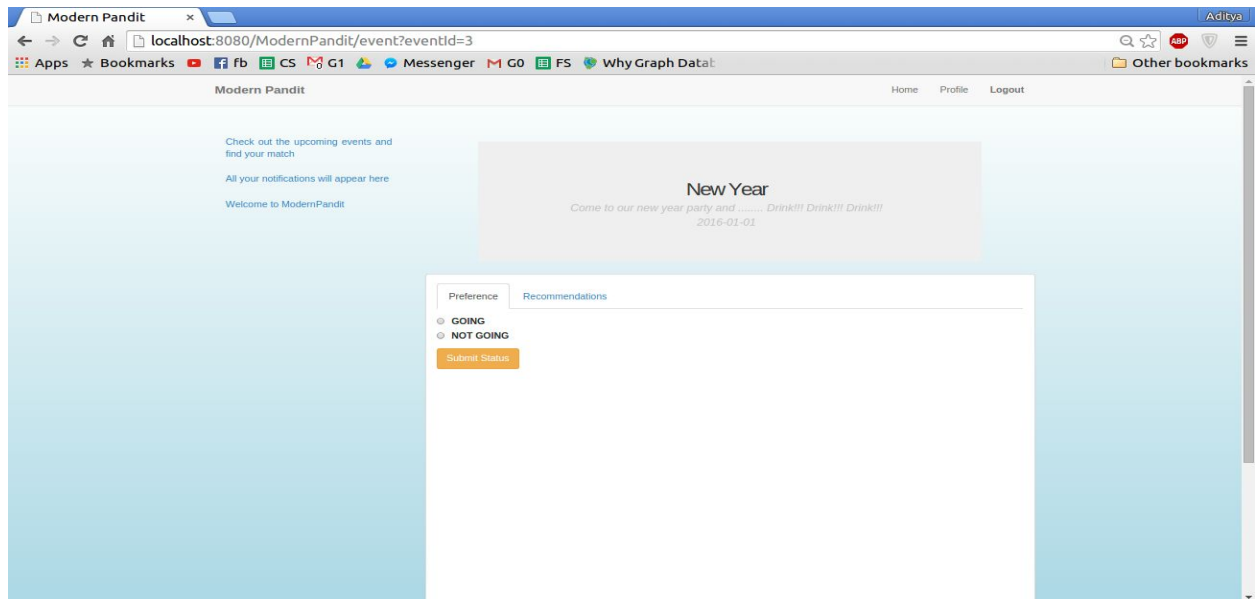
QuestionServlet

The servlet to render questions and updation of answers by a user. If a particular users has already answered a question then the question is retrieved along with the previous answers. Connected JSP Files : profile.jsp



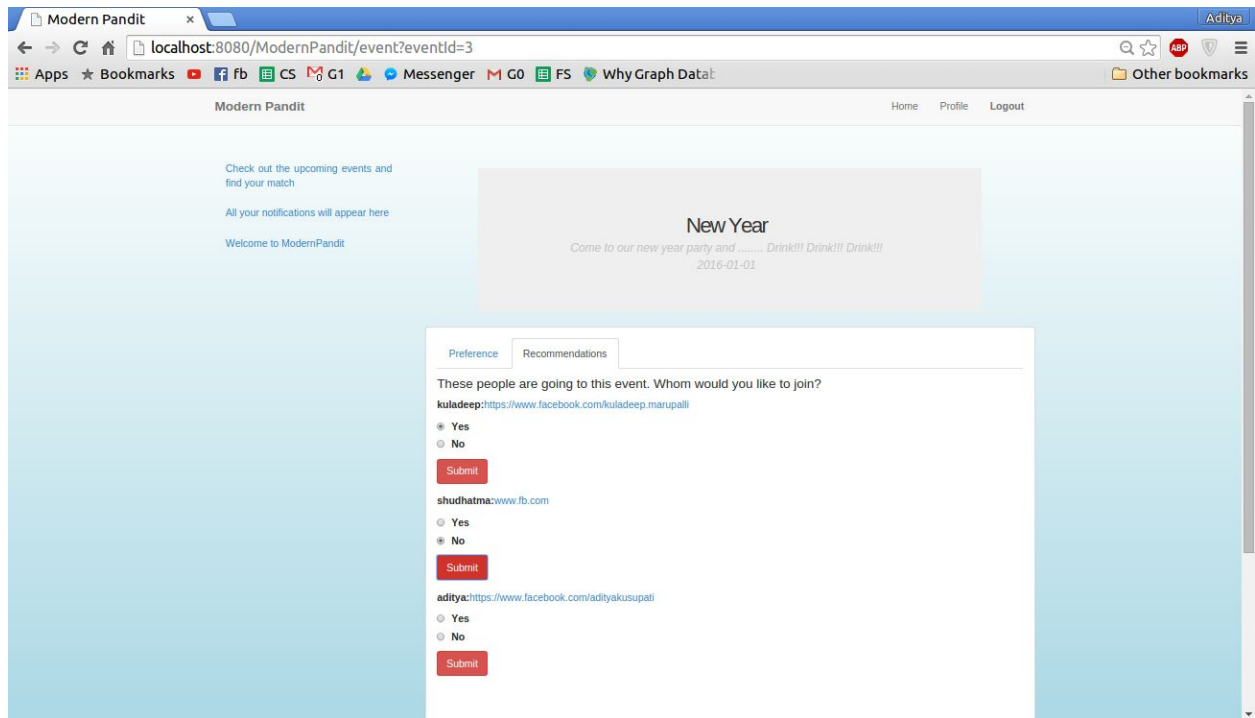
EventServlet

This servlet will allow the required redirection to the corresponding event page selected.



RecosServlet

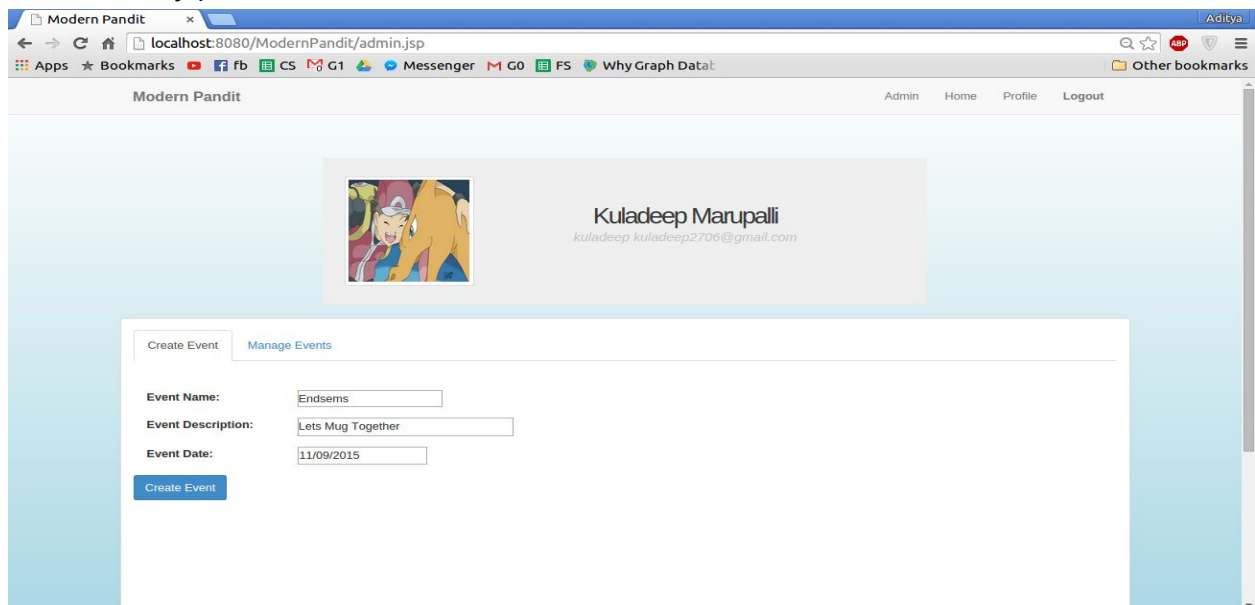
Handles AJAX requests for recommendations for a user. The servlet contacts the Recommendations class in order to achieve this.



CreateEventServlet :

The servlet to create a new event. Access is allowed only for admins.

JSPs: admin.jsp

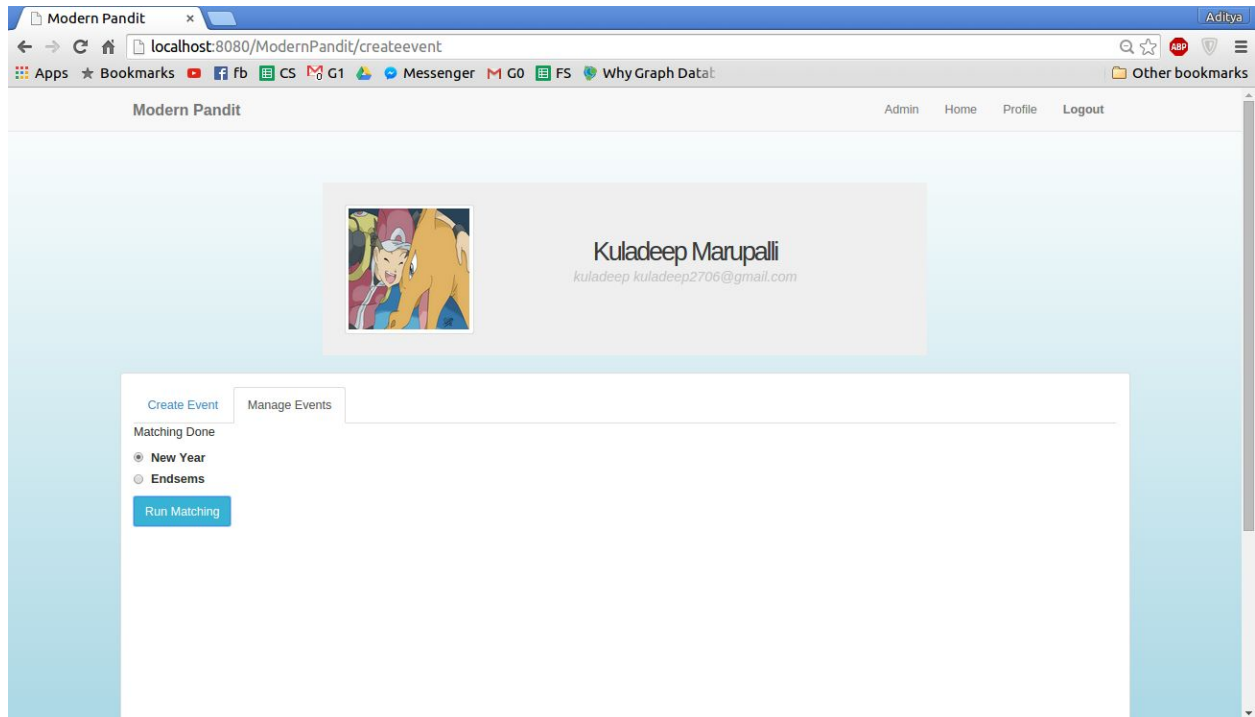


On creation of a new event a notification is sent to all the users.

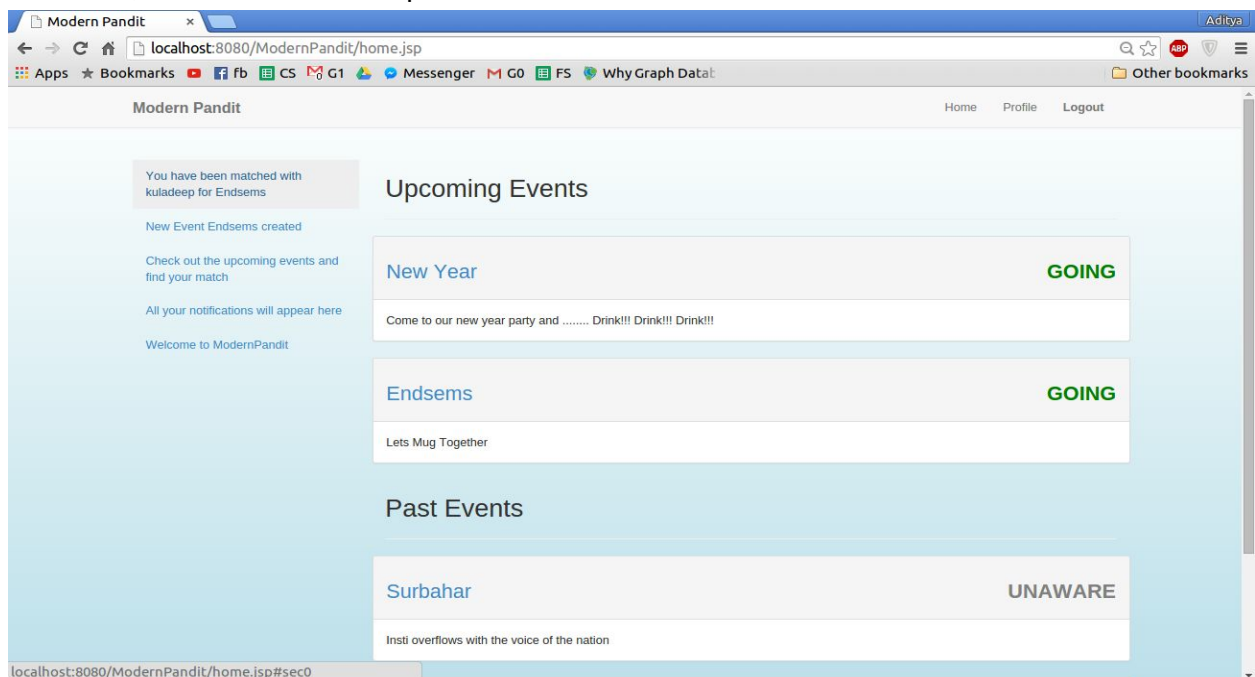
MatchingServlet

After an allotted time of registering for an event, it's up to the admin to make a request to this servlet to run the matching for an event.

Connected JSP : admin.jsp



On match, notifications come up to the users.



PassResetServlet

In case of a forgotten password a request for a new password is handled by the PassResetServlet. Later on the reset link and also the new password is validated through this servlet. It is **Defunct**.

Connected JSP Files: passreset.jsp reset.jsp

MessagesServlet

The servlet to handle messages and we used AJAX for polling for new messages. It is **Defunct** now

Classes:

Events

The creation and deletion of events is handled here. In case a new new event is created or if the event details are modified notifications are sent to users who are going for the event.

Matching

The bulk of the matching algorithm runs in this class. After receiving a request to start matching for a particular event we run the First Come First Serve matching algorithm and after a user is successfully matched.

Member

This class handles signup and login features. It also handles the attributes of a user(updation of answers, history maintenance etc.) . After a user is authenticated we create an object corresponding to that user's account and fill in the details from the database. These details are later used in the jsp pages in order to generate the jsp page specific to the user.

Notification

All the notification generation happens over here. This class provides an interface for creating and storing notifications for a particular user.

Later when the user is logged in theses notifications are polled for and retrieved.

Question

All the collection of data about a particular user is handled in this class. In the profile page we present the user with a list of questions which are retrieved from this class.

It also provides functions for creating new questions and deleting older ones.

Recommendations

In this class we have functions to find all the users which are attending a particular event and based on information previously collected we generate a list of recommendations. This will also make and update in the “welloff” table for the corresponding users.

Team

After the matching algorithm is run for an event in this class we store update the matches that are made form teams of the members, which is updated in the database.

FileUpload

Functions to upload image and get the image back from database are implemented properly in this class which will be used in FileUploadServlet and FileRetrieveServlet.

Status

Functions to update status table if a user is going to a certain event or not.

WellOff

Functions to update the welloff table of a user depending on his comfort matching with another user in the recommendation for a particular event.

Message

Functions to generate, send and get messages is implemented in this class.

PassReset

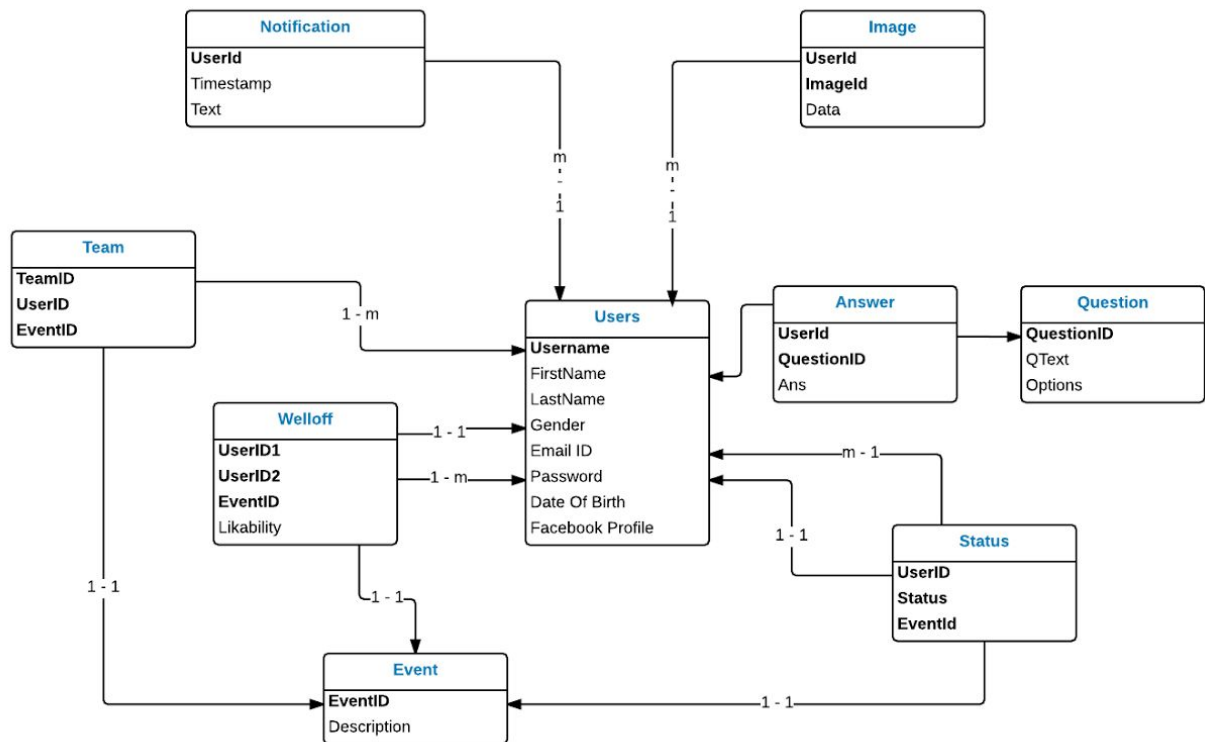
When a password request comes to this class. We use a hash function to generate a unique url which is later sent to the user's registered email.

When the user clicks on this url we delete the url from the database and provide him/her with the option of resetting the password for their account.

DB Designs :

Design Schema :

Database Schema - MP



- Bold Items in the talbes indicate primary keys
- Arrows denote the foreign key relationships

Access Controls : We have assigned different powers to Admin and general members. This functionality will be implemented using Java functions.

Features implemented but not Used :

- 1) **Password Reset** : In future we want to link the app to FB or gmail or LDAP, so there is no point in incorporating it even after implementation.
- 2) **Messages** : We have given the access to their fb profiles, so we feel that chat would be better on Fb rather than on an unfamiliar chat bot.

Improvements possible:

- 1) **Improve questions** : Improve questions so as to discretise and group the users based on the answers and interests
- 2) **Improve matching algorithm for recos and matching** : Using the answers of the questions to provide good recommendations for matching and a better algorithm using the ratings as a parameter
- 3) **LDAP / FB and G+ login using SSO** : Use the famous API for FB or G+ login or use LDAP login provided by asc
- 4) **Rating Feature** : Rate the recommendations so as to use it as a parameter for the algorithms used in matching
- 5) **Matching History** : We wish to put up the matching history of each person for memories and take feedback on the match after the event is over

Test Data

As of now we worked on self-generated data and took help from some of our batch mates.

Future Plans

We plan to check this on our batch (crowdsourcing). If everything goes fine, we intend to release this over the institute and fulfill our aim of comfortable and optimal team formation. And

Citations & References

- 1) <https://docs.oracle.com/javase/tutorial/>
- 2) <http://www.postgresql.org/>
- 3) www.w3schools.com
- 4) <http://stackoverflow.com/>