DECLARATION

We hereby declare that the work reported in the B. Tech. project entitled as "**Peer to Peer knowledge sharing platform**", in partial fulfillment for the award of degree of B. Tech submitted at Jaypee University of Engineering and Technology, Guna, as per best of my knowledge and belief there is no infringement of intellectual property right and copyright. In case of any violation, we will solely be responsible.

Signature of the student
Department of computer science and engineering
Jaypee University of engineering and technology
Guna, M.P., India
Date: 6/12/2019

CERTIFICATE

This is to certify that the work titled "Peer to Peer knowledge sharing platform" submitted by "**Utkarsh Mishra**", "**Vikrant Srivastava**", "**Yash Dwivedi**", in partial fulfillment for the award of degree of B.Tech of Jaypee University of Engineering & Technology, Guna has been carried out under my supervision. As per best of my knowledge and belief there is no infringement of intellectual property right and copyright. Also, this work has not been submitted partially or wholly to any other University or Institute for the award of this or any other degree or diploma. In case of any violation concern student will solely be responsible.

Signature of the Supervisor
(Dr. Neelesh Kumar Jain)
Date: 6/12/2019

ACKNOWLEDGEMENT

We would like to express our sincere thanks and gratitude to our advisor Dr. Neelesh Kumar Jain for his excellent guidance on our B. Tech. major project entitled "**Peer to Peer Knowledge Sharing Platform**". His kind encouragement, compassionate and invaluable detailed suggestion and supervision throughout the phases of project development, presentation, and paper organization helped us to successfully accomplish the goal. Secondly, we would like to express our gratitude to flask development course for helping us in understanding the basics of flask API development and other necessary components for completion of our project. Last, but not the least, we are sincerely grateful to our family, and friends, who gave their valuable suggestions and support to complete the project. The whole project work would not be possible without the active encouragement from all of them.

Signature of the Student

(Utkarsh Mishra)

(Vikrant Srivastava)

(Yash Dwivedi)

Date: 6/12/2019

EXECUTIVE SUMMARY

Peer to Peer knowledge sharing platform is way to unite learners across the globe with absolutely no constraints. The idea is to promote that knowledge should be free and highly accessible. The project uses different technological paradigms such as the concept of the peer to peer networking along with some development methodologies like development using REST APIs for the working.

The project not only enables the users to share knowledge over the platform but also encourages the use of technology to create such helpful tools in future. It can also help the users to harness the power of anonymous communication in a good way for the betterment of the knowledge sharing.

It doesn't require any central authority running the sharing and sync services so in PeerWeb basically all the users that connect to and use it become one of the many contributors to solve the problem and that's why this can be a very great step towards making the knowledge free and accessible.

SOFTWARE AND HARDWARE REQUIREMENTS

Software Requirements:

- Python
- Flask
- Web Browser (Chrome/Firefox/IE/Safari)

Hardware Requirements:

- Dual core Process (or above)
- Operating System: 64-bit
- RAM: 2GB or above
- Memory: 500MB free Disk

TABLE OF CONTENTS

Chapter 1

Introduction

1.1 Overview:

The project helps to solve the knowledge sharing problem in the current scenario with the help of the concepts of peer to peer communication. It also uses concepts of json for handling critical data and uses HTTP requests for achieving the services at the end to end level.

The development in this project was performed using Python, Flask, Requests and json library. In the remainder of this section we will first describe the PeerWeb, then we will provide an overview of how it works.

1.2 Description:

Peer to Peer knowledge sharing platform is way to unite learners across the globe with absolutely no constraints. The idea is to promote that knowledge should be free and highly accessible. The project uses different technological paradigms such as the concept of the peer to peer networking along with some development methodologies like development using REST APIs for the working.

The project not only enables the users to share knowledge over the platform but also encourages the use of technology to create such helpful tools in future. It can also help the users to harness the power of anonymous communication in a good way for the betterment of the knowledge sharing.

It doesn't require any central authority running the sharing and sync services so in PeerWeb basically all the users that connect to and use it become one of the many contributors to solve the problem and that's why this can be a very great step towards making the knowledge free and accessible.

- Peer Sync service
  PeerWeb has certain peer services to discover and communicate with peers. peer-sync-service is one of them, this service is used to sync the peers between two nodes on the network.
- Peer health service

This service is used to check whether the PeerWeb services are running on the remote note or not.

- Peer add service

  This service is internally used by the peer sync service. This is basically implemented using a POST method and it helps to create a peer entry on the remote node.
- Peer get service

  This service is also internally used by peer sync service. This is implemented using a GET method and it helps to get a list of all the peer entries of the remote node.
- Status service

  This service shows the current service status of the current node this is used to get a quick health check of all the services that are enabled using the PeerWeb.

- File share service

  This service is used to share files between nodes running peerweb. This is also internally used by File Sync service.

- File Sync service

  This service is used to sync the file content between two nodes of peerweb and then content that need to be transferred are stored in a json file and a diff is calculated between the two nodes and based on the diff the files are synced.

Chapter 2

Related Work

2.1 Existing System:

A centralized network architecture is built around a single server that handles all the major processing. Less powerful workstations connect to the server and submit their requests to the central server rather than performing them directly. This can include applications, data storage, and utilities. A knowledge sharing platform (KSP) is one where users can interact with each other to share knowledge and contribute formal and informal information to the knowledge base. You can also call it a knowledge management platform that allows users to connect with each other for close collaboration and unhindered flow of information among them about products, services, and internal operations of the company. It does not matter what type of organization you are because every company in the world faces internal and external challenges that it has to manage efficiently. The issue could be as simple as sending multiple users on leave and finding a way to manage headcount without compromising on the delivery of services. If you are a tech company providing IT related solutions to your clients, you might want the resolutions to the most complex problems at hand. When you have a knowledge sharing tool, you can do so effortlessly.

Some key advantages to centralized network management are consistency, efficiency, and affordability. Network administrators are under pressure to keep machines patched and up-to-date, so having one central server control the whole network means less IT management time and fewer admins. In addition, all the data on a centralized network is required to go through one place, so it’s very easy to track and collect data across the network. Centralized networks do have their downsides; for example, a single point of failure can be a risk factor for organizations. If the central—or master—server goes down, the individual “client” machines attached to it are unable to process user requests. The impact of this failure will depend on how much the server processes. If the client machines do little more than submit requests, system availability can be totally compromised. They also offer limited scalability. Because all applications and processing power are housed in a single server, the only way to scale your network is to add more storage, I/O bandwidth, or processing power to the server. This may not turn out to be a cost-effective solution in the long run.

2.2 Problems with Existing System

A lack of bandwidth can also become a handicap. If you have a business with fluctuating periods of activity, a single server can soon prove a bottleneck, as it can be difficult to keep up with the influx of concurrent user requests—and the number of requests the system can really process and as we are aiming to connect so many people this is the major problem

In most cases, a school or school district will have a leading group of people as a part of central administration. In a school district, these terms may include a Superintendent (education), chief operating officer, school headmaster, and/or other leadership roles in one or more specific department. People on central administration are usually appointed by a board, such as a Board of education. They are comparable to positions such as a Chief executive officer. They rank over all other administration, requiring leadership skills. Central administrative staff have an executive oversight and supervision on school and/or school district administration. The department exists in Universities as well again playing a key role in the organization of the department. The department is often also tasked with data protection, disaster control planning and other areas.

Central administration can refer both to people within a department as well as consoles, applications, and other tools that help its function. Central administration is part of Windows SharePoint server; it allows system administrators or those within Central admin departments the ability to priorities various tasks as well as allowing users to view resources and currently running services. Understanding the distinct variables and processes that make up these reforms can be adaunting task. Using key questions as an organizing device, this document identifies and explains the fundamental issues, goals, processes, and strategies that shape educational decentralization initiatives in Latin America. In addition, the report discusses the possibilities and pitfalls associated with decentralization processes, particularly as they are associated with political, financial, institutional, and educational quality issues. The document concludes with a series of policy considerations that can help guide the thinking and planning of leaders who are involved in decentralizing a public educational system. In evaluating the material presented here, it is important to note that because the countries of Latin America are so different in their political, economic, and social makeup, the historical experience (good or bad) of a strategy introduced in one country is not necessarily predictive of what might happen in another.

2.3 Improvements

Improving the quality of education is typically a key objectives of decentralization. However, numerous studies have concluded that "while parents, students, and educators appear to be more satisfied under decentralization, it is still unclear whether, and under what circumstances, it makes any real difference in levels of student attainment of academic or social objectives."24 Several analysts point out that there are simply too many intervening variables, such as parental attitudes, peer group support, administrative training, resource availability and teacher motivation, and this complicates measurement. Thus, direct cause (decentralization) and effect (test scores) relationships are of questionable validity. In the words of one expert: "This transfer of power provides the opportunity, but not the guarantee, for the quality of school decision making and action to benefit

Chapter 3

Proposed Work

Peer-to-peer (P2P) computing or networking is a distributed application architecture that partitions tasks or workloads between peers. Peers are equally privileged, equipotent participants in the application. They are said to form a peer-to-peer network of nodes. Peers make a portion of their resources, such as processing power, disk storage or network bandwidth, directly available to other network participants, without the need for central coordination by servers or stable hosts. Peers are both suppliers and consumers of resources, in contrast to the traditional client-server model in which the consumption and supply of resources is divided. Emerging collaborative P2P systems are going beyond the era of peers doing similar things while sharing resources, and are looking for diverse peers that can bring in unique resources and capabilities to a virtual community thereby empowering it to engage in greater tasks beyond those that can be accomplished by individual peers, yet that are beneficial to all the peers.

3.1 Must have features

- It's accessible and interactive

  These are definitely two of the most important features to look for, combined into one. The days when users were spending hours in cubicles are gone, as they are mobile and demand flexibility, so you need to choose a tool that can be accessed from anywhere, no matter the device.

  Also, knowledge sharing isn't a one-way process anymore. In order for users to do contribute, a platform should offer options like a chat or Q&A features, as well as a comments section, similar to the ones seen on social media.

  This way, anybody can interact with the already existing content and engage in healthy discussions, thus breaking down barriers between employees and promoting a general learning culture.

- It allows sharing content with external users

Your entire team works hard so the business can grow and this will lead to an impressive knowledge base, containing valuable information. But what's the whole purpose of having it, if it can't be read by everybody, including those who are not members of the platform?
A viable knowledge sharing solution features the ability to share content with external users, so you can track who opened the email and how long they spent studying the content. By doing this, you always know what kind of content is performing best and successfully driving prospects down the funnel, but also which parts need some fine adjustments.

3.2 P2P as a solution

Use of peer-to-peer technologies have been prevalent in sharing files and media over internet. This project proposes a new cooperative file sharing and storage system that allows users to store, share and find files over a robust and scalable network. This system is based on peer-to-peer technologies and provides a self-managed, decentralized, efficient, load-balanced and distributed way to store and retrieve files on peer nodes. Each node donates storage space to the system and also does routing of client messages based on protocol. This system uses replication based on erasure codes for increasing efficiency in providing persistent storage. The system also provides its users with a global hierarchical directory structure to browse for files like a file-system.

While P2P networks open a new channel for efficient downloading and sharing of files and data, users need to be fully aware of the security threats associated with this technology. Security measures and adequate prevention should be implemented to avoid any potential leakage of sensitive and/or personal information, and other security breaches. Before deciding to open firewall ports to allow for peer-to-peer traffic, system administrators should ensure that each request complies with the corporate security policy and should only open a minimal set of firewall ports needed to fulfil P2P needs. For end-users, including home users, care must be taken to avoid any possible spread of viruses over the peer-to-peer network.

3.3 Security

Peer-to-peer systems pose unique challenges from a computer security perspective.
Like any other form of software, P2P applications can contain vulnerabilities. What makes this particularly dangerous for P2P software, however, is that peer-to-peer applications act as servers as well as clients, meaning that they can be more vulnerable to remote exploits.

3.4 Distributed storage and search

There are both advantages and disadvantages in P2P networks related to the topic of data backup, recovery, and availability. In a centralized network, the system administrators are the only forces controlling the availability of files being shared. If the administrators decide to no longer distribute a file, they simply have to remove it from their servers, and it will no longer be available to users. Along with leaving the users powerless in deciding what is distributed throughout the community, this makes the entire system vulnerable to threats and requests from the government and other large forces. For example, YouTube has been pressured by the RIAA, MPAA, and entertainment industry to filter out copyrighted content. Although server-client networks are able to monitor and manage content availability, they can have more stability in the availability of the content they choose to host. A client should not have trouble accessing obscure content that is being shared on a stable centralized network. P2P networks, however, are more unreliable in sharing unpopular files because sharing files in a P2P network requires that at least one node in the network has the requested data, and that node must be able to connect to the node requesting the data. This requirement is occasionally hard to meet because users may delete or stop sharing data at any point.
In this sense, the community of users in a P2P network is completely responsible for deciding what content is available. Unpopular files will eventually disappear and become unavailable as more people stop sharing them. Popular files, however, will be highly and easily distributed. Popular files on a P2P network actually have more stability and availability than files on central networks. In a centralized network, a simple loss of connection between the server and clients is enough to cause a failure, but in P2P networks, the connections between every node must be lost in order to cause a data sharing failure. In a centralized system, the administrators are responsible for all data recovery and backups, while in P2P systems, each node requires its own backup system. Because of the lack of central authority in P2P networks, forces such as the

recording industry, RIAA, MPAA, and the government are unable to delete or stop the sharing of content on P2P systems.

3.5 Applications

- Content delivery

  In P2P networks, clients both provide and use resources. This means that unlike client-server systems, the content-serving capacity of peer-to-peer networks can actually increase as more users begin to access the content (especially with protocols such as Bittorrent that require users to share, refer a performance measurement study). This property is one of the major advantages of using P2P networks because it makes the setup and running costs very small for the original content distributor.

- File-sharing networks

  Many file peer-to-peer file sharing networks, such as Gnutella, G2, and the eDonkey network popularized peer-to-peer technologies.

- Peer-to-peer content delivery networks.

  Peer-to-peer content services, e.g. caches for improved performance such as Correli Caches. Software publication and distribution (Linux distribution, several games); via file sharing networks.

Chapter 4

System Analysis and Design

4.1 Analysis:

Among all policies in networks, PN selection policy and bandwidth allocation policy have a significant effect on the system's performance. The system is a new technology, although it uses P2P architecture at the server level, there are differences between it and P2P file sharing networks, particularly in some algorithms and protocols. To improve upon the centralization of the current approach and came up with a solution using the decentralized approach.

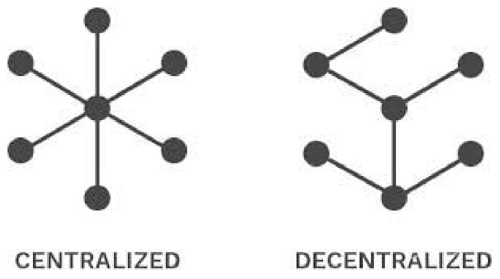
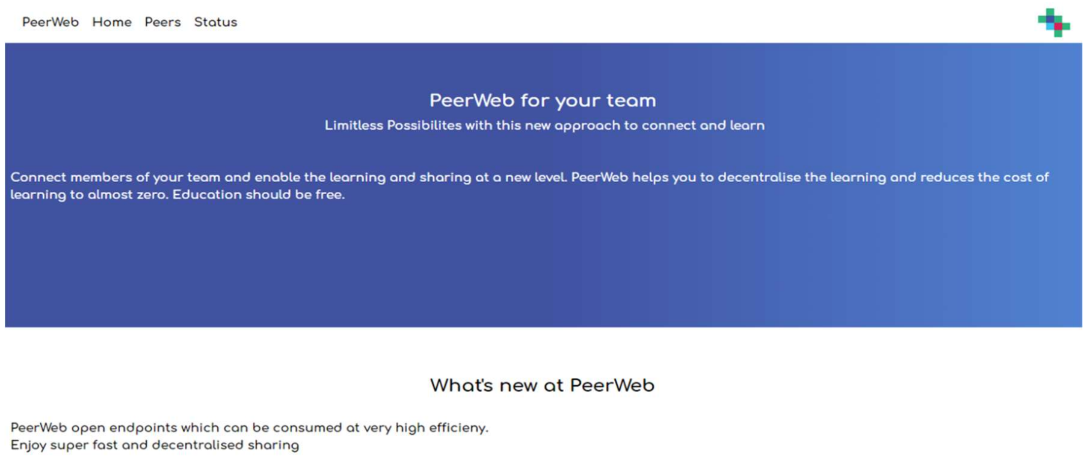


Fig 4.1 Structure of centralized and decentralized system

4.2 Design

- Landing page



PeerWeb for your team

Limitless Possibilites with this new approach to connect and learn

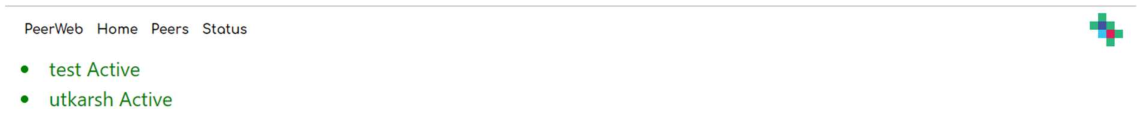
Connect members of your team and enable the learning and sharing at a new level. PeerWeb helps you to decentralise the learning and reduces the cost of learning to almost zero. Education should be free.

What's new at PeerWeb

PeerWeb open endpoints which can be consumed at very high efficieny.
Enjoy super fast and decentralised sharing

Fig 4.2 Landing Page of PeerWeb

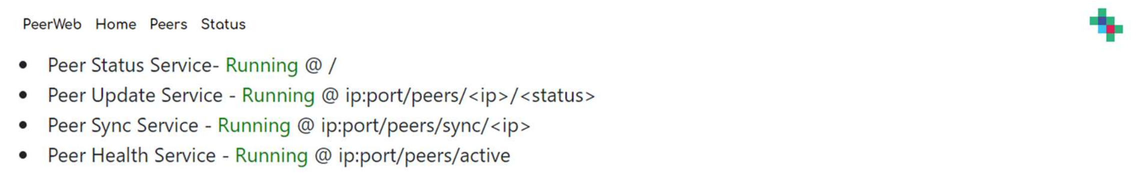
- Peers



- test Active
- utkarsh Active

Fig 4.3 Peers of a node

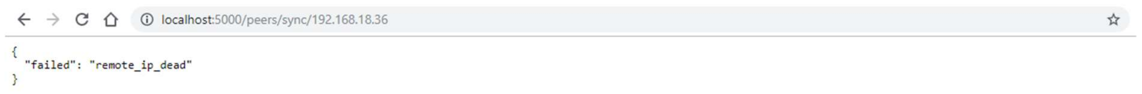
- Status



- Peer Status Service- Running @ /
- Peer Update Service - Running @ ip:port/peers/<ip>/<status>
- Peer Sync Service - Running @ ip:port/peers/sync/<ip>
- Peer Health Service - Running @ ip:port/peers/active

Fig 4.4 Status of services running on a node

- Sync



```
{
  "failed": "remote_ip_dead"
}
```

Fig 4.5 Sync service response when remote node is dead

CHAPTER 5

Implementation and Testing

5.1 Flask:

Flask is flexible. It doesn't require you to use any particular project or code layout. However, when first starting, it's helpful to use a more structured approach. This means that the tutorial will require a bit of boilerplate up front, but it's done to avoid many common pitfalls that new developers encounter, and it creates a project that's easy to expand on. Once you become more comfortable with Flask, you can step out of this structure and take full advantage of Flask's flexibility.

5.2 APIs

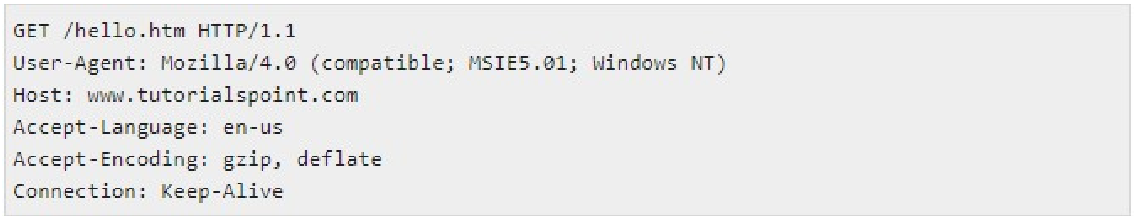
If you've heard the term API before, chances are it's been used not to refer to APIs in general, but instead to a specific kind of API, the web API. A web API allows for information or functionality to be manipulated by other programs via the internet. For example, with Twitter's web API, you can write a program in a language like Python or Javascript that can perform tasks such as favoring tweets or collecting tweet metadata.

In programming more generally, the term API, short for Application Programming Interface, refers to a part of a computer program designed to be used or manipulated by another program, as opposed to an interface designed to be used or manipulated In this tutorial, however, we'll be using the term API to refer specifically to web APIs.

5.3 HTTP Methods

- GET Method

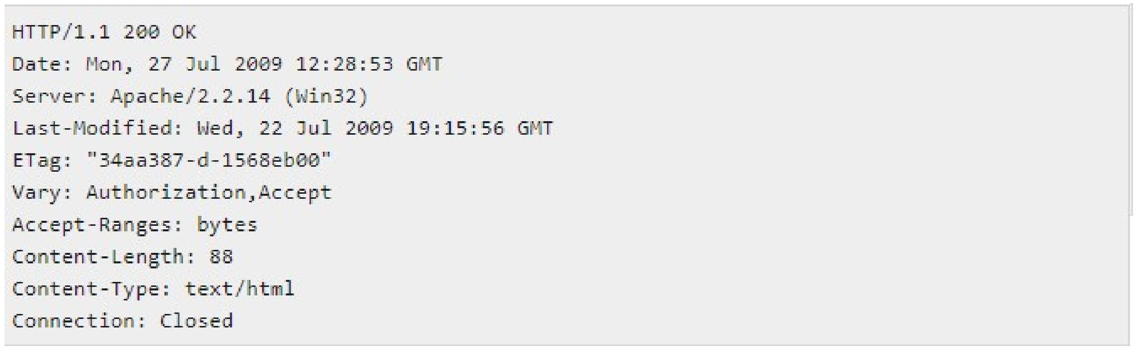
  A GET request retrieves data from a web server by specifying parameters in the URL portion of the request. This is the main method used for document retrieval. The following example makes use of GET method to fetch hello.htm



```
GET /hello.htm HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE5.01; Windows NT)
Host: www.tutorialspoint.com
Accept-Language: en-us
Accept-Encoding: gzip, deflate
Connection: Keep-Alive
```

Request format.

This is the how a GET request is made



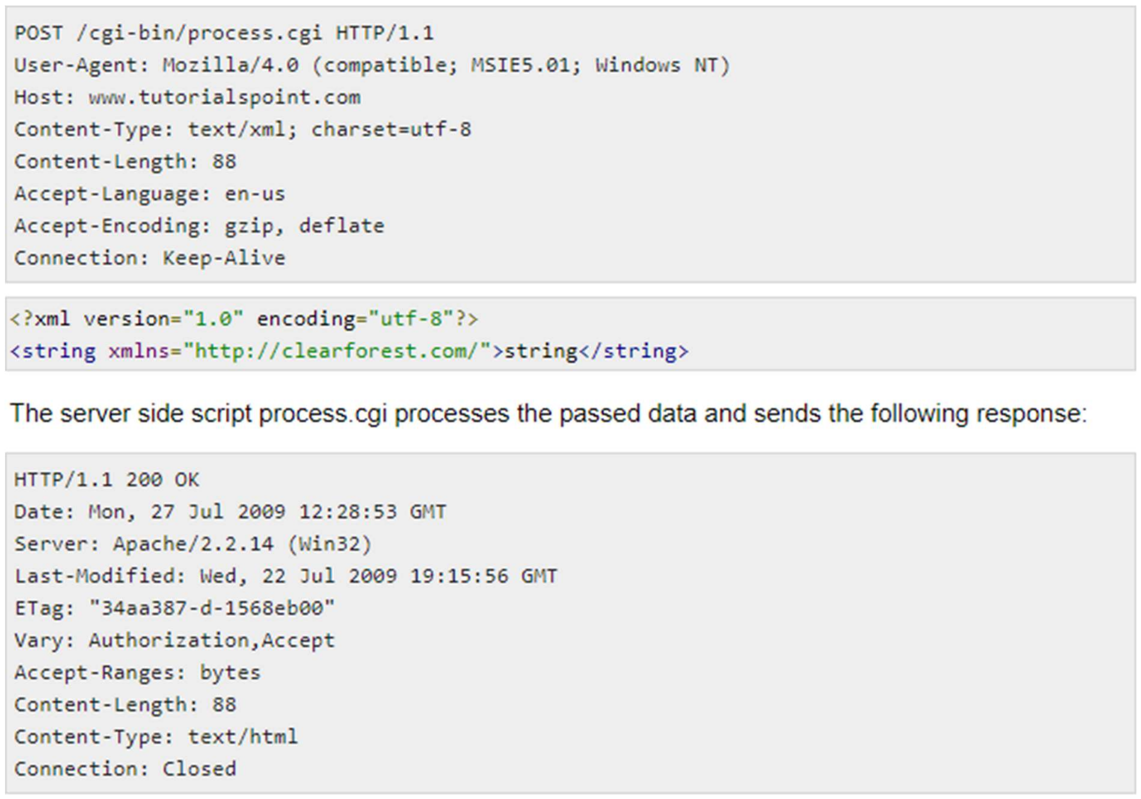
```
HTTP/1.1 200 OK
Date: Mon, 27 Jul 2009 12:28:53 GMT
Server: Apache/2.2.14 (Win32)
Last-Modified: Wed, 22 Jul 2009 19:15:56 GMT
ETag: "34aa387-d-1568eb00"
Vary: Authorization,Accept
Accept-Ranges: bytes
Content-Length: 88
Content-Type: text/html
Connection: Closed
```

This is the response based on the request sent.

- **POST**

The POST method is used when you want to send some data to the server, for example, file update, form data, etc. The following example makes use of POST method to send a form data to the server, which will be processed by a process.cgi and finally a response will be returned:

In our system this is used to save data to remote node and while syncing this is used to save content on our node.



```
POST /cgi-bin/process.cgi HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE5.01; Windows NT)
Host: www.tutorialspoint.com
Content-Type: text/xml; charset=utf-8
Content-Length: 88
Accept-Language: en-us
Accept-Encoding: gzip, deflate
Connection: Keep-Alive
```

```
<?xml version="1.0" encoding="utf-8"?>
<string xmlns="http://clearforest.com/">string</string>
```

The server side script process.cgi processes the passed data and sends the following response:

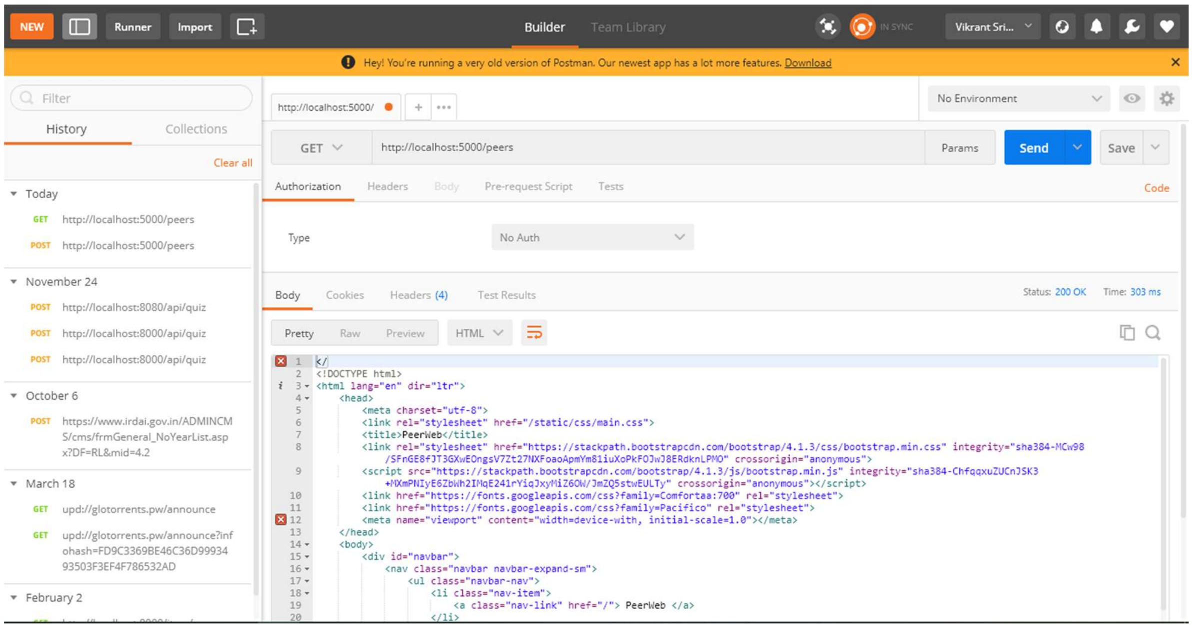
```
HTTP/1.1 200 OK
Date: Mon, 27 Jul 2009 12:28:53 GMT
Server: Apache/2.2.14 (Win32)
Last-Modified: Wed, 22 Jul 2009 19:15:56 GMT
ETag: "34aa387-d-1568eb00"
Vary: Authorization,Accept
Accept-Ranges: bytes
Content-Length: 88
Content-Type: text/html
Connection: Closed
```

This is the request and response of the POST request to the endpoints created using Flask

5.4 Implementation

We have used Postman for testing the endpoints of our project.

Our project requires two computers running peerweb and they should be on same network for local testing. If they are on local machine, then they can be easily contacted by the local address and the testing procedures can be performed using the POSTMAN or any other API testing software.


NEW
Runner
Import
Builder
Team Library
IN SYNC
Vikrant Sri...
Hey! You're running a very old version of Postman. Our newest app has a lot more features. Download
Filter
History
Collections
Clear all
Today
GET http://localhost:5000/peers
POST http://localhost:5000/peers
November 24
POST http://localhost:8080/api/quiz
POST http://localhost:8000/api/quiz
POST http://localhost:8000/api/quiz
October 6
POST https://www.irdai.gov.in/ADMINCMS/cms/frmGeneral_NoYearList.aspx?DF=RL&mid=4.2
March 18
GET upd://glotorrents.pw/announce
GET upd://glotorrents.pw/announce?infohash=FD9C3369BE46C36D9993493503F3EF4F786532AD
February 2
No Environment
Params
Send
Save
Authorization
Headers
Body
Pre-request Script
Tests
Code
Cookies
Test Results
Pretty
Raw
Preview
HTML

GET http://localhost:5000/peers

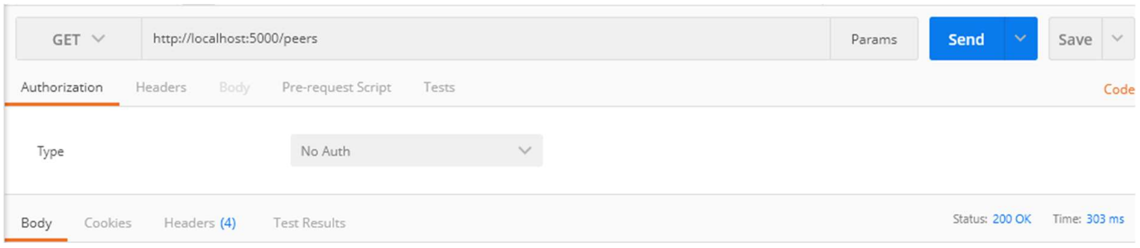
Type: No Auth

Headers (4) — Status: 200 OK — Time: 303 ms

```html
</
<!DOCTYPE html>
<html lang="en" dir="ltr">
    <head>
        <meta charset="utf-8">
        <link rel="stylesheet" href="/static/css/main.css">
        <title>PeerWeb</title>
        <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css" integrity="sha384-MCw98
            /SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO" crossorigin="anonymous">
        <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js" integrity="sha384-ChfqqxuZUCnJSK3
            +MXmPNIyE6ZbWh2IMqE241rYiqJxyMiZ6OW/JmZQ5stwEULTy" crossorigin="anonymous"></script>
        <link href="https://fonts.googleapis.com/css?family=Comfortaa:700" rel="stylesheet">
        <link href="https://fonts.googleapis.com/css?family=Pacifico" rel="stylesheet">
        <meta name="viewport" content="width=device-with, initial-scale=1.0"></meta>
    </head>
    <body>
        <div id="navbar">
            <nav class="navbar navbar-expand-sm">
                <ul class="navbar-nav">
                    <li class="nav-item">
                        <a class="nav-link" href="/"> PeerWeb </a>
                    </li>
```

Fig 5.1 POSTMAN client

Example:

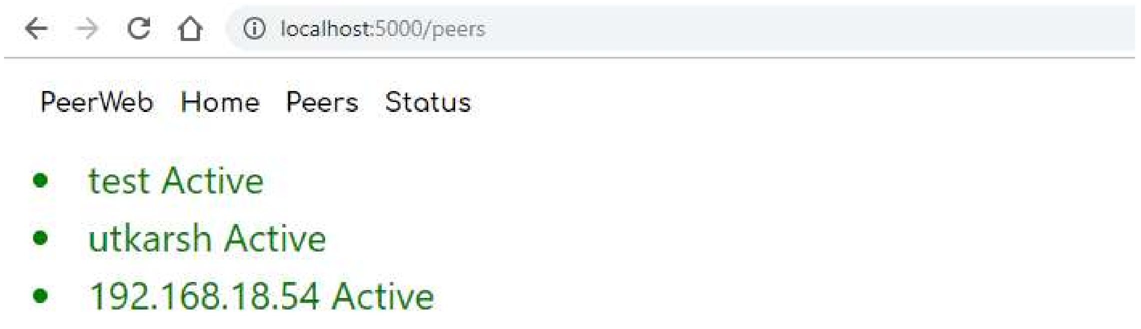


GET http://localhost:5000/peers

Type: No Auth

Headers (4) — Status: 200 OK — Time: 303 ms

Fig 5.2 Getting list of peers for node using a GET http request



localhost:5000/peers

PeerWeb Home Peers Status

- test Active
- utkarsh Active
- 192.168.18.54 Active

Fig 5.3 Getting peers using chrome as a client

6. Code of the Project:

6.1 Handling the routes

```
from flask import Flask, render_template

import flask

import json

from services.peer_discovery_service import get_active_peers

from services.peer_sync_service import send_peer_data, get_peer_data, sync_peer_data

from flask import request, jsonify

app = Flask(__name__)



@app.route('/')

def index():

    return render_template('about.html')



@app.route('/peers')

def all_peers():

    peerData = get_peer_data()

    peers = [peer for peer,status in peerData.items()]

    status = [status for peer, status in peerData.items()]

    return render_template('peers.html', peers = peers, status = status)
```

```
@app.route('/peers_json')

def all_peers_json():

    peerData = get_peer_data()

    return jsonify(peerData)



@app.route('/peers/active')

def discover_peers():

    return jsonify(get_active_peers())



@app.route('/peers/<ip>/<status>', methods = ['POST'])

def update_peer_list(ip, status):

     send_peer_data(ip, status)

     return jsonify({'success' : 'IP added'})



@app.route('/peers/sync/<ip>')

def sync_peer_list(ip):

    return jsonify(sync_peer_data(ip))



@app.route('/status')
```

```
def status():

    return render_template('status.html')



if __name__ == "__main__":

    app.run(debug=True, host = '0.0.0.0')
```

6.2 Peer discovery service

```
import os.path, sys
import json
import requests
import eventlet
from config import port

sys.path.append(os.path.join(os.path.dirname(os.path.realpath(__file__)), os.pardir))

def check_live_status(ip):
    endpoint = 'http://' + ip + ':' + port + '/status'
    try:
        with eventlet.Timeout(10):
            response = requests.get(endpoint)
        if response.status_code == 200:
            return True
        else:
            return False
    except:
        return False

def get_active_peers():
    with open('resources/peers.json') as data:
        peers = json.load(data)

    peerList = {}
    for ip, status in peers.items():
        if check_live_status(ip):
            peerList[ip] = "Active"

    with open('resources/peers.json') as data:
        peers = json.load(data)

    with open('resources/peers.json', mode = 'w') as file:
        for peer in peerList:
            peers[peer] = "Active"
        file.write(json.dumps(peers))
    return peerList
```

6.3 Peer Sync Service

```
import os.path, sys
sys.path.append(os.path.join(os.path.dirname(os.path.realpath(__file__)), os.pardir))
import json
import requests
from config import MY_IP, port

def sync_peer_data(remote_ip):
    #Merge remote peer data into local
    endpoint = 'http://' + remote_ip + ':5000' + '/peers_json'
    try:
        resp = requests.get(endpoint)
        remote_peer_list = {}
        if resp.status_code == 200:
            remote_peer_list = json.loads(resp.text)
    except:
        return {"failed" : "remote_ip_dead"}
    print(remote_peer_list)

    for ip, status in remote_peer_list.items():
        if ip != MY_IP:
            updateEndpoint = 'http://' + MY_IP + ':5000' + '/peers/' + ip + '/' + status
            updateStatus = requests.post(updateEndpoint)

    endpoint = 'http://' + MY_IP + ':5000' + '/peers_json'
    resp = requests.get(endpoint)
    remote_peer_list = {}

    if resp.status_code == 200:
        remote_peer_list = json.loads(resp.text)

    for ip, status in remote_peer_list.items():
        if ip != remote_ip:
            updateEndpoint = 'http://' + remote_ip + ':5000' + '/peers/' + ip + '/' + status
            updateStatus = requests.post(updateEndpoint)

    return {"success" : "Synced"}

def get_peer_data():
    with open('resources/peers.json') as data:
        peers = json.load(data)

    return peers

def send_peer_data(ip, status):
    with open('resources/peers.json') as data:
        peers = json.load(data)

    with open('resources/peers.json', mode = 'w') as file:
        peers[ip] = status
        file.write(json.dumps(peers))

    return {'success' : '200'}
```

These services interact with the methods to enable the service on the endpoints.

Implemented endpoints are –

- GET /
- GET /peers
- GET /peers_json
- GET /peers/active
- POST /peers/<ip>/<status>
- GET /peers/sync/<ip>
- GET /status

  etc

7. Conclusion

The developments of information technology have a high influence on knowledge sharing techniques. Poor information results in inadequate analysis, which leads to misguided policies on knowledge sharing. The current problem has many socio-economic, institutional and environmental aspects. Since, the use of knowledge-based systems and expert systems are dedicated in certain fields of application such as, Medical, Engineering, Control, Robots, manufacturing and what not.

In this study, we conclude that, the cooperation between the knowledge-based system and peer to peer system with the facilities provided by the application of p2p services will enhance the use of information technology in many fields, such as knowledge sharing, discussion and may other fields. Users are using platforms such as this to navigate in their life and learn things Moreover, these type of platforms exploit the two-dimensional capabilities of technology and present the information in a decentralized and accessible. Through the utilization of computer technology.

Our solutions help to solve the knowledge sharing problem in the current scenario with the help of the concepts of peer to peer communication. It also uses concepts of json for handling critical data and uses HTTP requests for achieving the services at the end to end level. The development in this project was performed using Python, Flask, Requests and json library. In the remainder of this section we will first describe the PeerWeb, then we will provide an overview of how it works.

8. References:

1. Collaboration and Knowledge Sharing Platform for Supporting a Risk Management Network of Practice (Katerina Papadaki ; Despina Polemi)

2. ISO/IEC 27005, Information Technology - Security Techniques - Information security risk management, Committee Draft, 2004.

3. Ghosh, T.: Creating Incentives for Knowledge Sharing. Technical report, MIT Open Courseware, Sloan school of management, Cambridge, Massachusetts, USA, 2004.

4. Brown, J S & Duguid, P, "Knowledge and organization: A social-practice perspective", Organization Science, 12, 2: 198-213, 2001.

5. Galuba, Wojciech; Girdzijauskas, Sarunas (2009), "Peer-to-Peer System", in LIU, LING; ÖZSU, M. TAMER (eds.), Encyclopedia of Database Systems, Springer US, pp. 2081–2082, doi:10.1007/978-0-387-39940-9_1230, ISBN 9780387399409

6. Rüdiger Schollmeier, A Definition of Peer-to-Peer Networking for the Classification of Peer-to-Peer Architectures and Applications, Proceedings of the First International Conference on Peer-to-Peer Computing, IEEE (2002).

7. Bandara, H. M. N. D; A. P. Jayasumana (2012). "Collaborative Applications over Peer-to-Peer Systems – Challenges and Solutions". Peer-to-Peer Networking and Applications. 6 (3): 257–276. arXiv:1207.0790. Bibcode:2012arXiv1207.0790D. doi:10.1007/s12083-012-0157-3.

8. Barkai, David (2001). Peer-to-peer computing: technologies for sharing and collaborating on the net. Hillsboro, OR: Intel Press. ISBN 978-0970284679. OCLC 49354877.

9. Oram, Andrew, ed. (2001). Peer-to-peer: harnessing the benefits of a disruptive technologies. Sebastopol, California: O'Reilly. ISBN 9780596001100. OCLC 123103147.

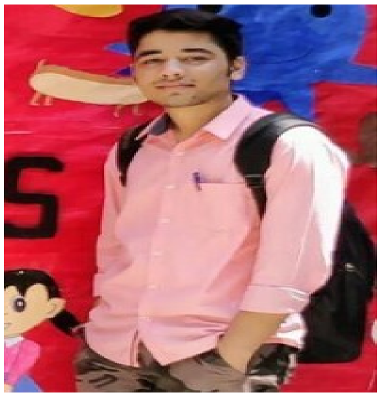
10. RFC 1, Host Software, S. Crocker, IETF Working Group (April 7, 1969)

11. Berners-Lee, Tim (August 1996). "The World Wide Web: Past, Present and Future". Retrieved 5 November 2011.

12. Steinmetz, Ralf; Wehrle, Klaus (2005). "2. What Is This "Peer-to-Peer" About?". Peer-to-Peer Systems and Applications. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg. pp. 9–16. doi:10.1007/11530657_2. ISBN 9783540291923.

13. Ahson, Syed A.; Ilyas, Mohammad, eds. (2008). SIP Handbook: Services, Technologies, and Security of Session Initiation Protocol. Taylor & Francis. p. 204. ISBN 9781420066043.

Profiles



Name: Utkarsh Mishra

Enrollment No.: 161B255

Project Group No.: 37

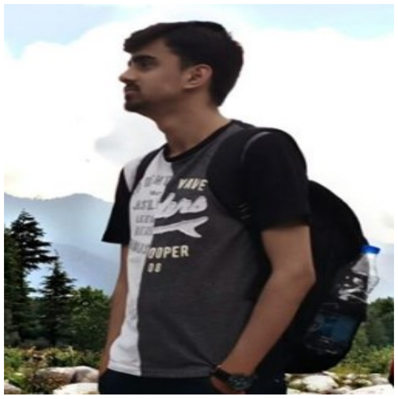
Mobile No.: 9519187890

Name: Vikrant Srivastava

Enrollment No.: 161B266

Project Group No.: 37

Mobile No.: 8090275581



Name: Yash Dwivedi

Enrollment No.: 161B277

Project Group No.:37

Mobile No.: 9755855136