**STORAGE AND ORGANIZATION OF LARGE GEOSPATIAL DATASETS ON A DISTRIBUTED BLOCKCHAIN**

***A Report submitted.***

***in partial fulfillment for the Degree of***

**B. Tech**

**in**

**Computer Science and Engineering**

***by***

**SHIVANSHU SINGH 12000120131**

**AYUSH TAH 12000119129**

*Mentored By*

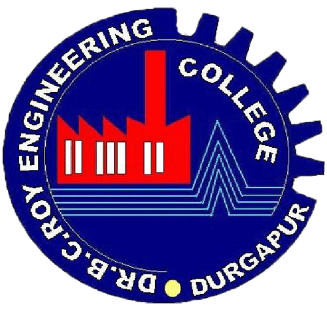
**PROF. DEBKANTA CHAKRABORTY**

pursued in

Department of Computer Science and Engineering

**DR. B. C. ROY ENGINEERING COLLEGE, DURGAPUR**

To



**DR. B. C. ROY ENGINEERING COLLEGE**

**DURGAPUR**

**NOVEMBER 2022**

**CERTIFICATE**

This is to certify that the project report entitled “Storage and Organization of large geospatial datasets on a distributed Blockchain” submitted by Ayush Tah and Shivanshu Singh to Dr. B. C. Roy Engineering College, Durgapur, in partial fulfillment for the award of the degree of B. Tech in (Computer Science and Engineering) is a bona fide record of project work carried out by him/her under my/our supervision. The contents of this report, in full or in parts, have not been submitted to any other Institution or University for the award of any degree or diploma.

……………………………

Prof. Debkanta Chakraborty

Supervisor,

……………………………

Dr. Arindam Ghosh

Head of Department,

Department of Computer Science and Engineering

Durgapur

Date

**DECLARATION**

I declare that this project report titled “Storage and Organization of large geospatial datasets on a distributed Blockchain” submitted in partial fulfillment of the degree of B. Tech in (Computer Science and Engineering) is a record of original work carried out by me under the supervision of Prof. Debkanta Chakraborty, and has not formed the basis for the award of any other degree or diploma, in this or any other Institution or University. In keeping with the ethical practice in reporting scientific information, due acknowledgements have been made wherever the findings of others have been cited.

………………………………..

Ayush Tah

12000119129

………………………………..

Shivanshu Singh

12000120131

Durgapur

DD/MM/YYYY

**ACKNOWLEDGMENTS**

We take this opportunity to thank Dr. Sanjay S. Pawar, Principal - BCREC, Prof.

Debkanta Chakraborty – Mentor, and other faculty members who helped in preparing the guidelines.

I extend my sincere thanks to one and all of BCREC family for the completion of

this document on the project report format guidelines.

Shivanshu Singh

Ayush Tah

**ABSTRACT**

With the current rapid expansion and development of global navigation systems, information and communication technology, and sensor technology and sheer number of geo satellites being launched an unprecedented amount of geospatial data is being generated every smallest of intervals. The geospatial data we are talking about consists of but not limited to individual trajectories, location-based services, remote sensing images and so on, thus there has been a persisting need for the ability of sharing and use of spatial information reliably. As any data of such magnitude and importance is concerned data privacy, integrity and security play a crucial role both in sharing and storage of such data, however, it is quite a challenge to guarantee data privacy and security in geospatial data when it is so extensively used in a myriad of technologies. In addition, geospatial data is stored in many different formats and calibrated by different standards. Any effort to compare, combine or map data first requires a significant amount of data scrubbing and reformatting thus making a claim to the integrity of data becomes difficult. Data integrity depicts the data consistency and accuracy while sharing or storing the data, which quantifies the validity and fidelity of data. To preserve privacy and integrity of geospatial data, geospatial data should be validated to satisfy the security requirements and be protected from unauthorized modifications.

Recently Blockchain has been realized as a possible solution to the problems we face with geospatial data because of the tamper-proof, traceable, trust-free, transparent, and decentralized characteristics it possesses. Blockchain achieves these de-centralized and security characteristics by integrating consensus mechanisms, asymmetric cryptographic algorithms, and distributed data storage to name a few. Hence, Blockchain presents itself as a major player in finding a solution for the problem we are trying to tackle.

**TABLE OF CONTENTS**

**DESCRIPTION PAGE NUMBER**

CERTIFICATE

DECLARATION

ACKNOWLEDGEMENTS

ABSTRACT

LIST OF FIGURES

LIST OF TABLES

ABBREVIATION/NOTATION/NOMENCLATURES

# CHAPTER 1: INTRODUCTION

## Geospatial Data

### Vector Data

### Raster Data

### File Formats

### Geospatial Data in the Industry

## Blockchain

## Why Blockchain

### Immutable

### Safer

### Resilient

### Cheaper

### Transparency and Accountability

### Control

### Trustless

### Smart Contracts

## IPFS

1.4.1 Why IPFS?

## Motivation

# PROBLEM STATEMENT

# 3 RESEARCH QUESTION

# CHAPTER 2: IMPLEMENTATION

## On-Chain Storage

## Off-Chain Storage

## Difference between On and Off-Chain storage

## The Final Decision

# CHAPTER 5: LITERATURE SURVEY, CURRENT STATUS AND GAPS

# CHAPTER 6: OBJECTIVES OF THE THESIS

# CHAPTER 7: WORK DONE TILL NOW

# CHAPTER 8: METHODOLOGY

## Setting up the environment

**LIST OF FIGURES**

**FIGURE TITLE PAGE NUMBER**

**1.1 Vector Data x**

**1.2 Raster Data xi**

**1.3 A Block in Blockchain xi**

**1.4 Genesis Block xii**

**1.5 HTTP Downloads xv**

**1.6 Fragility of Webpage xvi**

**1.7 Decentralization xvi**

**1.8 Resilient Networks xvi**

**ABBREVIATIONS/ NOTATIONS/ NOMENCLATURE**

IPFS: InterPlanetary File System

**CHAPTER 1**

**INTRODUCTION**

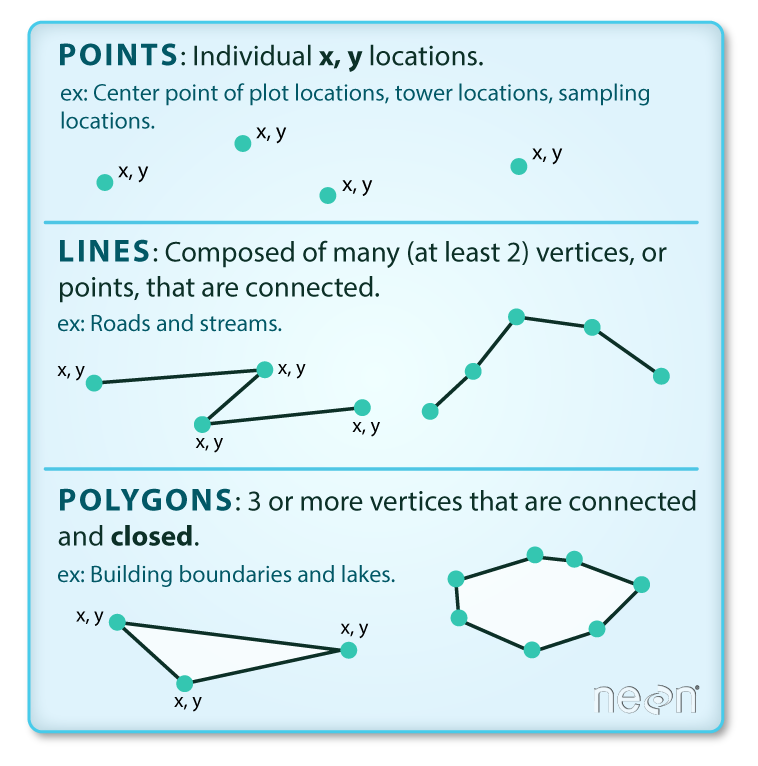
Before we dive into this, let us just see a brief introduction to all the technological terms and technologies involved in this project.

**1.1 GEOSPATIAL DATA**

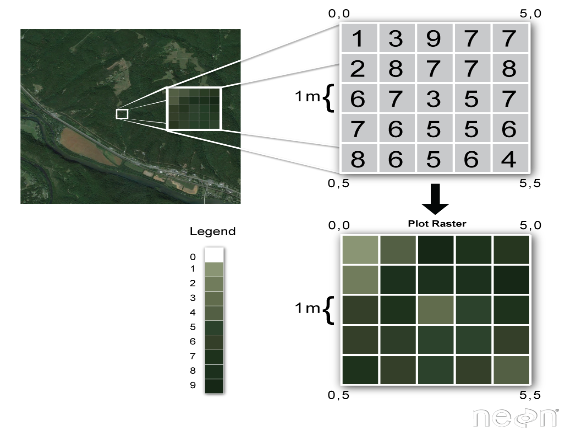
Geospatial data contains the data associated with a place on earth that contains but is not limited to latitude and longitude(location), characteristics of objects, event, or phenomenon(attributes) with temporal information(time) all of which we can use to draw a map of the said place. The location info may/may not be static or dynamic in nature.

**There are two main types of primary geospatial data models:**

* **Vector Data**: Co-ordinate based model where geographical locations are represented as points, lines, or polygon features. This works best when used to represent geographical features with defined boundaries. Each geographical feature has associated tabular/spreadsheet data(attributes).

 (Fig 1.1)

* **Raster Data:** Mostly used to represent dynamic data (e.g., temperatures) but can also be used to represent discrete data (e.g., land cover). It uses equal sized pixels like blocks arranged in a grid to represent geographic data.

(Fig 1.2)

* Minor data types that are part of the geospatial dataset include point clouds, census data, cell phone data, social media data etc.

**Some common file formats that are associated with this dataset are:**

• shapefiles (.shp)

• file geodatabases (.gdb)

• Keyhole Markup Language (.kml or .kmz)

• GeoTIFF (.tiff)

• comma-separated values (.csv)

• GeoJSON (.json)

• GPS Exchange Format (.gpx)

• LiDAR point cloud data (.las)

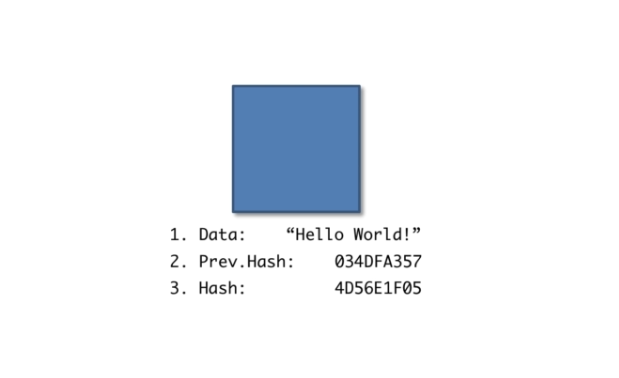
We will work with the (.tiff) format as this was the file type present in the dataset provided in the project on SIH where we took it form.

**Geospatial data applications by industry:**

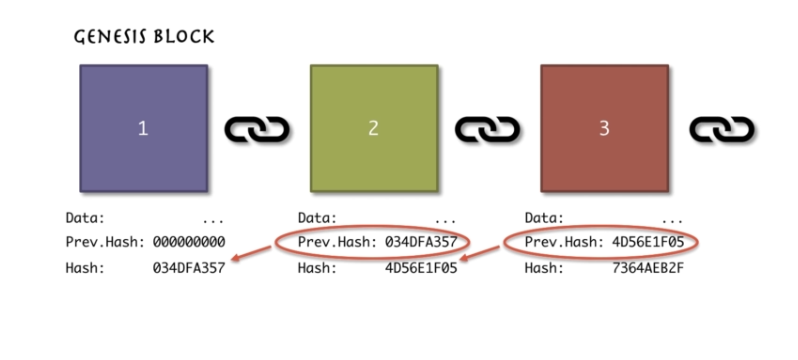
* Governments can take insights about health, disease and weather and use them to better advise the public when a natural disaster strikes, or an emergency health event occurs.
* Electric utilities providers can use data to help predict possible service disruptions and optimize maintenance and crew schedules.
* Insurers can do a more accurate job of projecting risks and warning policy holders about potential issues they may soon be facing.
* Farm and agricultural lenders can improve the methodology they use to assess credit risk scores and reduce bad loan placements.
* COVID-19 mapping - COVID-19 mapping is performed using geospatial analytic models, based on population data, live streaming video, maps, and weather.
* Vegetation management - Through user defined functions (UDFs), geospatial analytics enables those involved in vegetation management to assess water and moisture levels.
* Tornado tracking - User defined functions are also useful at helping meteorologists work with incoming data to chart the path of tornadoes that could be moving through an area.
* Wildfire mapping - Having relevant data, such as satellite imagery, census data and wind forecasts, in one platform lets incident commanders chart wildfire growth and movement.

**1.2 BLOCKCHAIN**

A blockchain is a type of distributed ledger technology (DLT) that has a continuously growing list of records, called blocks, which are linked and secured using cryptography. A block contains data, previous hash, and its own hash. It was created by a person or group using the pseudonym Satoshi Nakamoto in 2008 to serve as the public distributed ledger for bitcoin cryptocurrency transactions, based on previous work by Stuart Haber, W. Scott Stoenetta and Dave Bayer.

 (Fig 1.3)

The first block in a chain is called the "Genesis Block". It does not have any previous hash just its own. The blocks that come after the genesis block has previous hash.

(Fig 1.4)

This is the reason blocks are called to be cryptographically linked together.

## **1.3 WHY BLOCKCHAIN?**

* **Immutable:** because each block of entries (added every 1-10 minutes) carries a unique hash ‘fingerprint’ of the previous block. Hence, older blocks cannot be tampered with.
* **Safer:** because no one controls all the data (known as root privilege in existing databases). Each entry has its own pair of public and private encryption keys and only the holder of the private key can unlock the entry and transfer it to someone else.
* **Resilient:** because there is no single point of failure, there is practically nothing to attack. To compromise a blockchain, you’d have to hack each individual user one by one in order to get hold of their private encryption keys that give access to that user’s data only. Another option is to run over 50% of the nodes, which is virtually impossible and economically impractical.
* **Cheaper:** because anyone can set up a node and get paid in digital tokens (e.g., Bitcoin or Ether) for hosting a blockchain. This ensures that competition between nodes will minimize the cost of hosting it. It also saves the costs of massive security layers that otherwise apply to servers with sensitive data – this is because of the no-root-privilege security model and, with old entries being immutable, there is little need to protect them.
* **Transparency and accountability:** the fact that existing entries cannot be tampered with makes a blockchain a transparent source of truth and history for your application. The public nature of it makes it easy to hold people accountable for their activities.
* **Control:** the immutable and no-root-privilege character puts each user in full control of his/her own data using the private encryption keys. This leads to real peer-to-peer interaction without any middleman and without an administrator that can deny users access to their data.
* **Trustless:** because each user fully controls his/her own data, users can safely interact without knowing or trusting each other and without any trusted third parties.
* **Smart Contracts:** these are simply programs stored on the blockchain that run when predetermined conditions are met. They are typically used to automate the process of a transaction on the blockchain so that all participants can be certain of the outcome, without any intermediary's involvement or time loss.

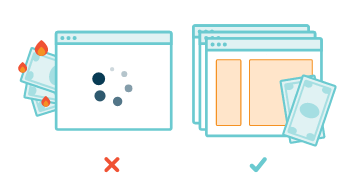
## **IPFS**

The InterPlanetary File System (IPFS) is a protocol, hypermedia, and file sharing peer-to-peer network for storing and sharing data in a distributed file system. IPFS uses content-addressing to uniquely identify each file in a global namespace connecting IPFS hosts.

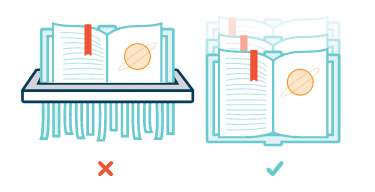
IPFS was created by Juan Benet, who later founded Protocol Labs in May 2014. IPFS was launched in an alpha version in February 2015, and by October of the same year was described by TechCrunch as "quickly spreading by word of mouth."

### **1.4.1 WHY IPFS AND HOW IT IS BETTER THAN HTTP**

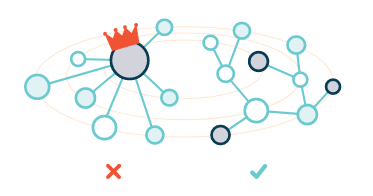
* Today's web is inefficient and expensive - HTTP downloads files from one server at a time, but peer-to-peer IPFS retrieves pieces from multiple nodes at once, enabling substantial bandwidth savings. With up to 60% savings for video, IPFS makes it possible to efficiently distribute high volumes of data without duplication.

 (Fig 1.5)

* Today's web cannot preserve humanity's history - The average lifespan of a web page is 100 days before it's gone forever. The medium of our era should not be this fragile. IPFS makes it simple to set up resilient networks for mirroring data, and thanks to content addressing, files stored using IPFS are automatically versioned.

 (Fig 1.6)

* Today's web is centralized, limiting opportunity - The Internet has turbocharged innovation by being one of the great equalizers in human history, but increasing consolidation of control threatens that progress. IPFS stays true to the original vision of an open, flat web by delivering technology to make that vision a reality.

(Fig 1.7)

* Today's web is addicted to the backbone - IPFS powers the creation of diversely resilient networks that enable persistent availability — with or without internet backbone connectivity. This means better connectivity for the developing world, during natural disasters, or just when you are on flaky coffee shop Wi-Fi.

 (Fig 1.8)

**1.5 MOTIVATION**

After a lot of time doing research on what our final year project should be. We stumbled upon this problem on the SIH platform and as we wanted to invest our time in something that had substance and was beneficial to our world. Even though we may not be able to realize the entire scope of what we set out to achieve, we will try our best to put a dent in this problem.

**CHAPTER 2**

**PROBLEM DEFINITION**

Ever-growing geospatial data needs secure and scalable storage for archiving data. Blockchain-based technologies can offer strong byzantine fault tolerance and distributed data storage and access. However, currently, there are no mature options for storing and organizing large geospatial datasets on a distributed blockchain. Our aim is to create a software toolset for storage browsing and permission-based access to geospatial data using blockchain.

**CHAPTER 3**

**RESEARCH QUESTION**

Can geospatial data storage using blockchain which provides distributed and decentralized data storage replace traditional centralized data storage?

**CHAPTER 4**

**IMPLEMENTATION**

Now let’s start with the meat and potatoes of this project where most of the work needs to be done, Blockchain. More specifically the creation of a blockchain network capable of storage of data.

There are two ways in which we can get started on this:

## **On chain storage:** These are transactions that take place on the blockchain. They are stored on the decentralized ledger and can be viewed by anyone who has a copy of the same. The blockchain network is updated every time an On-chain transaction occurs. When transaction volume is low, On-chain transactions can be quick. New network protocols and crypto assets have been designed to improve transaction speed and are becoming quite common.

## **Off chain storage:** These transactions take place outside the blockchain. A third-party, acts as a guarantee in an Off-Chain transaction. The participants of the transaction deal with each other outside of the blockchain. They do, however, rely on a third party to assist in the transaction. When all the criteria are satisfied, the transaction is performed and recorded on the blockchain. Layer 2 solutions play a key role in off-chain transactions. Some widely adopted off-chain solutions are the Lightning Network and the Liquid Network. A fast and cheap solution is the Lightning Network. The Lightning Network is built on the Bitcoin blockchain. This solution essentially opens a channel between two participants. These participants can now conduct Bitcoin transactions cheaply. This transaction data is not uploaded to the blockchain. Once the channel is closed, only then is the transaction recorded on the Bitcoin blockchain.

## **Difference Between on and Off Chain storage:**

On-chain transactions are irreversible and processed on the blockchain network. On-chain transactions take significantly longer than off-chain transactions. Since the transaction is confirmed by participants and published on the blockchain network it is highly secure.

Off-chain transactions take place without affecting the main blockchain. This removes the need to validate transactions. This also lowers transaction fees and speeds up the process. The transactions can be executed instantly. There is no lag time like on-chain transactions. The costs associated with off-chain transactions are minimal as they do not take place on the blockchain.

Unlike On-chain, Off-chain transactions are not recorded on the blockchain. If the participants no longer want to participate, they can do so without leaving a permanent record of the same. This offers anonymity to those involved in the transactions.

## **The final Decision**

The final decision in the On-chain vs Off-chain storage depends on the use case. Off-chain transactions are ideal for those looking for quick, anonymous, and cheap mechanisms. On-chain transactions provide you with security and immutability. Therefore, we would strive for on-chain storage. We might not be able to show much because of the higher cost of the tech in general but for storage of critical data like geospatial data we should be looking at On-chain on the longer run. Higher costs are justified for the perks it provides and as we have already discussed that this is for archival purposes the drawbacks are not that steep.

**CHAPTER 5**

**LITERATURE SURVEY, CURRENT STATUS AND GAPS**

There are several studies that have been conducted in this domain, though most are generally geared towards validation of data rather than the simple task of uploading and retention of data in IPFS. One study by J. R. Cedeno Jimenez, A. Folini, and M. A. Brovelli, pushed the idea of using IPFS to validate user contributed geospatial data in the IPFS network. They use the Velas blockchain infrastructure and implement a combination of a Discrete Global Grid System (DGCS) with smart contracts. Two kinds of smart contracts got produced: cell and registry smart contracts. Each DGGS partition has an individual cell smart contract that holds a list of observations present in that area. On the other hand, registry smart contracts monitor all DGGS cells that got added to the system. Presently, public authorities validate SIMILE (System for the Integrated Monitoring of Insubric Lakes and their Ecosystems) observations, which consumes time and may not be readily available. Hence, a fully functional prototype got developed to address this concern. Users can now add, oversee, and verify personal observations as well as those belonging to other users. This effort confirms the potential of creating decentralized apps for validating geographical data as a citizen science solution.

Another review paper by Jesus Rodrigo Cedeno Jimenez, Pengxiang Zhao, Ali Mansourian, and Maria Antonia Brovelli focuses more on the crowdsourcing of geospatial data and two main platforms that were developed for it, namely FOAM and D-GIS. But they found out the platforms were just conceptualized not realized/ deployed.

**CHAPTER 6**

**OBJECTIVES OF THE THESIS**

The objective of the project is to build a web app that uses IPFS to store data locally or on the IPFS network. The features we are striving towards are:

* To build a better understanding of file storage, especially geospatial data storage on a blockchain.
* To build a web app to demonstrate our project with a robust and efficiently coded backend and simple looking and easy to use GUI on the frontend.
* To develop a more secure file storage system than the existing traditional centralised data storage.

**CHAPTER 7**

**WORK DONE TILL NOW**

Currently, we are building the web app/web page for the project. We finished coding the uploading of files to our IPFS network. The GUI is currently under development. We are using JavaScript or more precisely Node.JS for the back end and HTML, CSS, and JavaScript for the front end. We are working on a lot more features that we want to add, so we work even harder towards adding new features and mechanisms later down the road.

Till the current date the back end is operational but right now only accepts 1 file at a time which we are working to be able to do as many as the user wants. There is also a basic front-end with renaming, uploading and downloading features but this will also be totally reworked to be more robust and fluent.

At present the backend is operational but only accepts one file at a time which we are working to be able to do as many as the user wants. There is also a basic front-end with renaming, uploading and downloading features but this will also be totally reworked to be more robust and with fluent ui.

The app can be made on any operating system, but we used ubuntu because of its reliability and general ease of installation of the various packages that we are going to use.

We are using WSL2 (Windows Subsystem for Linux) which is available for systems running Windows 10 ver. 1903 or above, because with its recent milestones, it is exceptionally reliable and easy to work with and as we will do most of our work in CLI we don’t need to do a standalone installation of ubuntu.

Our backbone of the project runs on IPFS (InterPlanetary File System). IPFS is a protocol, hypermedia, and file-sharing peer-to-peer network for storing and sharing data in a distributed file system. It uses content-based addressing to uniquely identify each file in a global namespace connecting IPFS hosts. IPFS can replace the location-based hypermedia server protocols HTTP and HTTPS to distribute the World Wide Web.

Every file added to IPFS is given a unique address derived from a hash of the file’s content. This address is called a Content Identifier (CID) and it combines the hash of the file and a unique identifier for the hash algorithm used into a single string. IPFS currently uses SHA-256 by default, which produces a 256-bit (32 byte) output. SHA-256 is also used by bitcoin.

Once a file is uploaded to a node, the file is broken to pieces. Based on the contents of each chunk of the file, each of them is assigned a CID, and the file is converted into a Merkle DAG (Directed Acyclic Graph). Each node of the Merkle DAG consist of chunk of the file.

So, if any chunk of the file is mutated, that will make the whole file invalid, as the hash and the DAG depends upon the content of the file. The CID of the root node of the Merkle DAG is given to the user who uploaded the file, which can be used to access and to retrieve the file anytime.

**CHAPTER 8**

**METHODOLOGY**

### **Setting up the work environment:**

We can build this app in any operating system, but we used ubuntu because of its reliability and general ease of installation of the various packages we are going to need.

We will use the WSL2 flavour of ubuntu i.e., Windows Subsystem for Linux, because with its recent milestones it is very reliable and easy to work with and as we will do most of our work in CLI we don’t need to do a standalone installation of ubuntu.

To install ubuntu:

* In the windows search bar type “Turn Windows Features on or off” and open the app.
* Look for Virtual Machine Platform and Windows Subsystem for Linux and check the boxes.
* After Restarting head to “<https://learn.microsoft.com/en-us/windows/wsl/install-manual#step-4---download-the-linux-kernel-update-package>” and download the Linux kernel package and install it
* Open a power shell window and run this command “wsl --set-default-version 2"
* Now we can head to the Microsoft store and download and install the LTS version of ubuntu from there.
* Click on the ubuntu icon that will be in the start menu after the installation and follow the on-screen instruction until the terminal is in the root directory.

Installing IPFS:

* We will use the go implementation of IPFS(Kubo), but the JavaScript implementation can also be used but it is new and has bugs so go it is.
* First run this command in the terminal: wget <https://dist.ipfs.tech/kubo/v0.18.1/kubo_v0.18.1_linux-amd64.tar.gz>
* Then extract this file with tar -xvzf kubo\_v0.18.1\_linux-amd64.tar.gz
* We will then go into the kudo folder which is one of the folders in the extracted files with: cd kubo.
* Then we will install Kubo using: sudo bash install.sh.
* That’s it IPFS is now installed and to check whether it is installed correct and generate some necessary files run: ipfs init.

Installing Node and NPM:

* Create a folder in the home folder in the ubuntu and open a Linux terminal at that location
* Run the following command to install NodeJS and NPM: sudo apt install NodeJS npm
* Then we use npm to install more packages that are required namely EJS or Embedded JavaScript, Express, FileUpload and Body Parser: npm install ejs express express-fileupload body-parser.
* There one more package that is required which is the client package of IPFS: npm install [ipfs-http-client@52.0.1](mailto:ipfs-http-client@52.0.1)
* And with this we are all set up to start coding.

**REFERENCES**

1. Briandeis Library
   1. <https://guides.library.brandeis.edu/c.php?g=990410&p=7164688>
2. IBM
   1. <https://www.ibm.com/in-en/topics/geospatial-data>
3. Vishal Chauhan
   1. <https://www.solulab.com/what-is-immutable-ledger-in-blockchain-and-its-benefits/>
4. Jonas Ellehuage, GIM International
   1. <https://www.giminternational.com/content/blog/blockchain-in-geospatial-applications-2?output=pdf>
5. Pengxiang Zhao, Jesus Rodrigo Cedeno Jimenez, Maria Antonia Brovelli, and Ali Mansourian, GIS Cente, AGILE.
   1. <https://agile-giss.copernicus.org/articles/3/71/2022/agile-giss-3-71-2022.pdf>
6. Zebpay
   1. <https://zebpay.com/in/blog/on-chain-vs-off-chain>