Table of Contents

ABSTRACT:

Chat application is a feature or a program on the Internet to communicate directly among Internet users who are using the internet. Chat applications allow users to communicate even though from a great distance. Therefore, this chat application must be real-time and multi platform to be used by many users. In this chat application we can Select the Interest from the menu and find the person with the common interests and send him Hello. We can get so much information about our interests .The Technologies used to build this application are HTML, CSS , JS with Django framework and PostgreSQL database.

INTRODUCTION:

Communication is a mean for people to exchange messages. It has started since the beginning of human creation. Distant communication began as early as 1800 century with the introduction of television, telegraph and then telephony. Interestingly enough, telephone communication stands out as the fastest growing technology, from fixed line to mobile wireless, from voice call to data transfer. The emergence of computer network and telecommunication technologies bears the same objective that is to allow people to communicate . All this while, much efforts has been drawn towards consolidating the device into one and therefore indiscriminate the services. Chatting is a method of using technology to bring people and ideas together despite of the geographical barriers. The technology has been available for years but the acceptance it was quite recent. Our project is an example of a chat server. It is an App based on the common interests. We can Select the Interest from the menu and find the person with the common interests and we can get so much information about our interests.

METHODOLOGY:

Technologies used

Hypertext Markup Language (HTML):

HTML is the standard markup language for documents designed to be displayed in a Web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.

Web Browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as <img> and <input> directly introduce content into the page. Other tags such as <p> surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

CASCADING STYLE SHEETS(CSS):

CSS is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

CSS is designed to enable the separation of presentation and content, including layout, colours, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate. CSS file which reduces complexity and repetition in the structural content as well as enabling the. CSS file to be cached to improve the page load speed between the pages that share the file and its formatting.

Separation of formatting and content also makes it feasible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice, and on Braille-Based tactile devices. CSS also has rules for alternate formatting if the content is accessed on a mobiledevices.

JAVA SCRIPT:

Java script is a programming language that conforms to the ECMA script specification.[7] JavaScript is high-level, often just-in-time compiled, and multi-paradigm. It has curly bracket syntax, object-orientation, and first-class-functions.

Alongside HTML and CSS, JavaScript is one of the core technologies of the World Wide Web JavaScript enables interactive web pages and is an essential part of web applications. The vast majority of websites use it for client-side page behavior,[9] and all major web browsers have a dedicated Java script engine to execute it.

As a multi-paradigm language, JavaScriptsupports eventdrive, functional, and imperative programming styles. It has APIs for working with text, dates, regular expressions, standard data structures, and DOM. However, the language itself does not include any I/O, such as networking, storage, or graphics facilities, as the host environment (usually a web browser) provides those APIs.

JavaScript engines were originally used only in web browsers, but they are now embedded in some servers, usually via Node.js. They are also embedded in a variety of applications created with frameworks such as Django.

DJANGO:

Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It’s free and open source.

Django's primary goal is to ease the creation of complex, database-driven websites. The framework emphasizes reusability and "pluggability" of components, less code, low coupling, rapid development. Python is used throughout, even for settings, files, and data models. Django also provides an optional administrative create, read, update, delete interface that is generated dynamically through introspection and configured via admin models.

Some well-known sites that use Django include Instagram, Mozilla, The Washington Times, Disgus, BitBucket.

PACKAGES USED:

psycopg2:

Psycopg is the most popular PostgreSQL database adapter for the Python programming language. Its main features are the complete implementation of the Python DB API 2.0 specification and the thread safety (several threads can share the same connection). It was designed for heavily multi-threaded applications that create and destroy lots of cursors and make a large number of concurrent “INSERT”s or “UPDATE”s.

Psycopg 2 is mostly implemented in C as a libpq wrapper, resulting in being both efficient and secure. It features client-side and server-side cursors, asynchronous communication and notifications, “COPY TO/COPY FROM” support. Many Python types are supported out-of-the-box and adapted to matching PostgreSQL data types; adaptation can be extended and customized thanks to a flexible objects adaptation system.

Psycopg 2 is both Unicode and Python 3 friendly.

Pyparsing:

The pyparsing module is an alternative approach to creating and executing simple grammars, vs. the traditional lex/yacc approach, or the use of regular expressions. The pyparsing module provides a library of classes that client code uses to construct the grammar directly in Python code.

python-decouple:

Decouple helps you to organize your settings so that you can change parameters without having to redeploy your app.

It also makes it easy for you to:

1. store parameters in *ini* or *.env* files;
2. define comprehensive default values;
3. properly convert values to the correct data type;
4. have **only one** configuration module to rule all your instances.

It was originally designed for Django, but became an independent generic tool for separating settings from code.

django-braces:

Mixins to add easy functionality to Django class-based views, forms, and models.

django-crispy-forms:

The best way to have Django DRY forms. Build programmatic reusable layouts out of components, having full control of the rendered HTML without writing HTML in templates. All this without breaking the standard way of doing things in Django, so it plays nice with any other form application.

The application mainly provides:

- A filter named `|crispy` that will render elegant div based forms. Think of it as the built-in methods: `as_table`, `as_ul` and `as_p`. You cannot tune up the output, but it is easy to start using it.
- A tag named `{% crispy %}` that will render a form based on your configuration and specific layout setup. This gives you amazing power without much hassle, helping you save tons of time.

Django-crispy-forms supports several frontend frameworks, such as Twitter Bootstrap (versions 2, 3, and 4), Uni-form and Foundation. You can also easily adapt your custom company's one, creating your own, see the docs for more information. You can easily switch among them using `CRISPY_TEMPLATE_PACK` setting variable.

django-model-utils:

To use `django-model-utils` in your Django project, just import and use the utility classes described in this documentation; there is no need to modify your `INSTALLED_APPS` setting.

django-turbolinks:

There are several ways for improving the speed of your website and get faster page loads, among the most effective ones is preventing full page reloads and only loading the portions of the page that change.

Turbolinks and PJAX are libraries that do exactly that, they load page contents via AJAX and replace a container with the HTML that's returned from the server, in this way the client doesn't have to load all JavaScript libraries and styles again, but only loads the part of the page that changed. Turbolinks and PJAX are very popular in the Rails community but haven't been quite used in Django.

pytz:

pytz brings the Olson tz database into Python. This library allows accurate and cross platform timezone calculations using Python 2.4 or higher. It also solves the issue of ambiguous times at the end of daylight saving time, which you can read more about in the Python Library Reference (`datetime.tzinfo`).

Note:

This library differs from the documented Python API for tzinfo implementations; if you want to create local wallclock times you need to use the `localize()` method documented in this document. In addition, if you perform date arithmetic on local times that cross DST boundaries, the result may be in an incorrect timezone (ie. subtract 1 minute from 2002-10-27 1:00 EST and you get 2002-10-27 0:59 EST instead of the correct 2002-10-27 1:59 EDT). A `normalize()` method is provided to correct this. Unfortunately these issues cannot be resolved without modifying the Python datetime implementation (see PEP-431).

sqlparse:

sqlparse is a non-validating SQL parser for Python. It provides support for parsing, splitting and formatting SQL statements.

The module is compatible with Python 3.5+ and released under the terms of the New BSD license.

Visit the project page at https://github.com/andialbrecht/sqlparse for further information about this project.

asgiref:

Contains various reference ASGI implementations, including:

- A base channel layer, `asgiref.base_layer`
- An in-memory channel layer, `asgiref.inmemory`
- WSGI-to-ASGI and ASGI-to-WSGI adapters, in `asgiref.wsgi`

Base Channel Layer

Provides an optional template to start ASGI channel layers from with the two exceptions you need provided and all API functions stubbed out.

Also comes with logic for doing per-channel capacities using channel names and globbing; use `self.get_capacity` and pass the arguments through to the base `__init__` if you want to use it.

In-memory Channel Layer

Simply instantiate `asgiref.inmemory.ChannelLayer`, or use the pre-made `asgiref.inmemory.channel_layer` for easy use. Implements the `group` extension, and is designed to support running multiple ASGI programs in separate threads within one process (the channel layer is threadsafe).

WSGI-ASGI Adapters

These are not yet complete and should not be used.

DATABASE USED:

POSTGRESQL:

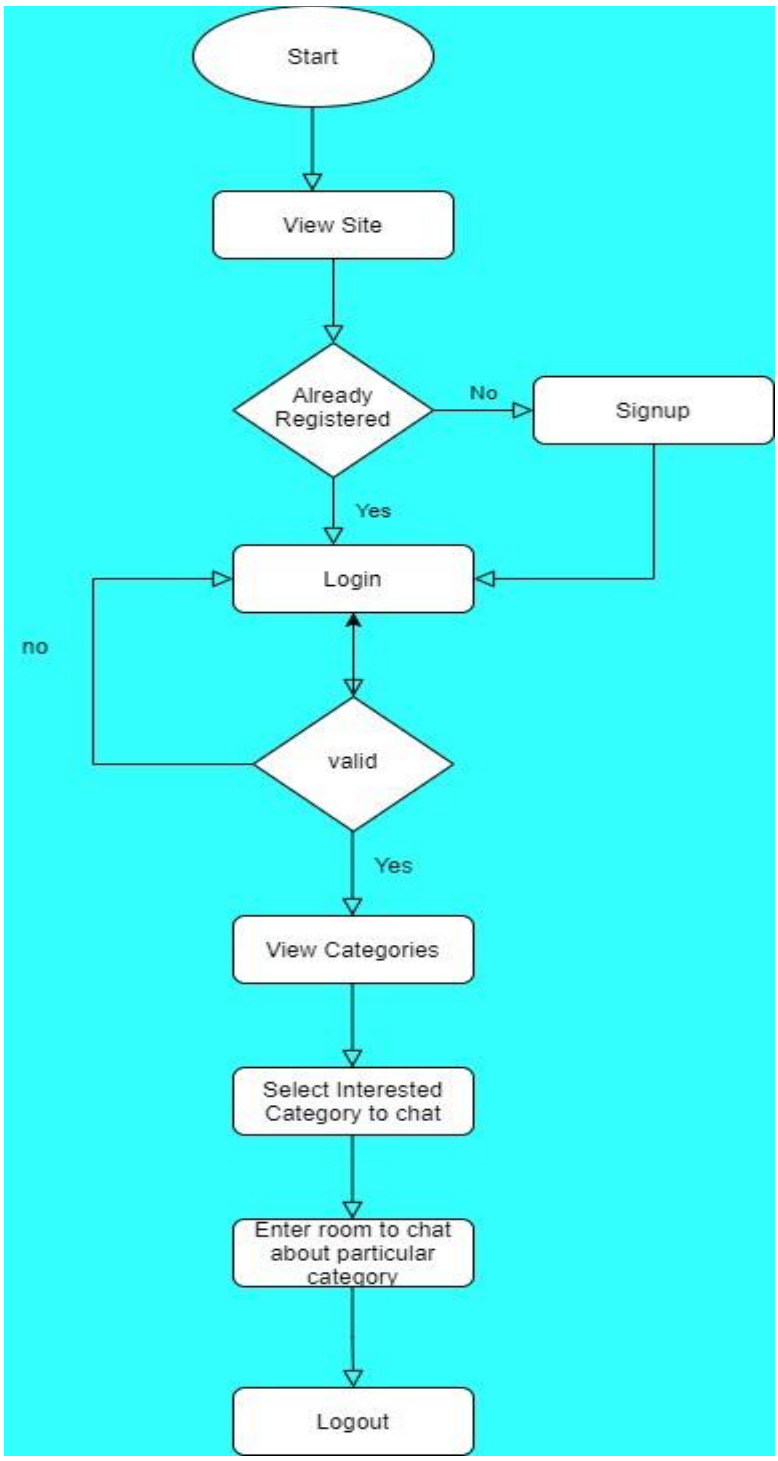
PostgreSQL is a powerful, open source object-relational database system that uses and extends the SQL language combined with many features that safely store and scale the most complicated data workloads. The origins of PostgreSQL date back to 1986 as part of the postgres project at the University of California at Berkeley and has more than 30 years of active development on the core platform.

PostgreSQL has earned a strong reputation for its proven architecture, reliability, data integrity, robust feature set, extensibility, and the dedication of the open source community behind the software to consistently deliver performant and innovative solutions. PostgreSQL runs on all major operating systems, has been ACID -compliant since 2001, and has powerful add-ons such as the popular postgis geospatial database extender. It is no surprise that PostgreSQL has become the open source relational database of choice for many people and organisations.

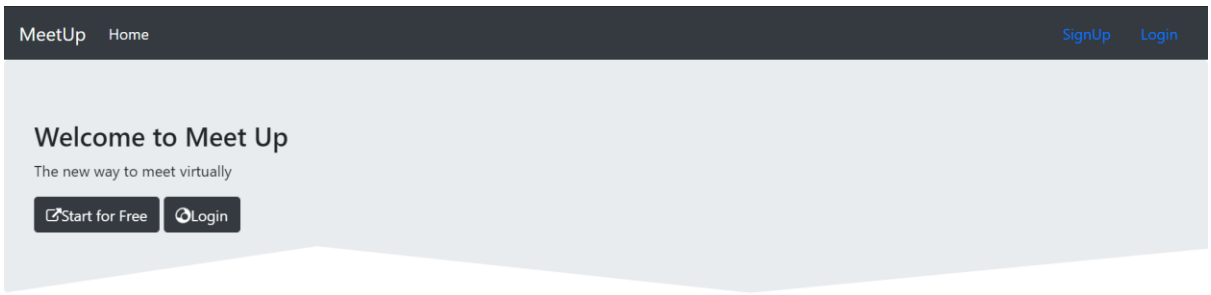
PostgreSQL comes with many features aimed to help developers build applications, administrators to protect data integrity and build fault-tolerant environments, and help you manage your data no matter how big or small the dataset. In addition to being free and open source, PostgreSQL is highly extensible.

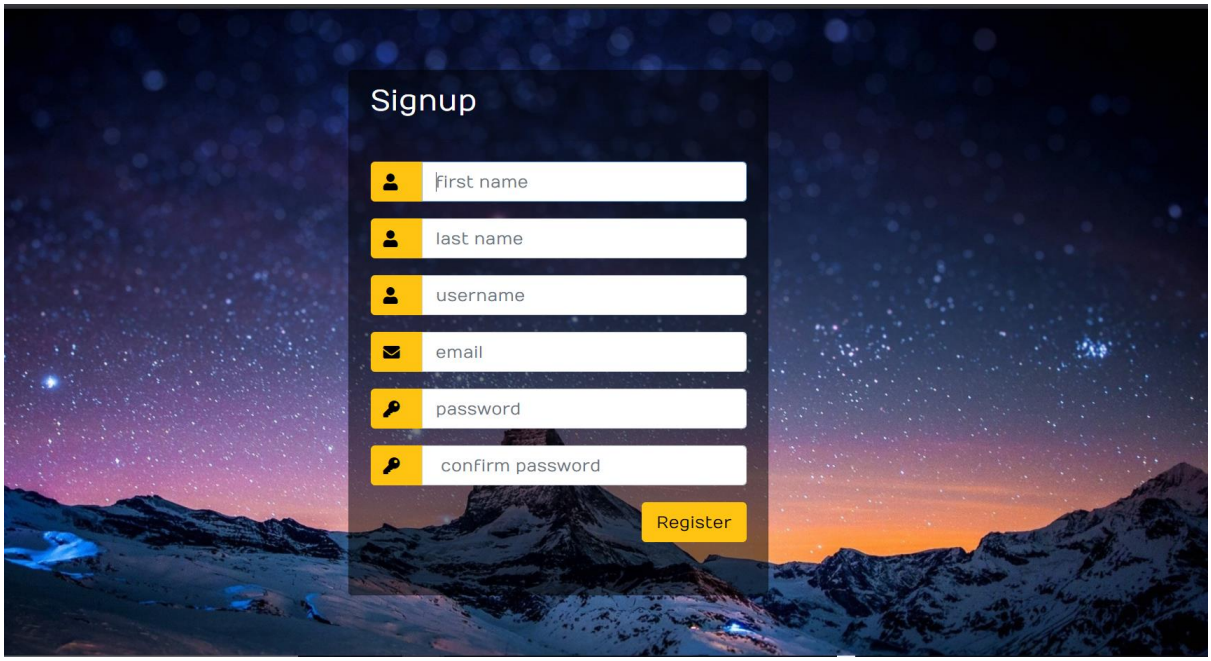
NAME: Pushpak Ganagoni **CBIT** **Rollno:**160118733107

Block Diagram:

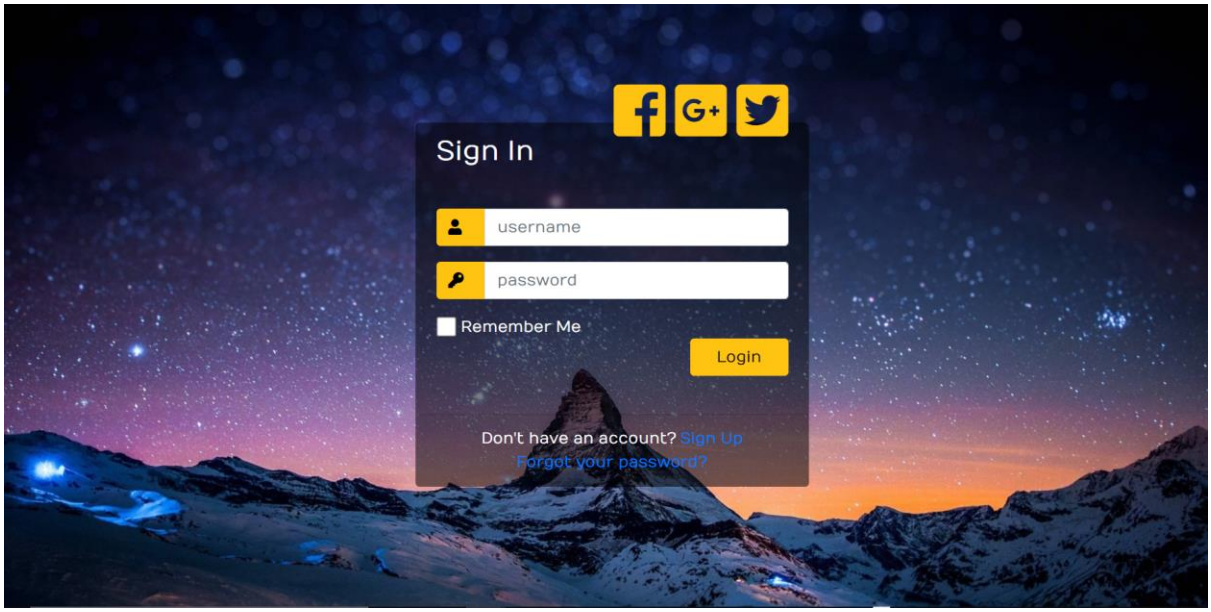


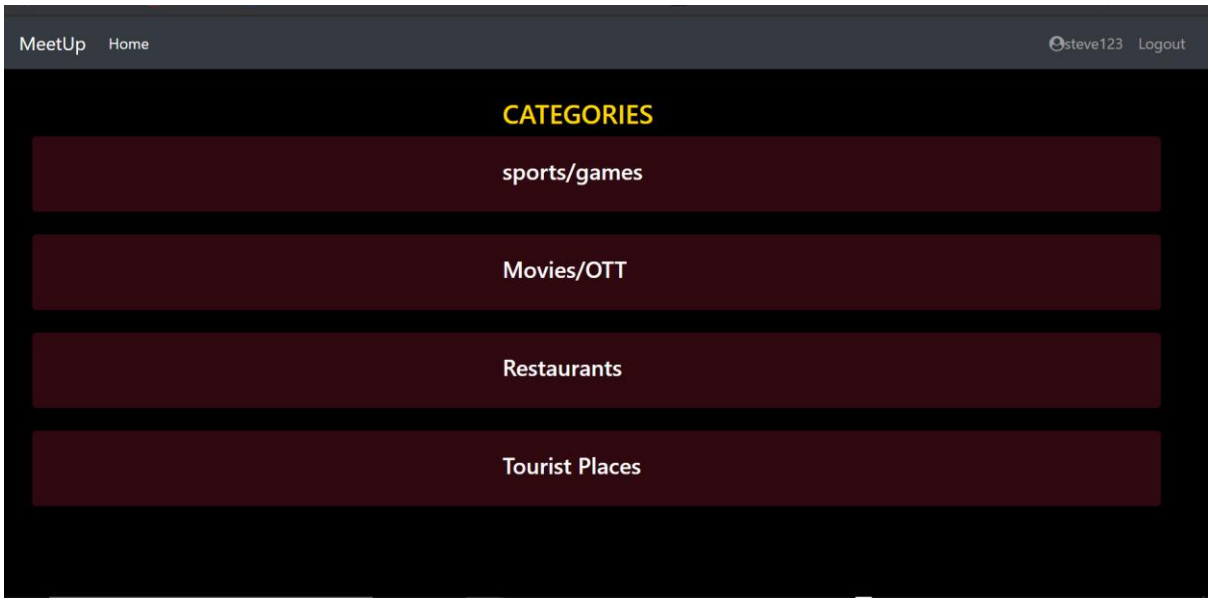
IMPLEMENTATION:

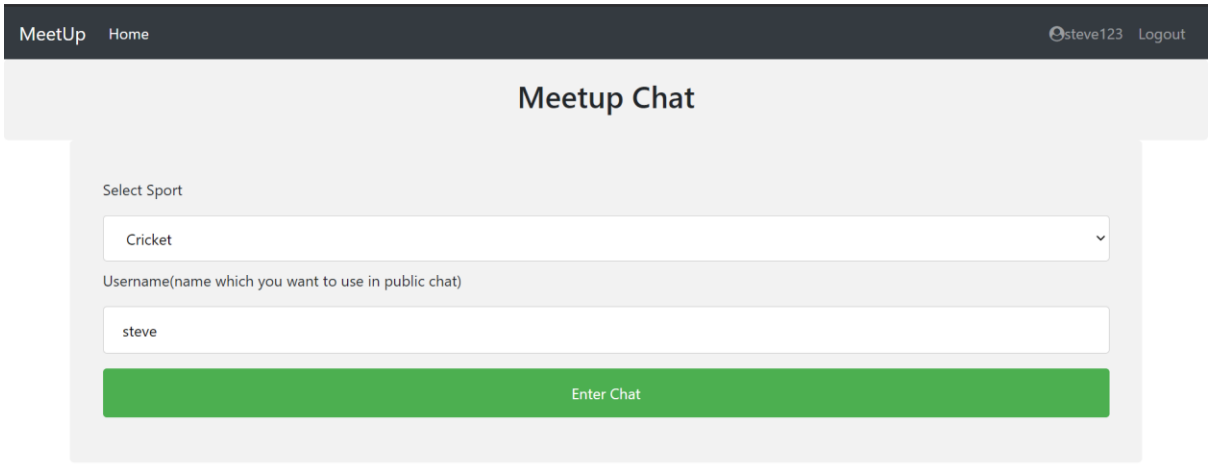


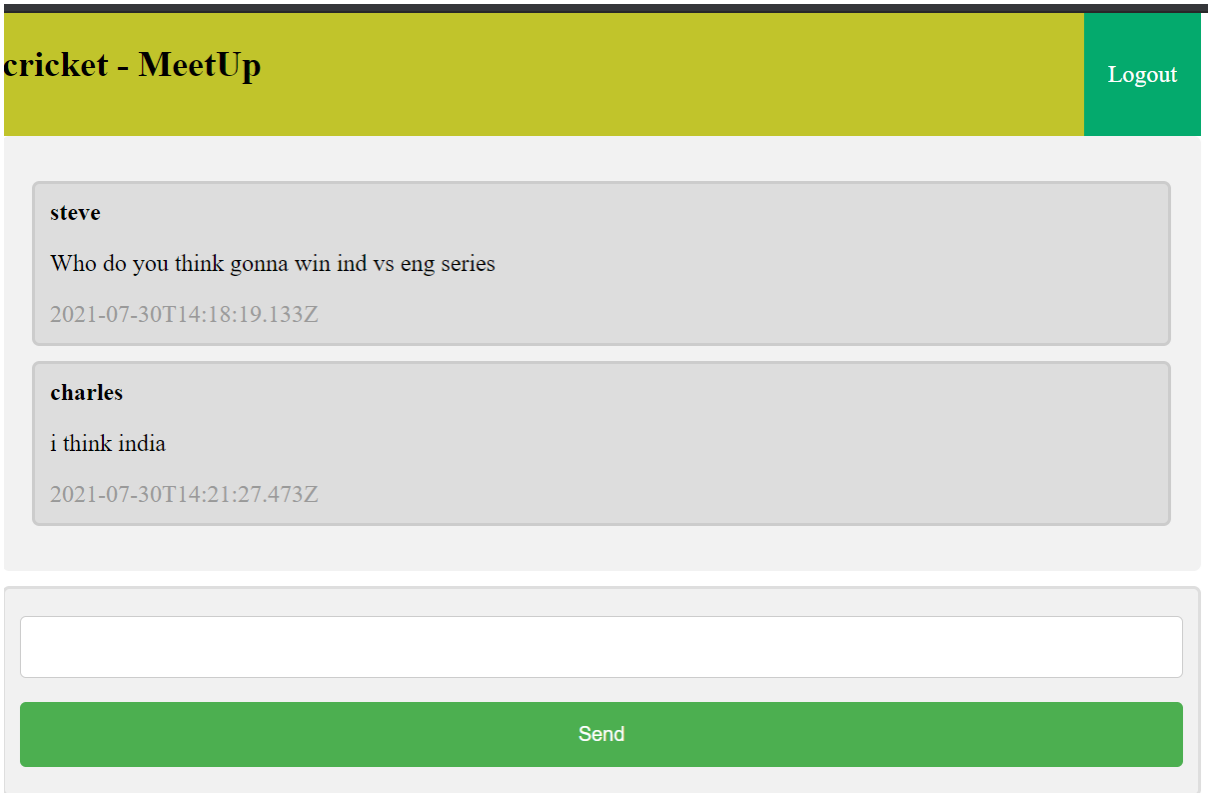


NAME: Pushpak Ganagoni **CBIT** **Rollno:**160118733107






MeetUp
Home
steve123
Logout
Meetup Chat
Select Sport
Cricket
Username(name which you want to use in public chat)
steve
Enter Chat


cricket - MeetUp
Logout
steve
Who do you think gonna win ind vs eng series
2021-07-30T14:18:19.133Z
charles
i think india
2021-07-30T14:21:27.473Z
Send

Conclusion:

The main objective of the project is to develop a Chat Application. I had taken a wide range of literature review in order to achieve all the tasks, where I came to know about some of the products that are existing in the market. I made a detailed research in that path to cover the loop holes that existing systems are facing and to eradicate them in our application.

Meetup is a web based Application which helps users to chat with different people based on their interests.

Future Scope:

With the knowledge I have gained by developing this application, I am confident that in the future I can make the application more effectively by adding this services.

- Extending this application by providing Authorisation service.
- Making it more user friendly by improving UI
- Increasing the effectiveness of the application by providing Voice Chat.
- Extending it to Mobile App Support.