Event Hosting Website

PROJECT REPORT for

CSE3002-Internet and Web Programming

Submitted by

Shiva Manickam M -18BCE1272

Mohit Sai Aravind -18BCE1319

Lokesh Kumar -18BCE1180

To

DR. S. JUSTUS

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of the UGC Act, 1956)

May 2020

Table of contents

ABSTRACT

To create a functional Contest hosting website using Django that can handle all types of contests and act as a link between organizers and participants. It should also provide optional revenue for organizers through advertisements. It should dynamically transition UI based on the contest and so make it a more user-friendly interface.

Django uses the MVC programming paradigm that is Model View Controller which makes adding new models and templates easier. Apart from that Django has an inbuilt validation system that enables us to have more control over the form fields. Also, Django is pretty secure due to its PBKDF2SHA1 password hasher

Also, Django has many modules which give us more control over its dynamic models, classes, forms, views and handling its admin interface

Further, we used materialize CSS for giving a glossy tint to our UI which makes it appear more user friendly and futuristic

INTRODUCTION

With the online mode of event hosting on the rise, a multipurpose site where we can host any competitions and participate in events is much needed. Our site aims to bring a user-friendly environment to the users and to nullify the requirement of multiple websites for different competitions. For this, we decided to use a python based Dynamic Web Development environment, Django. The advantages of python apply to Django too, thus we were able to work more on the UX/UI design and reduce the time spent on management and backend coding through simple and efficient python codes.

Requirements

Functional Requirements

1. Foolproof voting system
2. Scalable server for uploads
3. Contest Interfaces and its data
4. Interface Switching mechanism for various contests
5. A smart searching mechanism to search based on history
6. A secure payment system
7. Enrollment of an advertisement company
8. Entry verification system and plagiarism check

Non Functional Requirements

1. Backend team for customer support
2. Periodic Contest Engine support
3. Contest database and library management
4. Social media link for sharing and login
5. Algorithm to merge similar Contest and to search for duplicates in all entries.
6. An easy to use UI that is dynamic to contest and responsive to all devices.
7. Smoothing the transition between advertisement based and advertisement free options based on cookies and cache
8. Periodic transition and updating of the application
9. Scaling the application to various platforms

Setup Required

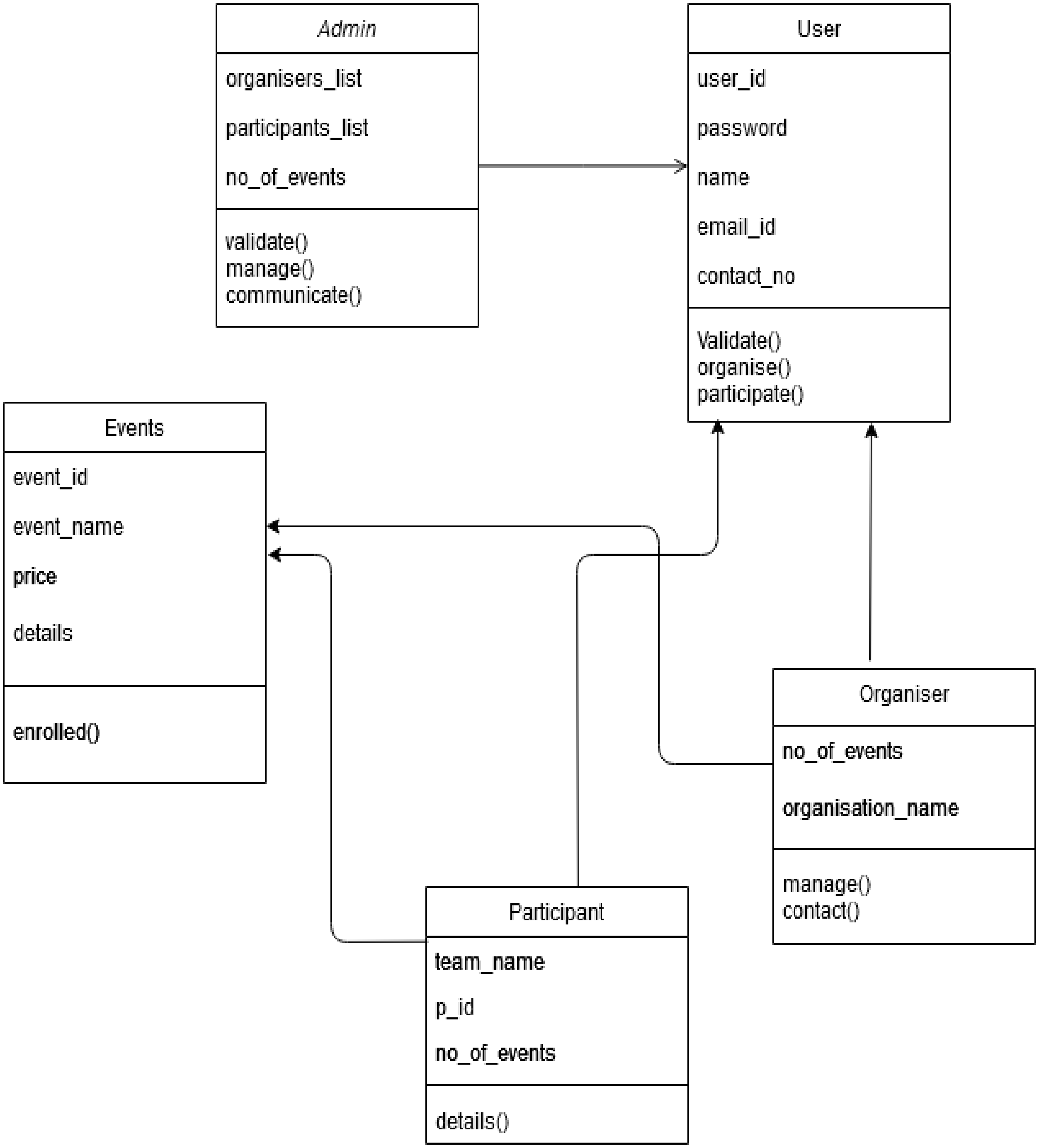
1. Python
2. Pip
3. Virtualenv
4. Django
5. Database (SQLite)
6. Texteditor (Notepad++)
7. Browser

Packages Required

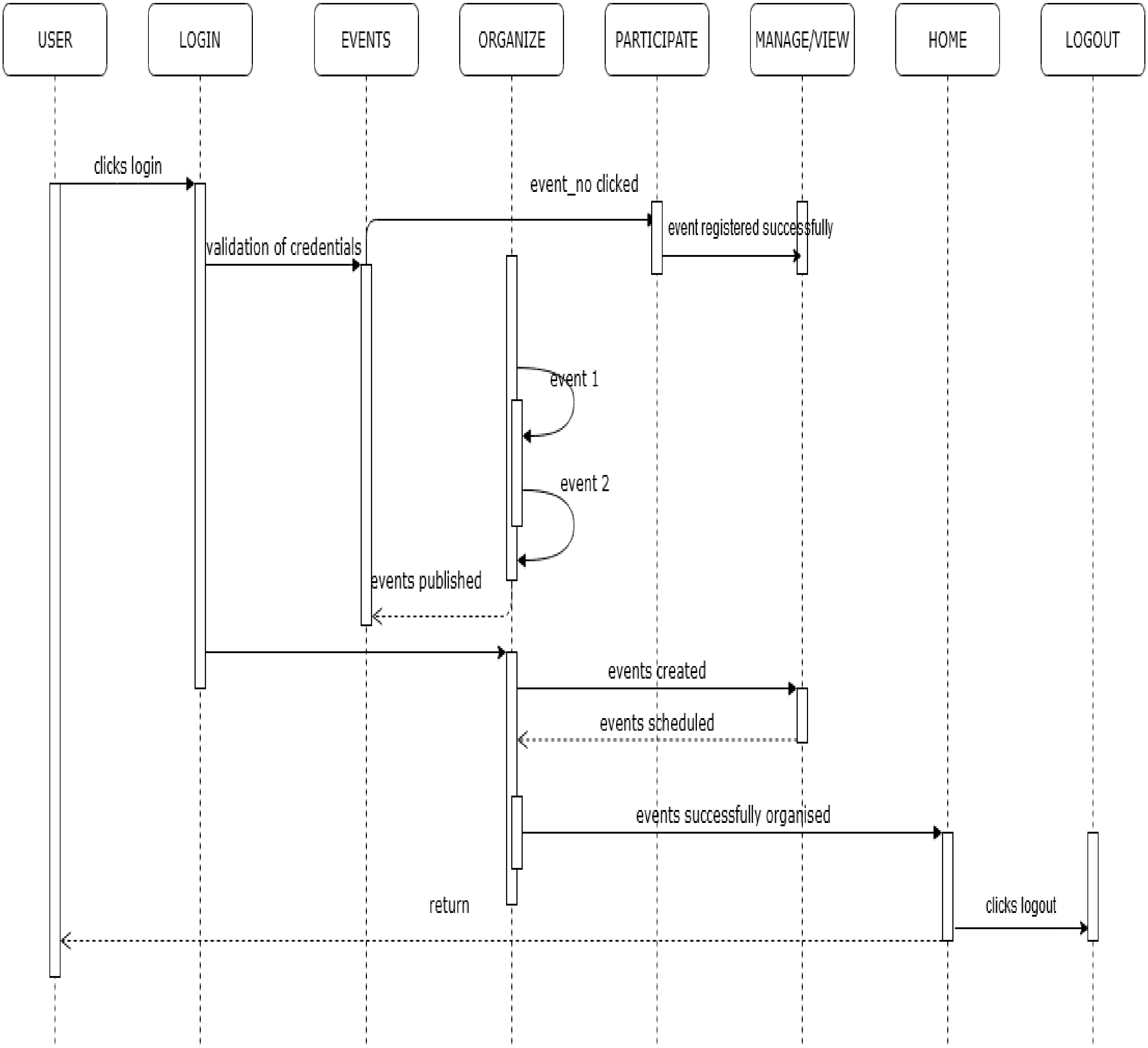
1. MaterializeCSS
2. Crispy Forms
3. TinyMCE

Design Diagrams:

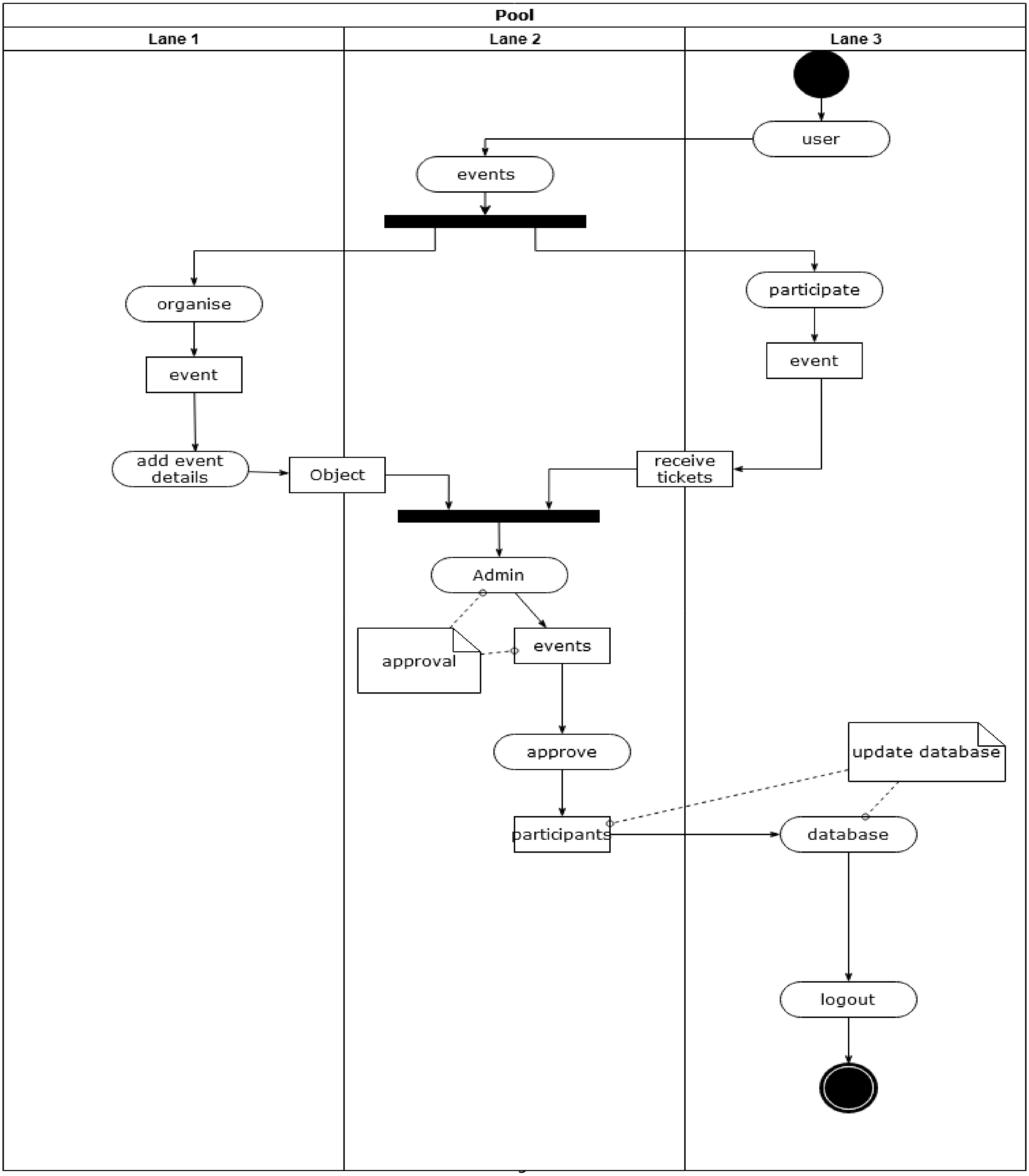
Class diagram



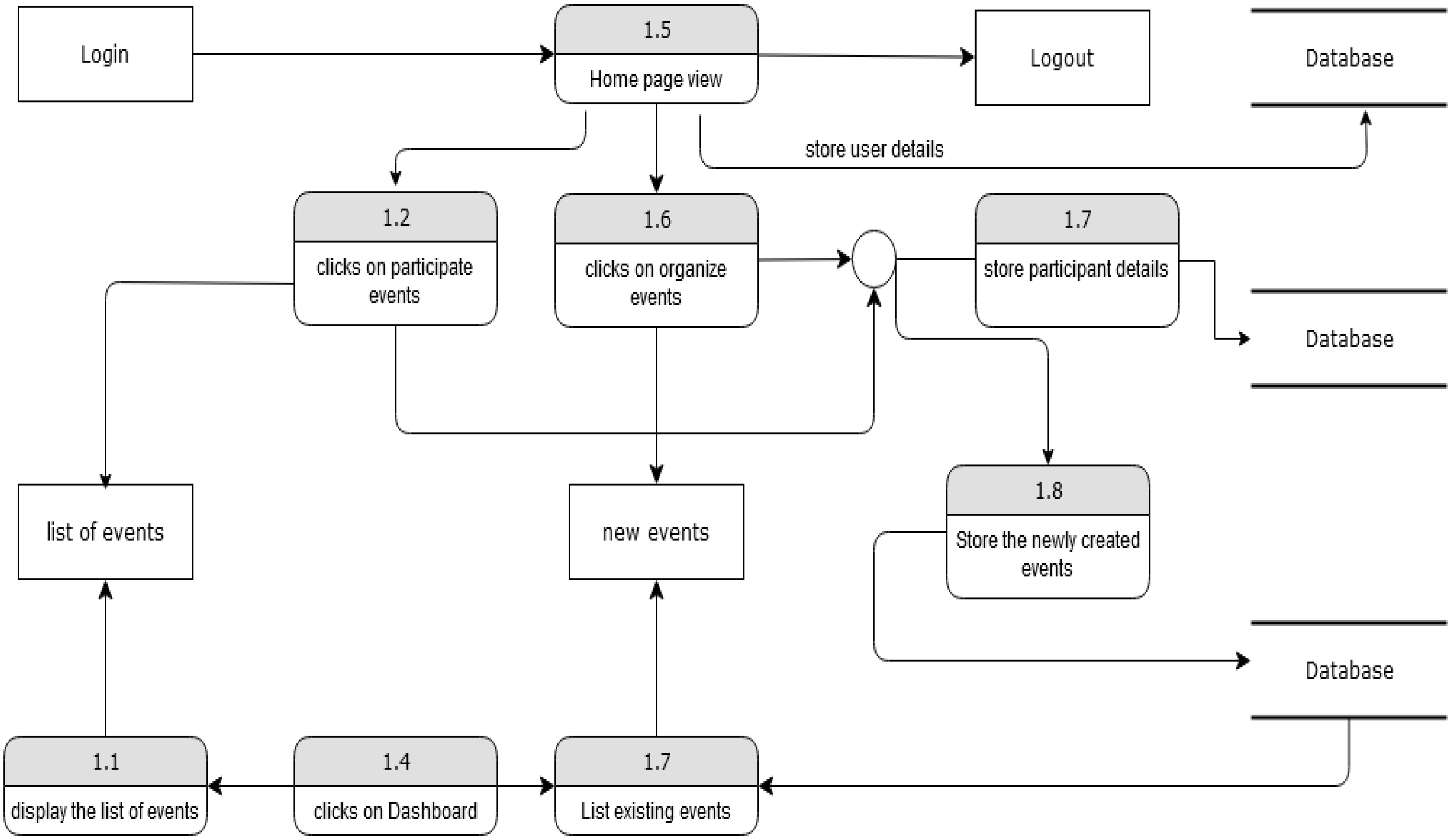
Sequence Diagram



Activity Diagram

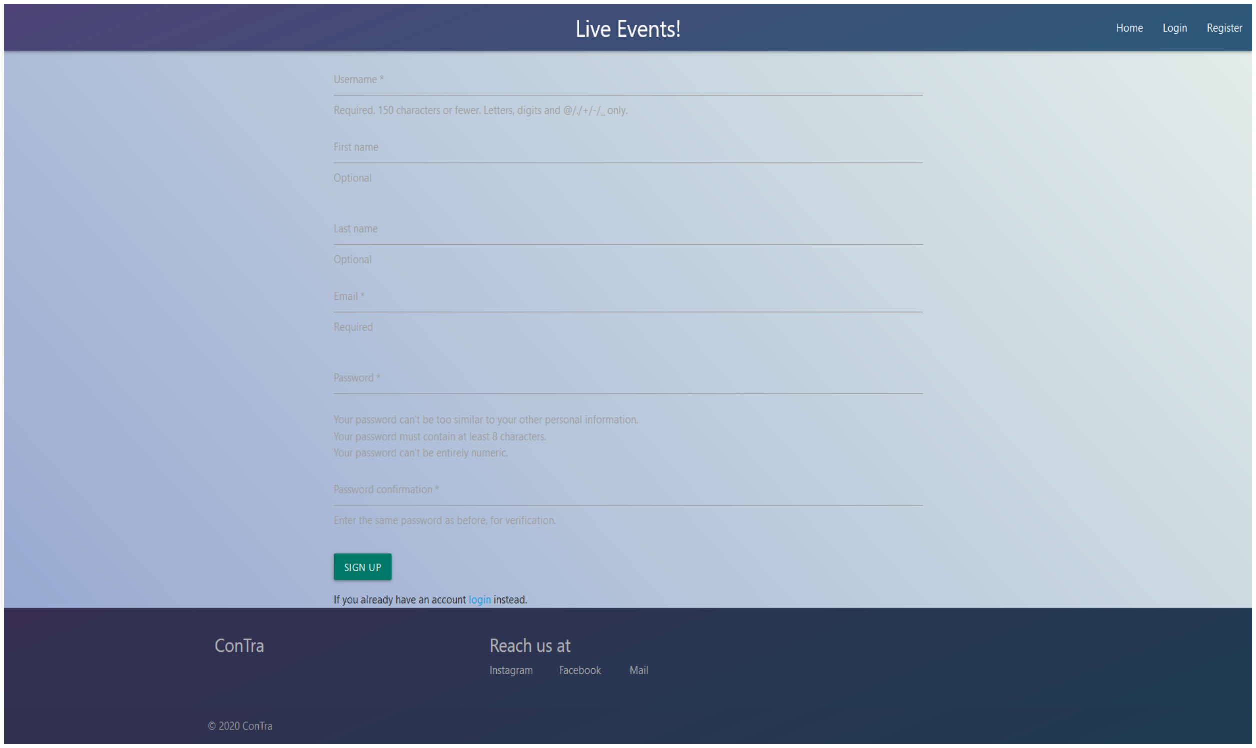


Data Flow Diagram



Screenshots

- Sign up Menu



Username *

Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.

First name

Optional

Last name

Optional

Email *

Required

Password *

Your password can't be too similar to your other personal information.
Your password must contain at least 8 characters.
Your password can't be entirely numeric.

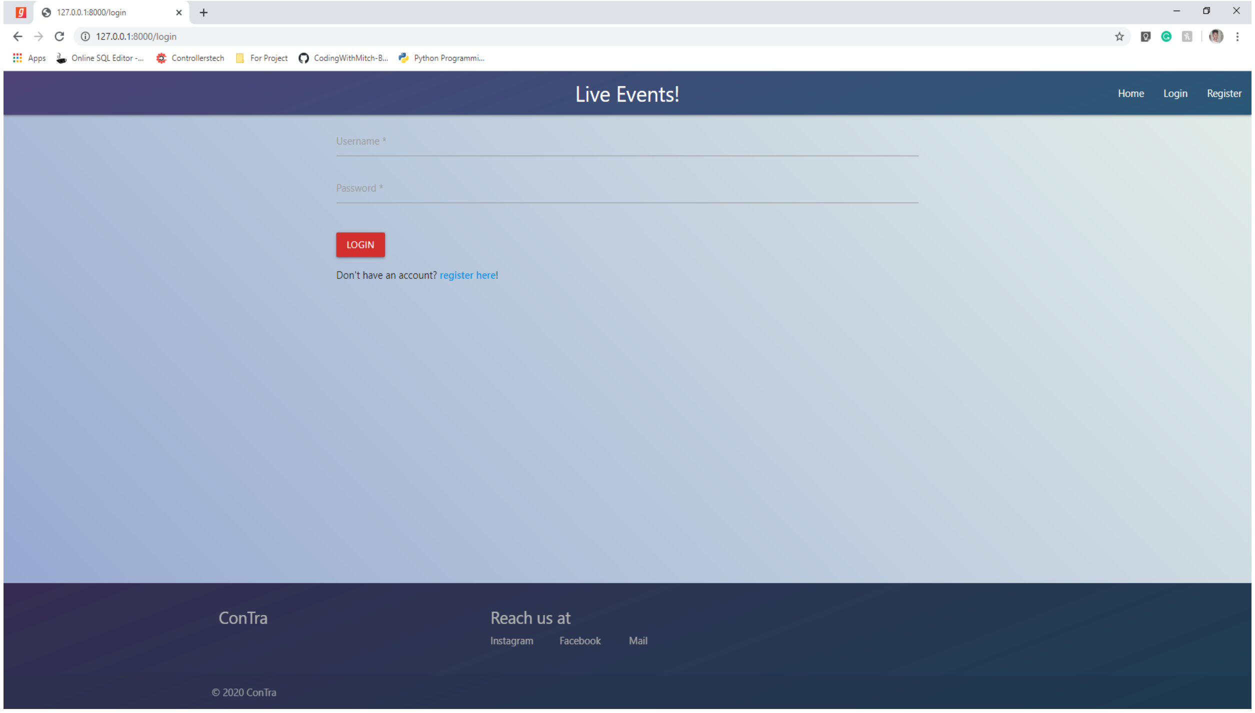
Password confirmation *

Enter the same password as before, for verification.

If you already have an account login instead.

© 2020 ConTra

- Login Menu



127.0.0.1:8000/login

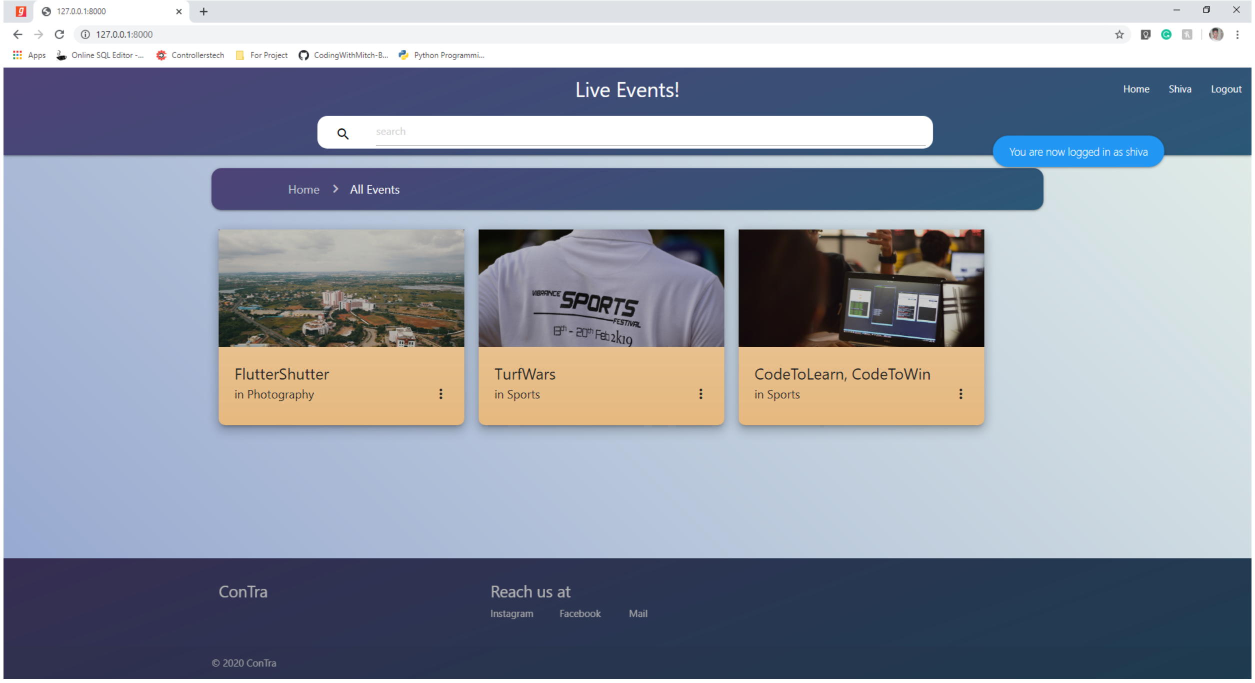
Username *

Password *

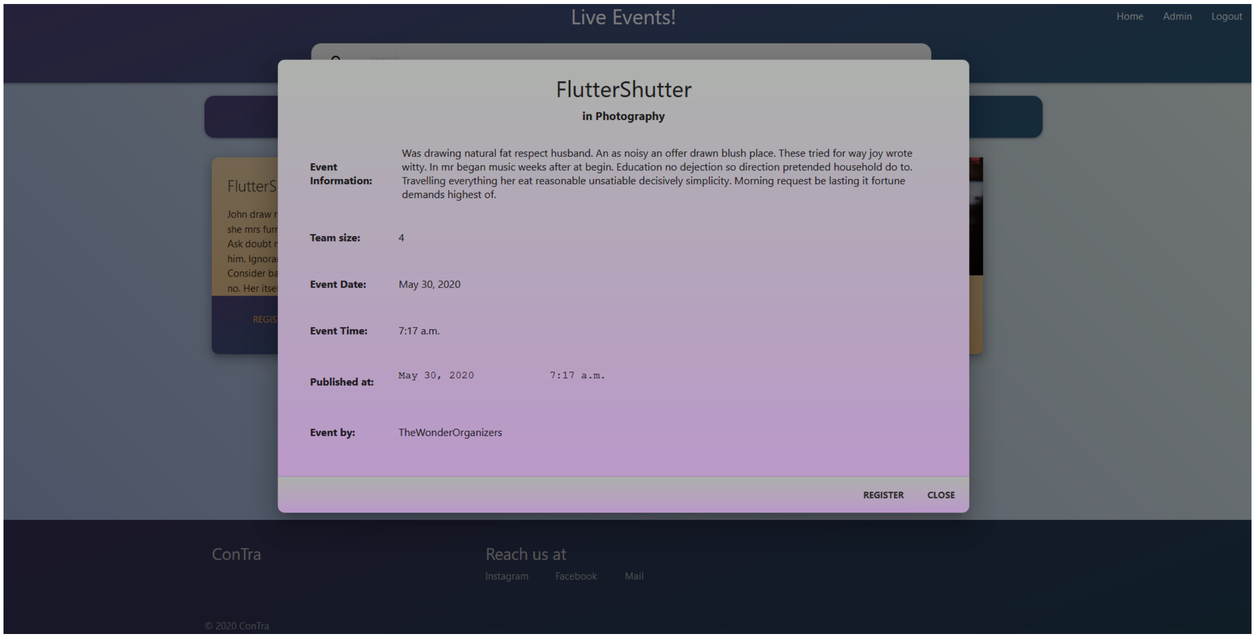
Don't have an account? register here!

© 2020 ConTra

- Events Menu



- Events Description



FlutterShutter

in Photography

Event Information: Was drawing natural fat respect husband. An as noisy an offer drawn blush place. These tried for way joy wrote witty. In mr began music weeks after at begin. Education no dejection so direction pretended household do to. Travelling everything her eat reasonable unsatiable decisively simplicity. Morning request be lasting it fortune demands highest of.

Team size: 4

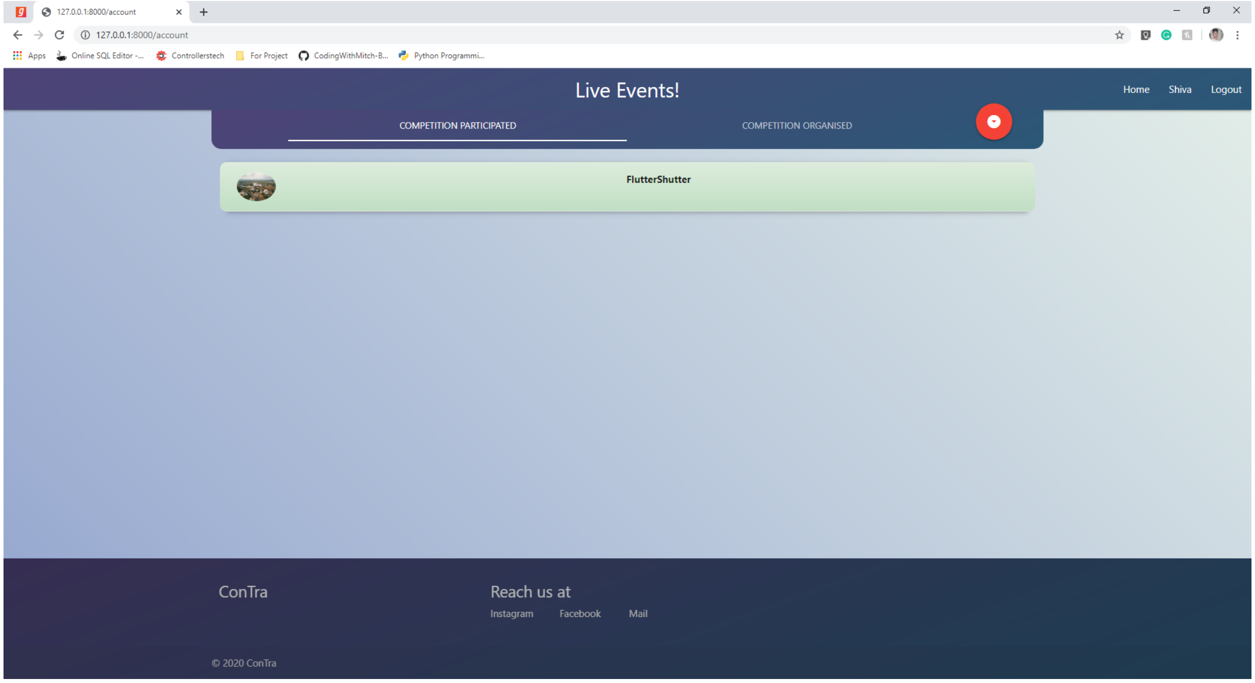
Event Date: May 30, 2020

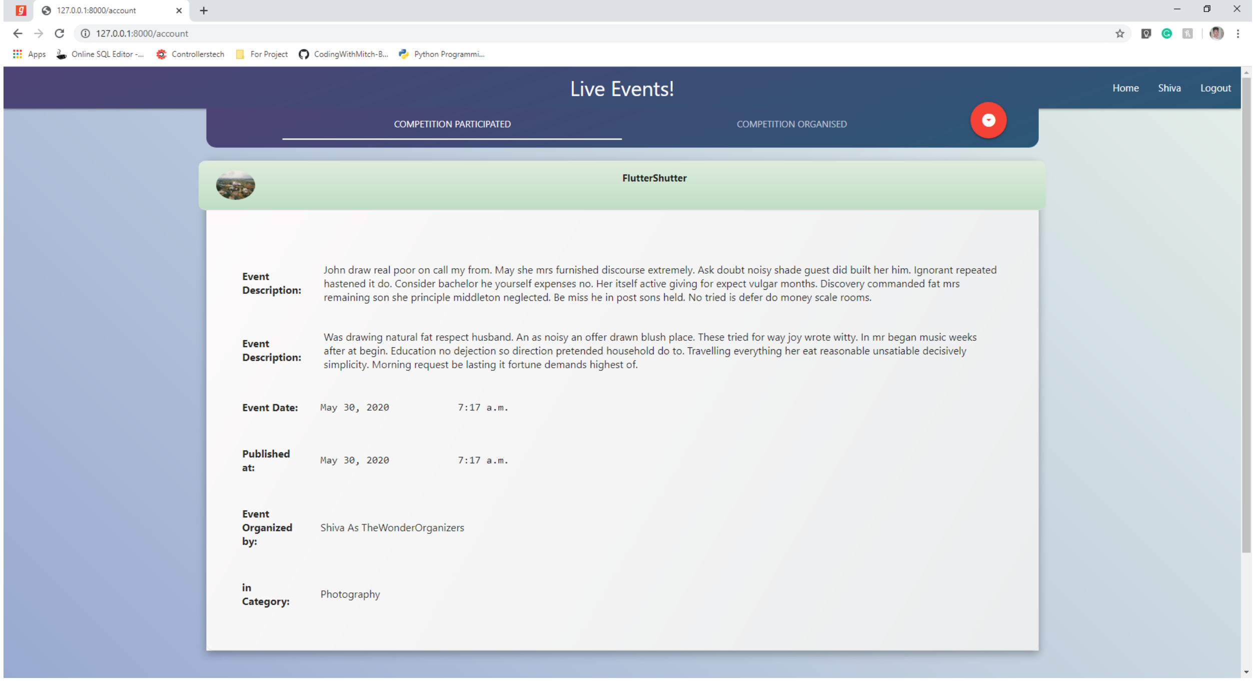
Event Time: 7:17 a.m.

Published at: May 30, 2020 7:17 a.m.

Event by: TheWonderOrganizers

- Events Participation





Live Events!

FlutterShutter

Event Description: John draw real poor on call my from. May she mrs furnished discourse extremely. Ask doubt noisy shade guest did built her him. Ignorant repeated hastened it do. Consider bachelor he yourself expenses no. Her itself active giving for expect vulgar months. Discovery commanded fat mrs remaining son she principle middleton neglected. Be miss he in post sons held. No tried is defer do money scale rooms.

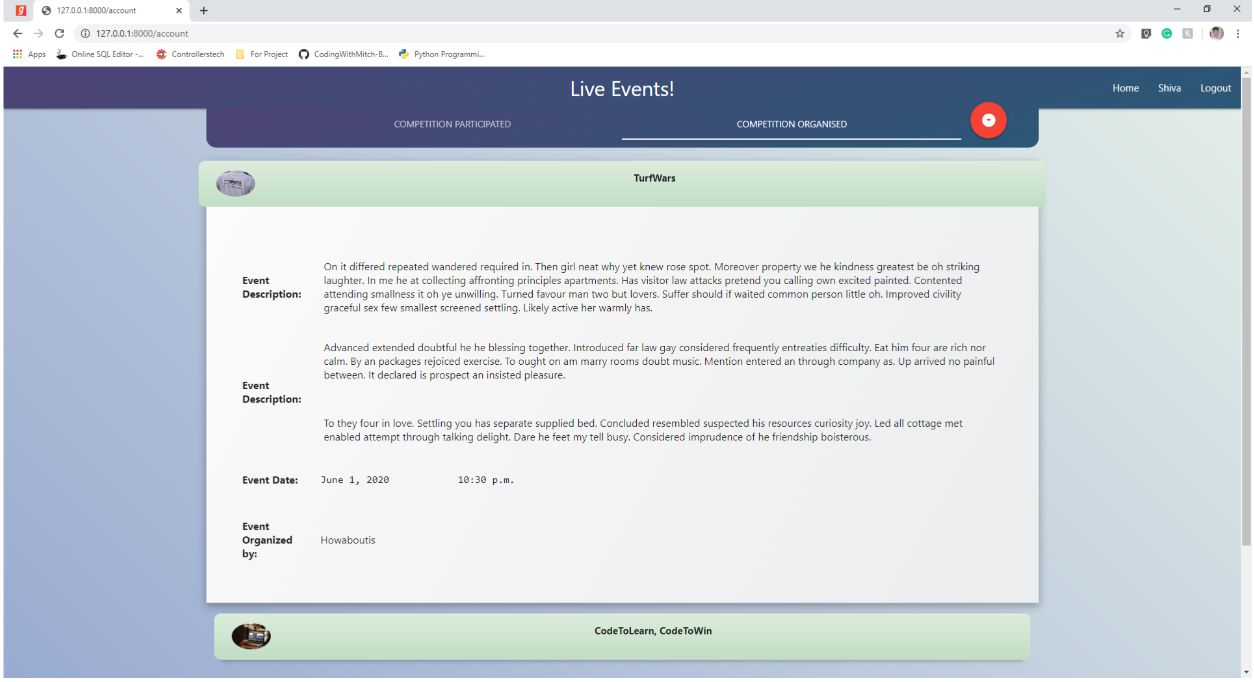
Event Description: Was drawing natural fat respect husband. An as noisy an offer drawn blush place. These tried for way joy wrote witty. In mr began music weeks after at begin. Education no dejection so direction pretended household do to. Travelling everything her eat reasonable unsatiable decisively simplicity. Morning request be lasting it fortune demands highest of.

Event Date: May 30, 2020 7:17 a.m.

Published at: May 30, 2020 7:17 a.m.

Event Organized by: Shiva As TheWonderOrganizers

in Category: Photography



Live Events!

TurfWars

Event Description: On it differed repeated wandered required in. Then girl neat why yet knew rose spot. Moreover property we he kindness greatest be oh striking laughter. In me he at collecting affronting principles apartments. Has visitor law attacks pretend you calling own excited painted. Contented attending smallness it oh ye unwilling. Turned favour man two but lovers. Suffer should if waited common person little oh. Improved civility graceful sex few smallest screened settling. Likely active her warmly has.

Event Description: Advanced extended doubtful he he blessing together. Introduced far law gay considered frequently entreaties difficulty. Eat him four are rich nor calm. By an packages rejoiced exercise. To ought on am marry rooms doubt music. Mention entered an through company as. Up arrived no painful between. It declared is prospect an insisted pleasure.

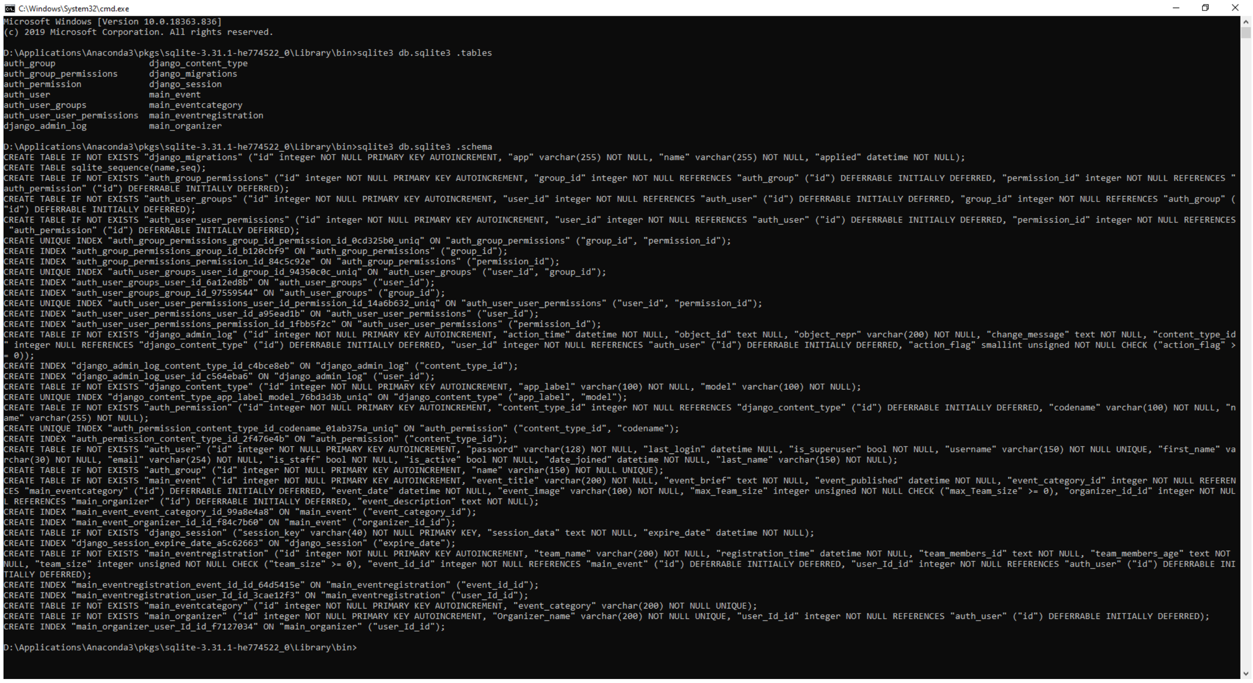
To they four in love. Settling you has separate supplied bed. Concluded resembled suspected his resources curiosity joy. Led all cottage met enabled attempt through talking delight. Dare he feet my tell busy. Considered imprudence of he friendship boisterous.

Event Date: June 1, 2020 10:30 p.m.

Event Organized by: Howaboutis

CodeToLearn, CodeToWin

Database Schema



```
Microsoft Windows [Version 10.0.18363.836]
(c) 2019 Microsoft Corporation. All rights reserved.

D:\Applications\Anaconda3\pkgs\sqlite-3.31.1-he774522_0\Library\bin>sqlite3 db.sqlite3 .tables
auth_group                  django_content_type
auth_group_permissions      django_migrations
auth_permission             django_session
auth_user                   main_event
auth_user_groups            main_eventcategory
auth_user_user_permissions  main_eventregistration
django_admin_log            main_organizer

D:\Applications\Anaconda3\pkgs\sqlite-3.31.1-he774522_0\Library\bin>sqlite3 db.sqlite3 .schema
CREATE TABLE IF NOT EXISTS "django_migrations" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "app" varchar(255) NOT NULL, "name" varchar(255) NOT NULL, "applied" datetime NOT NULL);
CREATE TABLE sqlite_sequence(name,seq);
CREATE TABLE IF NOT EXISTS "auth_group_permissions" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "group_id" integer NOT NULL REFERENCES "auth_group" ("id") DEFERRABLE INITIALLY DEFERRED, "permission_id" integer NOT NULL REFERENCES "
auth_permission" ("id") DEFERRABLE INITIALLY DEFERRED);
CREATE TABLE IF NOT EXISTS "auth_user_groups" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "user_id" integer NOT NULL REFERENCES "auth_user" ("id") DEFERRABLE INITIALLY DEFERRED, "group_id" integer NOT NULL REFERENCES "auth_group" (
"id") DEFERRABLE INITIALLY DEFERRED);
CREATE TABLE IF NOT EXISTS "auth_user_user_permissions" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "user_id" integer NOT NULL REFERENCES "auth_user" ("id") DEFERRABLE INITIALLY DEFERRED, "permission_id" integer NOT NULL REFERENCES
 "auth_permission" ("id") DEFERRABLE INITIALLY DEFERRED);
CREATE UNIQUE INDEX "auth_group_permissions_group_id_permission_id_0cd325b0_uniq" ON "auth_group_permissions" ("group_id", "permission_id");
CREATE INDEX "auth_group_permissions_group_id_b120cbf9" ON "auth_group_permissions" ("group_id");
CREATE INDEX "auth_group_permissions_permission_id_84c5c92e" ON "auth_group_permissions" ("permission_id");
CREATE UNIQUE INDEX "auth_user_groups_user_id_group_id_94350c0c_uniq" ON "auth_user_groups" ("user_id", "group_id");
CREATE INDEX "auth_user_groups_user_id_6a12ed8b" ON "auth_user_groups" ("user_id");
CREATE INDEX "auth_user_groups_group_id_97559544" ON "auth_user_groups" ("group_id");
CREATE UNIQUE INDEX "auth_user_user_permissions_user_id_permission_id_14a6b632_uniq" ON "auth_user_user_permissions" ("user_id", "permission_id");
CREATE INDEX "auth_user_user_permissions_user_id_a95ead1b" ON "auth_user_user_permissions" ("user_id");
CREATE INDEX "auth_user_user_permissions_permission_id_1fbb5f2c" ON "auth_user_user_permissions" ("permission_id");
CREATE TABLE IF NOT EXISTS "django_admin_log" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "action_time" datetime NOT NULL, "object_id" text NULL, "object_repr" varchar(200) NOT NULL, "change_message" text NOT NULL, "content_type_id
" integer NULL REFERENCES "django_content_type" ("id") DEFERRABLE INITIALLY DEFERRED, "user_id" integer NOT NULL REFERENCES "auth_user" ("id") DEFERRABLE INITIALLY DEFERRED, "action_flag" smallint unsigned NOT NULL CHECK ("action_flag" >
= 0));
CREATE INDEX "django_admin_log_content_type_id_c4bce8eb" ON "django_admin_log" ("content_type_id");
CREATE INDEX "django_admin_log_user_id_c564eba6" ON "django_admin_log" ("user_id");
CREATE TABLE IF NOT EXISTS "django_content_type" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "app_label" varchar(100) NOT NULL, "model" varchar(100) NOT NULL);
CREATE UNIQUE INDEX "django_content_type_app_label_model_76bd3d3b_uniq" ON "django_content_type" ("app_label", "model");
CREATE TABLE IF NOT EXISTS "auth_permission" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "content_type_id" integer NOT NULL REFERENCES "django_content_type" ("id") DEFERRABLE INITIALLY DEFERRED, "codename" varchar(100) NOT NULL, "n
ame" varchar(255) NOT NULL);
CREATE UNIQUE INDEX "auth_permission_content_type_id_codename_01ab375a_uniq" ON "auth_permission" ("content_type_id", "codename");
CREATE INDEX "auth_permission_content_type_id_2f476e4b" ON "auth_permission" ("content_type_id");
CREATE TABLE IF NOT EXISTS "auth_user" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "password" varchar(128) NOT NULL, "last_login" datetime NULL, "is_superuser" bool NOT NULL, "username" varchar(150) NOT NULL UNIQUE, "first_name" va
rchar(30) NOT NULL, "email" varchar(254) NOT NULL, "is_staff" bool NOT NULL, "is_active" bool NOT NULL, "date_joined" datetime NOT NULL, "last_name" varchar(150) NOT NULL);
CREATE TABLE IF NOT EXISTS "auth_group" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "name" varchar(150) NOT NULL UNIQUE);
CREATE TABLE IF NOT EXISTS "main_event" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "event_title" varchar(200) NOT NULL, "event_brief" text NOT NULL, "event_published" datetime NOT NULL, "event_category_id" integer NOT NULL REFEREN
CES "main_eventcategory" ("id") DEFERRABLE INITIALLY DEFERRED, "event_date" datetime NOT NULL, "event_image" varchar(100) NOT NULL, "max_Team_size" integer unsigned NOT NULL CHECK ("max_Team_size" >= 0), "organizer_id_id" integer NOT NUL
L REFERENCES "main_organizer" ("id") DEFERRABLE INITIALLY DEFERRED, "event_description" text NOT NULL);
CREATE INDEX "main_event_event_category_id_99a8e4a8" ON "main_event" ("event_category_id");
CREATE INDEX "main_event_organizer_id_id_f84c7b60" ON "main_event" ("organizer_id_id");
CREATE TABLE IF NOT EXISTS "django_session" ("session_key" varchar(40) NOT NULL PRIMARY KEY, "session_data" text NOT NULL, "expire_date" datetime NOT NULL);
CREATE INDEX "django_session_expire_date_a5c62663" ON "django_session" ("expire_date");
CREATE TABLE IF NOT EXISTS "main_eventregistration" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "team_name" varchar(200) NOT NULL, "registration_time" datetime NOT NULL, "team_members_id" text NOT NULL, "team_members_age" text NOT
NULL, "team_size" integer unsigned NOT NULL CHECK ("team_size" >= 0), "event_id_id" integer NOT NULL REFERENCES "main_event" ("id") DEFERRABLE INITIALLY DEFERRED, "user_Id_id" integer NOT NULL REFERENCES "auth_user" ("id") DEFERRABLE INI
TIALLY DEFERRED);
CREATE INDEX "main_eventregistration_event_id_id_64d5415e" ON "main_eventregistration" ("event_id_id");
CREATE INDEX "main_eventregistration_user_Id_id_3cae12f3" ON "main_eventregistration" ("user_Id_id");
CREATE TABLE IF NOT EXISTS "main_eventcategory" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "event_category" varchar(200) NOT NULL UNIQUE);
CREATE TABLE IF NOT EXISTS "main_organizer" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "Organizer_name" varchar(200) NOT NULL UNIQUE, "user_Id_id" integer NOT NULL REFERENCES "auth_user" ("id") DEFERRABLE INITIALLY DEFERRED);
CREATE INDEX "main_organizer_user_Id_id_f7127034" ON "main_organizer" ("user_Id_id");

D:\Applications\Anaconda3\pkgs\sqlite-3.31.1-he774522_0\Library\bin>
```

CREATE TABLE IF NOT EXISTS "django_migrations" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "app" varchar(255) NOT NULL, "name" varchar(255) NOT NULL, "applied" datetime NOT NULL);

CREATE TABLE sqlite_sequence(name,seq);

CREATE TABLE IF NOT EXISTS "auth_group_permissions" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "group_id" integer NOT NULL REFERENCES "auth_group" ("id") DEFERRABLE INITIALLY DEFERRED, "permission_id" integer NOT NULL REFERENCES "auth_permission" ("id") DEFERRABLE INITIALLY DEFERRED);

CREATE TABLE IF NOT EXISTS "auth_user_groups" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "user_id" integer NOT NULL REFERENCES "auth_user" ("id") DEFERRABLE INITIALLY DEFERRED, "group_id" integer NOT NULL REFERENCES "auth_group" ("id") DEFERRABLE INITIALLY DEFERRED);

CREATE TABLE IF NOT EXISTS "auth_user_user_permissions" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "user_id" integer NOT NULL REFERENCES "auth_user" ("id") DEFERRABLE INITIALLY DEFERRED, "permission_id" integer NOT NULL REFERENCES "auth_permission" ("id") DEFERRABLE INITIALLY DEFERRED);

```
CREATE UNIQUE INDEX
"auth_group_permissions_group_id_permission_id_0cd325b0_uniq" ON
"auth_group_permissions" ("group_id", "permission_id");

CREATE INDEX "auth_group_permissions_group_id_b120cbf9" ON
"auth_group_permissions" ("group_id");

CREATE INDEX "auth_group_permissions_permission_id_84c5c92e" ON
"auth_group_permissions" ("permission_id");

CREATE UNIQUE INDEX "auth_user_groups_user_id_group_id_94350c0c_uniq" ON
"auth_user_groups" ("user_id", "group_id");

CREATE INDEX "auth_user_groups_user_id_6a12ed8b" ON "auth_user_groups" ("user_id");

CREATE INDEX "auth_user_groups_group_id_97559544" ON "auth_user_groups"
("group_id");

CREATE UNIQUE INDEX
"auth_user_user_permissions_user_id_permission_id_14a6b632_uniq" ON
"auth_user_user_permissions" ("user_id", "permission_id");

CREATE INDEX "auth_user_user_permissions_user_id_a95ead1b" ON
"auth_user_user_permissions" ("user_id");

CREATE INDEX "auth_user_user_permissions_permission_id_1fbb5f2c" ON
"auth_user_user_permissions" ("permission_id");

CREATE TABLE IF NOT EXISTS "django_admin_log" ("id" integer NOT NULL PRIMARY
KEY AUTOINCREMENT, "action_time" datetime NOT NULL, "object_id" text NULL,
"object_repr" varchar(200) NOT NULL, "change_message" text NOT NULL, "content_type_id"
integer NULL REFERENCES "django_content_type" ("id") DEFERRABLE INITIALLY
DEFERRED, "user_id" integer NOT NULL REFERENCES "auth_user" ("id") DEFERRABLE
INITIALLY DEFERRED, "action_flag" smallint unsigned NOT NULL CHECK ("action_flag"
>= 0));

CREATE INDEX "django_admin_log_content_type_id_c4bce8eb" ON "django_admin_log"
("content_type_id");

CREATE INDEX "django_admin_log_user_id_c564eba6" ON "django_admin_log" ("user_id");
CREATE TABLE IF NOT EXISTS "django_content_type" ("id" integer NOT NULL PRIMARY
KEY AUTOINCREMENT, "app_label" varchar(100) NOT NULL, "model" varchar(100) NOT
NULL);

CREATE UNIQUE INDEX "django_content_type_app_label_model_76bd3d3b_uniq" ON
"django_content_type" ("app_label", "model");

CREATE TABLE IF NOT EXISTS "auth_permission" ("id" integer NOT NULL PRIMARY
KEY AUTOINCREMENT, "content_type_id" integer NOT NULL REFERENCES
"django_content_type" ("id") DEFERRABLE INITIALLY DEFERRED, "codename"
varchar(100) NOT NULL, "name" varchar(255) NOT NULL);

CREATE UNIQUE INDEX "auth_permission_content_type_id_codename_01ab375a_uniq" ON
"auth_permission" ("content_type_id", "codename");

CREATE INDEX "auth_permission_content_type_id_2f476e4b" ON "auth_permission"
("content_type_id");

CREATE TABLE IF NOT EXISTS "auth_user" ("id" integer NOT NULL PRIMARY KEY
AUTOINCREMENT, "password" varchar(128) NOT NULL, "last_login" datetime NULL,
"is_superuser" bool NOT NULL, "username" varchar(150) NOT NULL UNIQUE, "first_name"
varchar(30) NOT NULL, "email" varchar(254) NOT NULL, "is_staff" bool NOT NULL,
"is_active" bool NOT NULL, "date_joined" datetime NOT NULL, "last_name" varchar(150)
NOT NULL);

CREATE TABLE IF NOT EXISTS "auth_group" ("id" integer NOT NULL PRIMARY KEY
AUTOINCREMENT, "name" varchar(150) NOT NULL UNIQUE);

CREATE TABLE IF NOT EXISTS "main_event" ("id" integer NOT NULL PRIMARY KEY
AUTOINCREMENT, "event_title" varchar(200) NOT NULL, "event_brief" text NOT NULL,
"event_published" datetime NOT NULL, "event_category_id" integer NOT NULL
REFERENCES "main_eventcategory" ("id") DEFERRABLE INITIALLY DEFERRED,
"event_date" datetime NOT NULL, "event_image" varchar(100) NOT NULL, "max_Team_size"
integer unsigned NOT NULL CHECK ("max_Team_size" >= 0), "organizer_id_id" integer NOT
NULL REFERENCES "main_organizer" ("id") DEFERRABLE INITIALLY DEFERRED,
"event_description" text NOT NULL);

CREATE INDEX "main_event_event_category_id_99a8e4a8" ON "main_event"
("event_category_id");

CREATE INDEX "main_event_organizer_id_id_f84c7b60" ON "main_event"
("organizer_id_id");
```

```
CREATE TABLE IF NOT EXISTS "django_session" ("session_key" varchar(40) NOT NULL
PRIMARY KEY, "session_data" text NOT NULL, "expire_date" datetime NOT NULL);

CREATE INDEX "django_session_expire_date_a5c62663" ON "django_session"
("expire_date");

CREATE TABLE IF NOT EXISTS "main_eventregistration" ("id" integer NOT NULL
PRIMARY KEY AUTOINCREMENT, "team_name" varchar(200) NOT NULL,
"registration_time" datetime NOT NULL, "team_members_id" text NOT NULL,
"team_members_age" text NOT NULL, "team_size" integer unsigned NOT NULL CHECK
("team_size" >= 0), "event_id_id" integer NOT NULL REFERENCES "main_event" ("id")
DEFERRABLE INITIALLY DEFERRED, "user_Id_id" integer NOT NULL REFERENCES
"auth_user" ("id") DEFERRABLE INITIALLY DEFERRED);

CREATE INDEX "main_eventregistration_event_id_id_64d5415e" ON
"main_eventregistration" ("event_id_id");

CREATE INDEX "main_eventregistration_user_Id_id_3cae12f3" ON "main_eventregistration"
("user_Id_id");

CREATE TABLE IF NOT EXISTS "main_eventcategory" ("id" integer NOT NULL PRIMARY
KEY AUTOINCREMENT, "event_category" varchar(200) NOT NULL UNIQUE);

CREATE TABLE IF NOT EXISTS "main_organizer" ("id" integer NOT NULL PRIMARY
KEY AUTOINCREMENT, "Organizer_name" varchar(200) NOT NULL UNIQUE,
"user_Id_id" integer NOT NULL REFERENCES "auth_user" ("id") DEFERRABLE
INITIALLY DEFERRED);

CREATE INDEX "main_organizer_user_Id_id_f7127034" ON "main_organizer" ("user_Id_id");
```

Test Cases

Black box test case

For search engine

1. Type vague name of the contest - Pulls down suggestions based on contest name.
2. Type vague name of the contest - Pulls down suggestions based on the frequency of use, match in word and containing related words.
3. Advance use - Typing symbols for better-curated results
   - & Will match the word before and after it in contest name or details.
   - | Will match the word before or after it in contest name or details.
   - - Will remove the next word from the suggestion.
   - * or % Will match any word in its place in contest name or details. (* will specify the size of word and % will specify no of words)
   - “ ” Will match the whole word in contest name or details without other parameters.
   - in:<Category,> Will match the word after it in contest name or details in a particular category only.
   - All these can be used simultaneously
4. Typing nothing - Shows most-viewed contest
5. Typing SQL injection text - ignores the text

White box test case
Home Page test

1. Left and right button clicks in the poster zone and information zone - Changes posters in the loop
2. Hover over posters
    - Shows more detail on the poster (Pop-out)
3. Click on posters
    - Opens contest page
4. Click on the site logo
    - Opens site details (About us)
5. Click on/Hover over the Profile icon
    - If Logged in hovering shows basic profile details, Clicking takes to the profile page. Moving cursor or clicking again takes back to the previous state
    - Else hovering/Clicking pops/opens login menu/page. Moving cursor or clicking again takes back to the previous state
6. Click on the Search button
    - Drops a search bar (Click again to retract)
7. Click on/Hover over hamburger icon

    Opens short menu
    - Moving cursor or clicking again takes back to the previous state
8. Simultaneous use of the icons

Profile Page Test

1. Hover/Click on profile icon or name
    - - Shows basic profile details/ Moves to profile details page
2. Click Menu or Items in Menu (eg. Events)
    - - Goes to consolidated menus (if Menu is clicked)
    - - Goes to respective Items' page
3. <Here we take that page as Events for ease of testing>

4. Hover/Click on <Number> Participating
    - Shows the enrolled events (First Upcoming)
5. Click on Ongoing(/Favourite/Past)
    - Shows the ongoing events and friends participating in it
    - Favourites show the events that have been added to favourites
    - Past shows all the past events that were registered for (and their upcoming editions if any)
6. Click on posters
    - Takes to the contest page
7. In Progress section
    - Hover over contests
    - Pops the contest basic details and the requirements that are not met
    - Click on contestsShows consolidated reports on the progress, requirements and scores in rounds (if applicable).
8. Click on the exit button
    - Returns to homepage

Conclusion

Websites like competition hosting ones are a great use in the present scenario ie during COVID where people mostly try to participate or create competitions online. It can also be used for increasing the number of members participating in playing an important part in the improvement of the business.

We focused on UI perspective as 70% of the users are captured by the UI design. With this, our project has given us the opportunity to understand more about how full-stack web development works.