**Budget Tracking using Blockchain**

**1. INTRODUCTION:**

Now-a-days to ease user work many online applications are available which can track user daily activity and in all human the biggest activity is to monitor their expenditure and available budget limit so they can be in limit of expenditure to avoid unnecessary debt.

All existing applications are based on single centralized servers and this server’s data can be easily tamper and there is no way to track such data tamper, sometime hackers can hack this server to crash and in such scenarios services will not be available. To overcome from above issues many applications are migrating to Decentralized Blockchain storage where data will be stored at multiple nodes in a decentralized way and if one node down then services can be access from any other working nodes.

Blockchain stored each record as block or transaction and associate each block with unique hash code, while storing new record Blockchain will verify hash code of all previous blocks and if data not tamper then all blocks will generate same hash code and verification get successful and if block data alter then it will result into incorrect hash code and data alteration will get detected. Hash code verification of Blockchain make it tamper proof and impossible for data alteration.

**1.1 Objective of the project:**

This is the century of technology and digitization. Existing problems require modern solutions. The problems that have existed through decades or even centuries can now be solved through current technologies. Blockchain is one of the technologies that is going to have an extensive and comprehensive change in people's lives. Blockchain is an emerging technology and its fields of application are being explored. The domain of field chosen in this paper for its application is the public sector. Pioneering research done in the field of public services working shows that bureaucracy, corruption can be reduced as well as on the other hand, efficiency and level of trust in public record keeping can be increased using blockchain technology. Blockchain can be used in government fund management. Whole government budgeting, funding, auditing and financial records can be brought on the blockchain network. Today, in general, the allocated funds are not traceable, there is no concrete knowledge as to how, where and what percentage of funds are being utilized and a large part of these expenditures are never shown in records due to corruption. To culminate this problem, a Blockchain system has been proposed to provide transparency and proficiency.

**2. LITERATURE SURVEY:**

**Online Income and Expense Tracker**

the web application “Expense Tracker” is developed to manage the daily expenses in a more efficient and manageable way. By using this application we can reduce the manual calculations of the daily expenses and keep track of the expenditure. In this application, user can provide his income to calculate his total expenses per day and these results will be stored for each user. The application has the provision to predict the income and expense for the manager using data mining. In this application, there are 3 logins such as admin, manager and staff. Admin has the privilege to add, edit, delete manager, add, edit, delete staff, and to get all custom reports. For Manager, the privileges are to add type of expense, verify expense, add type of income, verify income and generate reports. For staff, the privileges are to add and edit expense, income and calculations, and send for verifications.

**Personalized Expense Managing Assistant Using Android**

Mobile applications stood top among usability and user convenience. Many applications are available in the market to manage personal and group expenses. Not many applications provides a comprehensive view of both use cases. In this project, we develop a mobile application that keeps track of user personal expenses, his/her personal contribution towards group expenses; maintain monthly incomes, recurring and adhoc payments. It provides information of "who owes who and by how much". The proposed application would eliminate sticky note, spreadsheet and ledger that cause confusions, data inconsistency problems while recording and splitting of expenses. With our application user can manage his expenses more effectively. This application will not only helps users to manage their expenses but also help marketing executives to plan marketing according to the needs of users.

**Automated Payroll Using GPS Tracking and Image Capture**

This system is a combination of web as well as android application where the user will be using the android application and admin as well as HR will work with web application. This application is meant for field work users. The user will have this application in his android phone, when the user will login to the system his image will be captured and his GPS location will be send to the admin where admin will view image and GPS location in web application. After Login, GPS location of the employee will be tracked automatically by the system and send to the admin after every 5 minutes .When user logout the system again the image will be captured as well as GPS location will be send to the admin. In order to keep track of the attendance as well as payroll of the field work people, this system plays a major role. The role of the admin is to add new employee by entering his personal details and admin will provide the employee with identity number and password to the user so that he can access the application in his android phone. Admin can view the GPS location of the employee by entering Employee Identity Number as well as Date. Admin can check the salary of the particular employee by entering date and employee ID.

**eExpense: A smart approach to track everyday expense.**

This application extracts the textual information from the receipts and amount and description for further processing. It also monitors user's income by tracking the received SMS's from the user's saving accounts. By calculating income and expense it produces the user's balance in monthly and yearly basis.

**3. SYSTEM ANALYSIS**

**3.1 Existing System**

Now-a-days to ease user work many online applications are available which can track user daily activity and in all human the biggest activity is to monitor their expenditure and available budget limit so they can be in limit of expenditure to avoid unnecessary debt.All existing applications are based on single centralized servers and this server’s data can be easily tamper and there is no way to track such data tamper, sometime hackers can hack this server to crash and in such scenarios services will not be available.

**Disadvantages of Existing System:**

* Less security

**3.2 Proposed System**

To overcome from above issues many applications are migrating to Decentralized Blockchain storage where data will be stored at multiple nodes in a decentralized way and if one node down then services can be access from any other working nodes.Blockchain stored each record as block or transaction and associate each block with unique hash code, while storing new record Blockchain will verify hash code of all previous blocks and if data not tamper then all blocks will generate same hash code and verification get successful and if block data alter then it will result into incorrect hash code and data alteration will get detected. Hash code verification of Blockchain make it tamper proof and impossible for data alteration.

**Advantages of Proposed System:**

* Accuracy more.
* Provides Highly Efficient Security

**Modules Information:**

1. User signup: Using this module we will signup
2. User Login: using this module user will login
3. Add budget details: using this module we will add the budget details
4. Track budget: Using this module we will track the budget

**FUNCTIONAL REQUIREMENTS:**

**SOFTWARE REQIREMENTS:**

**System Attributes:**

1. filename
2. X, Y
3. dataset
4. classifier
5. X\_train, X\_test, y\_train, y\_test

**Data base Requirements:**

No need

**Prototype:**

python 3.7.0 or 3.7.4

opencv-python==4.5.1.48

keras==2.3.1

tensorflow==1.14.0

protobuf==3.16.0

h5py==2.10.0

sklearn-extensions==0.0.2

scikit-learn==0.22.2.post1

Numpy

Pandas

**NON-FUNCTIONAL REQUIREMENT:**

**Usability:**  Usability is a quality attribute that assesses how easy user interfaces are to use. The word "usability" also refers to methods for improving ease-of-use during the design process.(how it was handle entire project easy)

**Security:** the quality or state of being secure: such as. a : freedom from danger : safety. b : freedom from fear or anxiety. c : freedom from the prospect of being laid off job security.

**Readability:** Readability is the ease with which a reader can understand a written text.

**Performance**: the execution of an action. : something accomplished : deed, feat. : the fulfillment of a claim, promise, or request : implementation. 3. : the action of representing a character in a play.

**Availability**: the quality or state of being available trying to improve the availability of affordable housing. 2 : an available person or thing.

**Scalability**: Scalability is the measure of a system's ability to increase or decrease in performance and cost in response to changes in application and system processing demands.

**3.3. PROCESS MODEL USED WITH JUSTIFICATION**

**SDLC (Umbrella Model):**

**Umbrella Activity**

**Umbrella Activity**

**Umbrella Activity**

1. Feasibility Study
2. TEAM FORMATION
3. Project Specification PREPARATION

Business Requirement Documentation

ANALYSIS & DESIGN

CODE

UNIT TEST

DOCUMENT CONTROL

ASSESSMENT

TRAINING

INTEGRATION & SYSTEM TESTING

DELIVERY/INSTALLATION

ACCEPTANCE TEST

Requirements Gathering

SDLC is nothing but Software Development Life Cycle. It is a standard which is used by software industry to develop good software.

**Stages in SDLC:**

* Requirement Gathering
* Analysis
* Designing
* Coding
* Testing
* Maintenance

**Requirements Gathering stage:**

The requirements gathering process takes as its input the goals identified in the high-level requirements section of the project plan. Each goal will be refined into a set of one or more requirements. These requirements define the major functions of the intended application, define operational data areas and reference data areas, and define the initial data entities. Major functions include critical processes to be managed, as well as mission critical inputs, outputs and reports. A user class hierarchy is developed and associated with these major functions, data areas, and data entities. Each of these definitions is termed a Requirement. Requirements are identified by unique requirement identifiers and, at minimum, contain a requirement title and textual description.



These requirements are fully described in the primary deliverables for this stage: the Requirements Document and the Requirements Traceability Matrix (RTM). The requirements document contains complete descriptions of each requirement, including diagrams and references to external documents as necessary. Note that detailed listings of database tables and fields are *not* included in the requirements document.

The title of each requirement is also placed into the first version of the RTM, along with the title of each goal from the project plan. The purpose of the RTM is to show that the product components developed during each stage of the software development lifecycle are formally connected to the components developed in prior stages.

In the requirements stage, the RTM consists of a list of high-level requirements, or goals, by title, with a listing of associated requirements for each goal, listed by requirement title. In this hierarchical listing, the RTM shows that each requirement developed during this stage is formally linked to a specific product goal. In this format, each requirement can be traced to a specific product goal, hence the term requirements traceability.

The outputs of the requirements definition stage include the requirements document, the RTM, and an updated project plan.

* Feasibility study is all about identification of problems in a project.
* No. of staff required to handle a project is represented as Team Formation, in this case only modules are individual tasks will be assigned to employees who are working for that project.
* Project Specifications are all about representing of various possible inputs submitting to the server and corresponding outputs along with reports maintained by administrator.

**Analysis Stage:**

The planning stage establishes a bird's eye view of the intended software product, and uses this to establish the basic project structure, evaluate feasibility and risks associated with the project, and describe appropriate management and technical approaches.



The most critical section of the project plan is a listing of high-level product requirements, also referred to as goals. All of the software product requirements to be developed during the requirements definition stage flow from one or more of these goals. The minimum information for each goal consists of a title and textual description, although additional information and references to external documents may be included. The outputs of the project planning stage are the configuration management plan, the quality assurance plan, and the project plan and schedule, with a detailed listing of scheduled activities for the upcoming Requirements stage, and high level estimates of effort for the out stages.

**Designing Stage:**

The design stage takes as its initial input the requirements identified in the approved requirements document. For each requirement, a set of one or more design elements will be produced as a result of interviews, workshops, and/or prototype efforts. Design elements describe the desired software features in detail, and generally include functional hierarchy diagrams, screen layout diagrams, tables of business rules, business process diagrams, pseudo code, and a complete entity-relationship diagram with a full data dictionary. These design elements are intended to describe the software in sufficient detail that skilled programmers may develop the software with minimal additional input.

  
When the design document is finalized and accepted, the RTM is updated to show that each design element is formally associated with a specific requirement. The outputs of the design stage are the design document, an updated RTM, and an updated project plan.

**Development (Coding) Stage:**

The development stage takes as its primary input the design elements described in the approved design document. For each design element, a set of one or more software artifacts will be produced. Software artifacts include but are not limited to menus, dialogs, and data management forms, data reporting formats, and specialized procedures and functions. Appropriate test cases will be developed for each set of functionally related software artifacts, and an online help system will be developed to guide users in their interactions with the software.



The RTM will be updated to show that each developed artifact is linked to a specific design element, and that each developed artifact has one or more corresponding test case items. At this point, the RTM is in its final configuration. The outputs of the development stage include a fully functional set of software that satisfies the requirements and design elements previously documented, an online help system that describes the operation of the software, an implementation map that identifies the primary code entry points for all major system functions, a test plan that describes the test cases to be used to validate the correctness and completeness of the software, an updated RTM, and an updated project plan.

**Integration & Test Stage:**

During the integration and test stage, the software artifacts, online help, and test data are migrated from the development environment to a separate test environment. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite confirms a robust and complete migration capability. During this stage, reference data is finalized for production use and production users are identified and linked to their appropriate roles. The final reference data (or links to reference data source files) and production user list are compiled into the Production Initiation Plan.



The outputs of the integration and test stage include an integrated set of software, an online help system, an implementation map, a production initiation plan that describes reference data and production users, an acceptance plan which contains the final suite of test cases, and an updated project plan.

* **Installation & Acceptance Test:**

During the installation and acceptance stage, the software artifacts, online help, and initial production data are loa ded onto the production server. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite is a prerequisite to acceptance of the software by the customer.

After customer personnel have verified that the initial production data load is correct and the test suite has been executed with satisfactory results, the customer formally accepts the delivery of the software.



The primary outputs of the installation and acceptance stage include a production application, a completed acceptance test suite, and a memorandum of customer acceptance of the software. Finally, the PDR enters the last of the actual labor data into the project schedule and locks the project as a permanent project record. At this point the PDR "locks" the project by archiving all software items, the implementation map, the source code, and the documentation for future reference.

**Maintenance:**

Outer rectangle represents maintenance of a project, Maintenance team will start with requirement study, understanding of documentation later employees will be assigned work and they will undergo training on that particular assigned category. For this life cycle there is no end, it will be continued so on like an umbrella (no ending point to umbrella sticks).

**3.4. Software Requirement Specification**

**3.4.1. Overall Description**

A Software Requirements Specification (SRS) – a [requirements specification](http://en.wikipedia.org/wiki/Requirements_specification) for a [software system](http://en.wikipedia.org/wiki/Software_system) is a complete description of the behavior of a system to be developed. It includes a set of [use cases](http://en.wikipedia.org/wiki/Use_case) that describe all the interactions the users will have with the software. In addition to use cases, the SRS also contains non-functional requirements. [Nonfunctional requirements](http://en.wikipedia.org/wiki/Non-functional_requirements) are requirements which impose constraints on the design or implementation (such as [performance engineering](http://en.wikipedia.org/wiki/Performance_engineering) requirements, [quality](http://en.wikipedia.org/wiki/Quality_%28business%29) standards, or design constraints).

System requirements specification: A structured collection of information that embodies the requirements of a system. A [business analyst](http://en.wikipedia.org/wiki/Business_analyst), sometimes titled [system analyst](http://en.wikipedia.org/wiki/System_analyst), is responsible for analyzing the business needs of their clients and stakeholders to help identify business problems and propose solutions. Within the [systems development lifecycle](http://en.wikipedia.org/wiki/Systems_development_life_cycle) domain, the BA typically performs a liaison function between the business side of an enterprise and the information technology department or external service providers. Projects are subject to three sorts of requirements:

* [Business requirements](http://en.wikipedia.org/wiki/Business_requirements) describe in business terms what must be delivered or accomplished to provide value.
* Product requirements describe properties of a system or product (which could be one of several ways to accomplish a set of business requirements.)
* Process requirements describe activities performed by the developing organization. For instance, process requirements could specify .Preliminary investigation examine project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All system is feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:
* **ECONOMIC FEASIBILITY**

A system can be developed technically and that will be used if installed must still be a good investment for the organization. In the economical feasibility, the development cost in creating the system is evaluated against the ultimate benefit derived from the new systems. Financial benefits must equal or exceed the costs. The system is economically feasible. It does not require any addition hardware or software. Since the interface for this system is developed using the existing resources and technologies available at NIC, There is nominal expenditure and economical feasibility for certain.

* **Operational Feasibility**

Proposed projects are beneficial only if they can be turned out into information system. That will meet the organization’s operating requirements. Operational feasibility aspects of the project are to be taken as an important part of the project implementation. This system is targeted to be in accordance with the above-mentioned issues. Beforehand, the management issues and user requirements have been taken into consideration. So there is no question of resistance from the users that can undermine the possible application benefits. The well-planned design would ensure the optimal utilization of the computer resources and would help in the improvement of performance status.

* **TECHNICAL FEASIBILITY**

Earlier no system existed to cater to the needs of ‘Secure Infrastructure Implementation System’. The current system developed is technically feasible. It is a web based user interface for audit workflow at NIC-CSD. Thus it provides an easy access to .the users. The database’s purpose is to create, establish and maintain a workflow among various entities in order to facilitate all concerned users in their various capacities or roles. Permission to the users would be granted based on the roles specified. Therefore, it provides the technical guarantee of accuracy, reliability and security.

**3.4.2. External Interface Requirements**

**User Interface**

The user interface of this system is a user friendly python Graphical User Interface.

**Hardware Interfaces**

The interaction between the user and the console is achieved through python capabilities.

**Software Interfaces**

The required software is python.

**SYSTEM REQUIREMENT:**

**HARDWARE REQUIREMENTS:**

# Processor - Intel i3(min)

* Speed - 1.1 GHz
* RAM - 4GB(min)
* Hard Disk - 500 GB
* Key Board - Standard Windows Keyboard
* Mouse - Two or Three Button Mouse
* Monitor - SVGA

**SOFTWARE REQUIREMENTS:**

* Operating System - Windows10(min)
* Programming Language - Python

**4. SYSTEM DESIGN**

**CLASS DIAGRAM:**

The class diagram is the main building block of object oriented modeling. It is used both for general conceptual modeling of the systematic of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling. The classes in a class diagram represent both the main objects, interactions in the application and the classes to be programmed. In the diagram, classes are represented with boxes which contain three parts:

* The upper part holds the name of the class
* The middle part contains the attributes of the class
* The bottom part gives the methods or operations the class can take or undertake



**USECASE DIAGRAM:**

A **use case diagram** at its simplest is a representation of a user's interaction with the system and depicting the specifications of a use case. A use case diagram can portray the different types of users of a system and the various ways that they interact with the system. This type of diagram is typically used in conjunction with the textual use case and will often be accompanied by other types of diagrams as we



**SEQUENCE DIAGRAM**

A **sequence diagram** is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called **event diagrams**, **event scenarios**, and timing diagrams.



**COLLABORATION DIAGRAM:**

A collaboration diagram describes interactions among objects in terms of sequenced messages. Collaboration diagrams represent a combination of information taken from class, sequence, and use case diagrams describing both the static structure and dynamic behaviour of a system.

****

**COMPONENT DIAGRAM:**

In the Unified Modelling Language, a component diagram depicts how components are wired together to form larger components and or software systems. They are used to illustrate the structure of arbitrarily complex systems.

Components are wired together by using an assembly connector to connect the required interface of one component with the provided interface of another component. This illustrates the service consumer - service provider relationship between the two components.



**DEPLOYMENT DIAGRAM:**

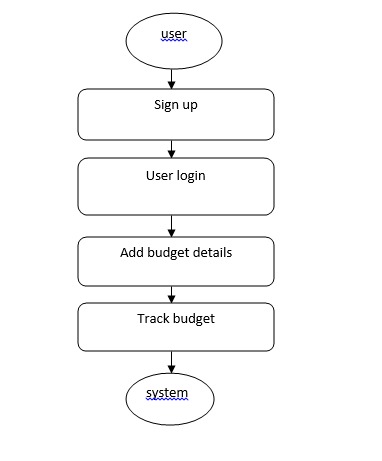
A **deployment diagram** in the Unified Modeling Language models the *physical* deployment of artifacts on nodes. To describe a web site, for example, a deployment diagram would show what hardware components ("nodes") exist (e.g., a web server, an application server, and a database server), what software components ("artifacts") run on each node (e.g., web application, database), and how the different pieces are connected (e.g. JDBC, REST, RMI).

The nodes appear as boxes, and the artifacts allocated to each node appear as rectangles within the boxes. Nodes may have sub nodes, which appear as nested boxes. A single node in a deployment diagram may conceptually represent multiple physical nodes, such as a cluster of database servers.

****

**ACTIVITY DIAGRAM:**

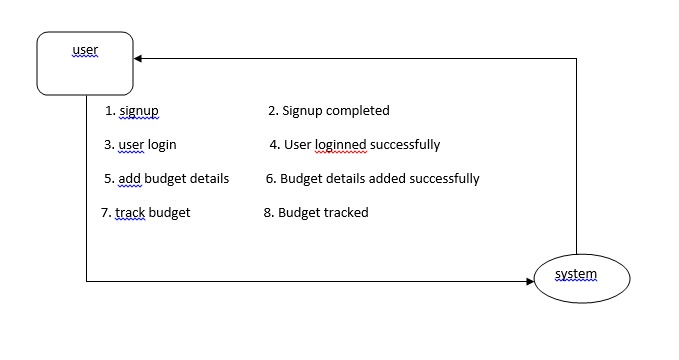
Activity diagram is another important diagram in UML to describe dynamic aspects of the system. It is basically a flow chart to represent the flow form one activity to another activity. The activity can be described as an operation of the system. So the control flow is drawn from one operation to another. This flow can be sequential, branched or concurrent



**Data flow :**

Data flow diagrams illustrate how data is processed by a system in terms of inputs and outputs. Data flow diagrams can be used to provide a clear representation of any business function. The technique starts with an overall picture of the business and continues by analyzing each of the functional areas of interest. This analysis can be carried out in precisely the level of detail required. The technique exploits a method called top-down expansion to conduct the analysis in a targeted way.

As the name suggests, Data Flow Diagram (DFD) is an illustration that explicates the passage of information in a process. A DFD can be easily drawn using simple symbols. Additionally, complicated processes can be easily automated by creating DFDs using easy-to-use, free downloadable diagramming tools. A DFD is a model for constructing and analyzing information processes. DFD illustrates the flow of information in a process depending upon the inputs and outputs. A DFD can also be referred to as a Process Model. A DFD demonstrates business or technical process with the support of the outside data saved, plus the data flowing from the process to another and the end results.



**5. IMPLEMETATION**

**5.1 Python**

Python is a general-purpose language. It has wide range of applications from Web development (like: Django and Bottle), scientific and mathematical computing (Orange, SymPy, NumPy) to desktop graphical user Interfaces (Pygame, Panda3D). The syntax of the language is clean and length of the code is relatively short. It's fun to work in Python because it allows you to think about the problem rather than focusing on the syntax.

**History of Python:**

Python is a fairly old language created by Guido Van Rossum. The design began in the late 1980s and was first released in February 1991.

**Why Python was created?**

In late 1980s, Guido Van Rossum was working on the Amoeba distributed operating system group. He wanted to use an interpreted language like ABC (ABC has simple easy-to-understand syntax) that could access the Amoeba system calls. So, he decided to create a language that was extensible. This led to design of a new language which was later named Python.

**Why the name Python?**

No. It wasn't named after a dangerous snake. Rossum was fan of a comedy series from late seventies. The name "Python" was adopted from the same series "Monty Python's Flying Circus".

**Features of Python:**

**A simple language which is easier to learn**

Python has a very simple and elegant syntax. It's much easier to read and write Python programs compared to other languages like: C++, Java, C#. Python makes programming fun and allows you to focus on the solution rather than syntax.

If you are a newbie, it's a great choice to start your journey with Python.

**Free and open-source**

You can freely use and distribute Python, even for commercial use. Not only can you use and distribute software’s written in it, you can even make changes to the Python's source code.

Python has a large community constantly improving it in each iteration.

**Portability**

You can move Python programs from one platform to another, and run it without any changes.

It runs seamlessly on almost all platforms including Windows, Mac OS X and Linux.

**Extensible and Embeddable**

Suppose an application requires high performance. You can easily combine pieces of C/C++ or other languages with Python code.

This will give your application high performance as well as scripting capabilities which other languages may not provide out of the box.

**A high-level, interpreted language**

Unlike C/C++, you don't have to worry about daunting tasks like memory management, garbage collection and so on.

Likewise, when you run Python code, it automatically converts your code to the language your computer understands. You don't need to worry about any lower-level operations.

**Large standard libraries to solve common tasks**

Python has a number of standard libraries which makes life of a programmer much easier since you don't have to write all the code yourself. For example: Need to connect MySQL database on a Web server? You can use MySQLdb library using import MySQLdb .

Standard libraries in Python are well tested and used by hundreds of people. So you can be sure that it won't break your application.

**Object-oriented**

Everything in Python is an object. Object oriented programming (OOP) helps you solve a complex problem intuitively.

With OOP, you are able to divide these complex problems into smaller sets by creating objects.

**Applications of Python:**

**1. Simple Elegant Syntax**

Programming in Python is fun. It's easier to understand and write Python code. Why? The syntax feels natural. Take this source code for an example:

a = 2

b = 3

sum = a + b

print(sum)

**2. Not overly strict**

You don't need to define the type of a variable in Python. Also, it's not necessary to add semicolon at the end of the statement.

Python enforces you to follow good practices (like proper indentation). These small things can make learning much easier for beginners.

**3. Expressiveness of the language**

Python allows you to write programs having greater functionality with fewer lines of code. Here's a link to the source code of Tic-tac-toe game with a graphical interface and a smart computer opponent in less than 500 lines of code. This is just an example. You will be amazed how much you can do with Python once you learn the basics.

**4. Great Community and Support**

Python has a large supporting community. There are numerous active forums online which can be handy if you are stuck.

**5.2 Sample Code:**

from django.shortcuts import render

from django.template import RequestContext

from django.contrib import messages

from django.http import HttpResponse

from django.core.files.storage import FileSystemStorage

from datetime import date

import os

import json

from web3 import Web3, HTTPProvider

global details, username, limit

details=''

global contract

def readDetails(contract\_type):

global details

details = ""

print(contract\_type+"======================")

blockchain\_address = 'http://127.0.0.1:9545' #Blokchain connection IP

web3 = Web3(HTTPProvider(blockchain\_address))

web3.eth.defaultAccount = web3.eth.accounts[0]

compiled\_contract\_path = 'BudgetContract.json' #BudgetContract contract code

deployed\_contract\_address = '0xD0213d792A052691d2e26E4Ca993238D1f5126Bd' #hash address to access Budget contract

with open(compiled\_contract\_path) as file:

contract\_json = json.load(file) # load contract info as JSON

contract\_abi = contract\_json['abi'] # fetch contract's abi - necessary to call its functions

file.close()

contract = web3.eth.contract(address=deployed\_contract\_address, abi=contract\_abi) #now calling contract to access data

if contract\_type == 'users':

details = contract.functions.getUsers().call()

if contract\_type == 'budget':

details = contract.functions.getBudgetDetails().call()

print(details)

def saveDataBlockChain(currentData, contract\_type):

global details

global contract

details = ""

blockchain\_address = 'http://127.0.0.1:9545'

web3 = Web3(HTTPProvider(blockchain\_address))

web3.eth.defaultAccount = web3.eth.accounts[0]

compiled\_contract\_path = 'BudgetContract.json' #BudgetContract contract file

deployed\_contract\_address = '0xD0213d792A052691d2e26E4Ca993238D1f5126Bd' #contract address

with open(compiled\_contract\_path) as file:

contract\_json = json.load(file) # load contract info as JSON

contract\_abi = contract\_json['abi'] # fetch contract's abi - necessary to call its functions

file.close()

contract = web3.eth.contract(address=deployed\_contract\_address, abi=contract\_abi)

readDetails(contract\_type)

if contract\_type == 'users':

details+=currentData

msg = contract.functions.addUsers(details).transact()

tx\_receipt = web3.eth.waitForTransactionReceipt(msg)

if contract\_type == 'budget':

details+=currentData

msg = contract.functions.addBudgetDetails(details).transact()

tx\_receipt = web3.eth.waitForTransactionReceipt(msg)

def index(request):

if request.method == 'GET':

return render(request, 'index.html', {})

def Login(request):

if request.method == 'GET':

return render(request, 'Login.html', {})

def Register(request):

if request.method == 'GET':

return render(request, 'Register.html', {})

def AddBudget(request):

if request.method == 'GET':

return render(request, 'AddBudget.html', {})

def AddBudgetAction(request):

if request.method == 'POST':

global username

name = request.POST.get('t1', False)

amount = request.POST.get('t2', False)

desc = request.POST.get('t3', False)

today = date.today()

data = username+"#"+name+"#"+amount+"#"+desc+"#"+str(today)+"\n"

saveDataBlockChain(data,"budget")

context= {'data': 'Budget details saved in Blockchain'}

return render(request, 'AddBudget.html', context)

def Signup(request):

if request.method == 'POST':

username = request.POST.get('username', False)

password = request.POST.get('password', False)

contact = request.POST.get('contact', False)

email = request.POST.get('email', False)

address = request.POST.get('address', False)

limit = request.POST.get('limit', False)

record = 'none'

readDetails("users")

rows = details.split("\n")

for i in range(len(rows)-1):

arr = rows[i].split("#")

if arr[1] == username:

record = "exists"

break

if record == 'none':

data = username+"#"+password+"#"+contact+"#"+email+"#"+address+"#"+limit+"\n"

saveDataBlockChain(data,"users")

context= {'data':'Signup process completed and record saved in Blockchain'}

return render(request, 'Register.html', context)

else:

context= {'data':username+'Username already exists'}

return render(request, 'Register.html', context)

def UserLogin(request):

if request.method == 'POST':

global username, limit

limit = 0

username = request.POST.get('username', False)

password = request.POST.get('password', False)

status = 'none'

readDetails("users")

rows = details.split("\n")

for i in range(len(rows)-1):

arr = rows[i].split("#")

if arr[0] == username and arr[1] == password:

status = 'success'

limit = arr[5]

break

if status == 'success':

context= {'data':"Welcome "+username}

return render(request, 'UserScreen.html', context)

else:

context= {'data':"Invalid Login"}

return render(request, 'Login.html', context)

def getExpenditure(exp, dd):

amount = 0

for i in range(len(exp)):

amt = exp[i]

value = amt[0]

if value == dd:

amount = str(amt[1])

break

return amount

def TrackBudget(request):

if request.method == 'GET':

global username, limit

output = '<table border=1 align=center>'

output+='<tr><th><font size=3 color=black>Username</font></th>'

output+='<th><font size=3 color=black>Budget Name</font></th>'

output+='<th><font size=3 color=black>Amount</font></th>'

output+='<th><font size=3 color=black>Description</font></th>'

output+='<th><font size=3 color=black>Budget Feed Date</font></th></tr>'

readDetails("budget")

rows = details.split("\n")

dates = []

expenditure = []

for i in range(len(rows)-1):

arr = rows[i].split("#")

if arr[0] == username:

dd = arr[4].split("-")

temp = dd[0]+"-"+dd[1]

if temp not in dates:

dates.append(temp)

for i in range(len(dates)):

amount = 0

for j in range(len(rows)-1):

arr = rows[j].split("#")

if arr[0] == username:

dd = arr[4].split("-")

temp = dd[0]+"-"+dd[1]

if dates[i] == temp:

amount = amount + float(arr[2])

expenditure.append([dates[i], amount])

dup = []

for i in range(len(rows)-1):

arr = rows[i].split("#")

if arr[0] == username:

dd = arr[4].split("-")

temp = dd[0]+"-"+dd[1]

if temp not in dup:

exp\_value = getExpenditure(expenditure, temp)

output+='<tr><td><font size=3 color=red>Max Limit = '+str(limit)+'</font></td>'

output+='<td><font size=3 color=red>Expenditure for '+temp+' = '+str(exp\_value)+'</font></td></tr>'

dup.append(temp)

output+='<tr><td><font size=3 color=black>'+arr[0]+'</font></td>'

output+='<td><font size=3 color=black>'+arr[1]+'</font></td>'

output+='<td><font size=3 color=black>'+arr[2]+'</font></td>'

output+='<td><font size=3 color=black>'+arr[3]+'</font></td>'

output+='<td><font size=3 color=black>'+arr[4]+'</font></td></tr>'

context= {'data': output}

return render(request, 'UserScreen.html', context)

**6. TESTING:**

**Implementation and Testing:**

Implementation is one of the most important tasks in project is the phase in which one has to be cautions because all the efforts undertaken during the project will be very interactive. Implementation is the most crucial stage in achieving successful system and giving the users confidence that the new system is workable and effective. Each program is tested individually at the time of development using the sample data and has verified that these programs link together in the way specified in the program specification. The computer system and its environment are tested to the satisfaction of the user.

## Implementation

## The implementation phase is less creative than system design. It is primarily concerned with user training, and file conversion. The system may be requiring extensive user training. The initial parameters of the system should be modifies as a result of a programming. A simple operating procedure is provided so that the user can understand the different functions clearly and quickly. The different reports can be obtained either on the inkjet or dot matrix printer, which is available at the disposal of the user. The proposed system is very easy to implement. In general implementation is used to mean the process of converting a new or revised system design into an operational one.

## Testing

Testing is the process where the test data is prepared and is used for testing the modules individually and later the validation given for the fields. Then the system testing takes place which makes sure that all components of the system property functions as a unit. The test data should be chosen such that it passed through all possible condition. Actually testing is the state of implementation which aimed at ensuring that the system works accurately and efficiently before the actual operation commence. The following is the description of the testing strategies, which were carried out during the testing period.

### System Testing

Testing has become an integral part of any system or project especially in the field of information technology. The importance of testing is a method of justifying, if one is ready to move further, be it to be check if one is capable to with stand the rigors of a particular situation cannot be underplayed and that is why testing before development is so critical. When the software is developed before it is given to user to use the software must be tested whether it is solving the purpose for which it is developed. This testing involves various types through which one can ensure the software is reliable. The program was tested logically and pattern of execution of the program for a set of data are repeated. Thus the code was exhaustively checked for all possible correct data and the outcomes were also checked.

**Module Testing**

To locate errors, each module is tested individually. This enables us to detect error and correct it without affecting any other modules. Whenever the program is not satisfying the required function, it must be corrected to get the required result. Thus all the modules are individually tested from bottom up starting with the smallest and lowest modules and proceeding to the next level. Each module in the system is tested separately. For example the job classification module is tested separately. This module is tested with different job and its approximate execution time and the result of the test is compared with the results that are prepared manually. The comparison shows that the results proposed system works efficiently than the existing system. Each module in the system is tested separately. In this system the resource classification and job scheduling modules are tested separately and their corresponding results are obtained which reduces the process waiting time.

**Integration Testing**

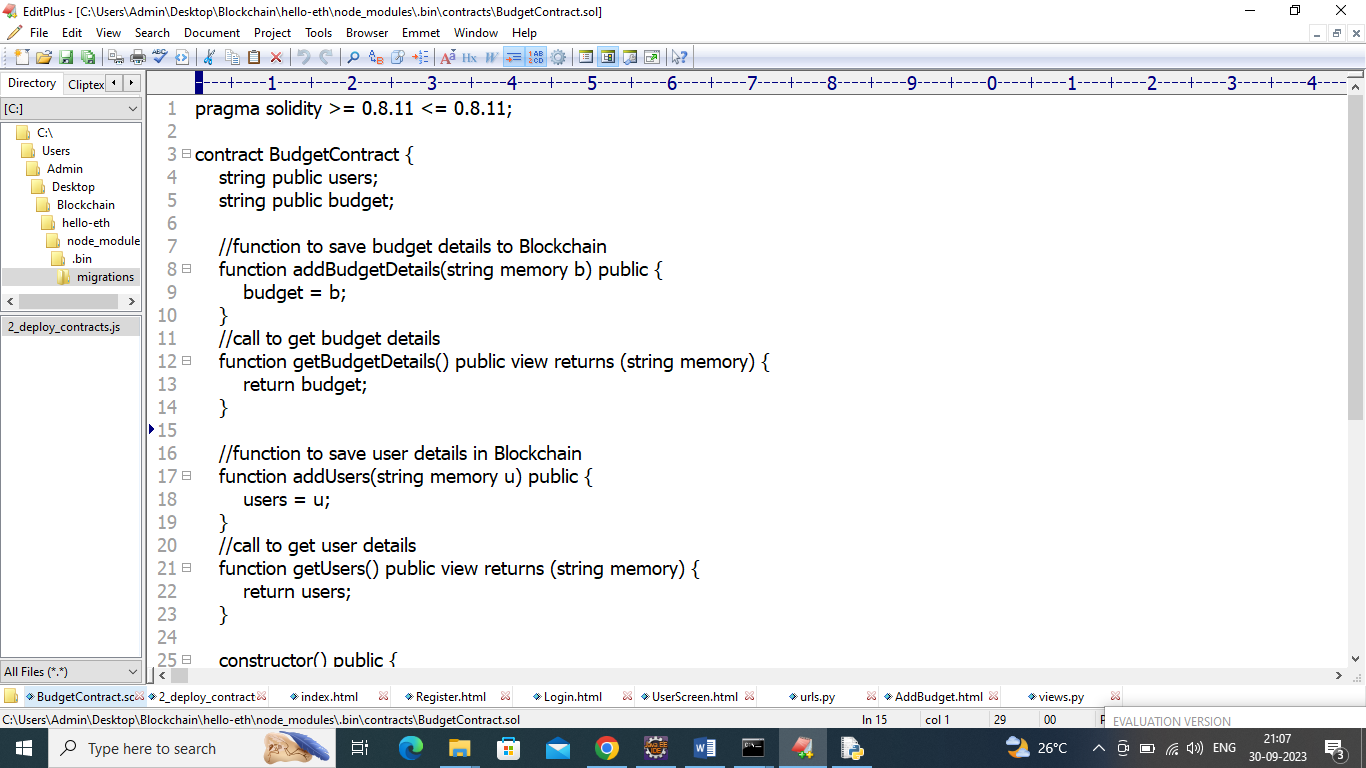
After the module testing, the integration testing is applied. When linking the modules there may be chance for errors to occur, these errors are corrected by using this testing. In this system all modules are connected and tested. The testing results are very correct. Thus the mapping of jobs with resources is done correctly by the system.

**Acceptance Testing**

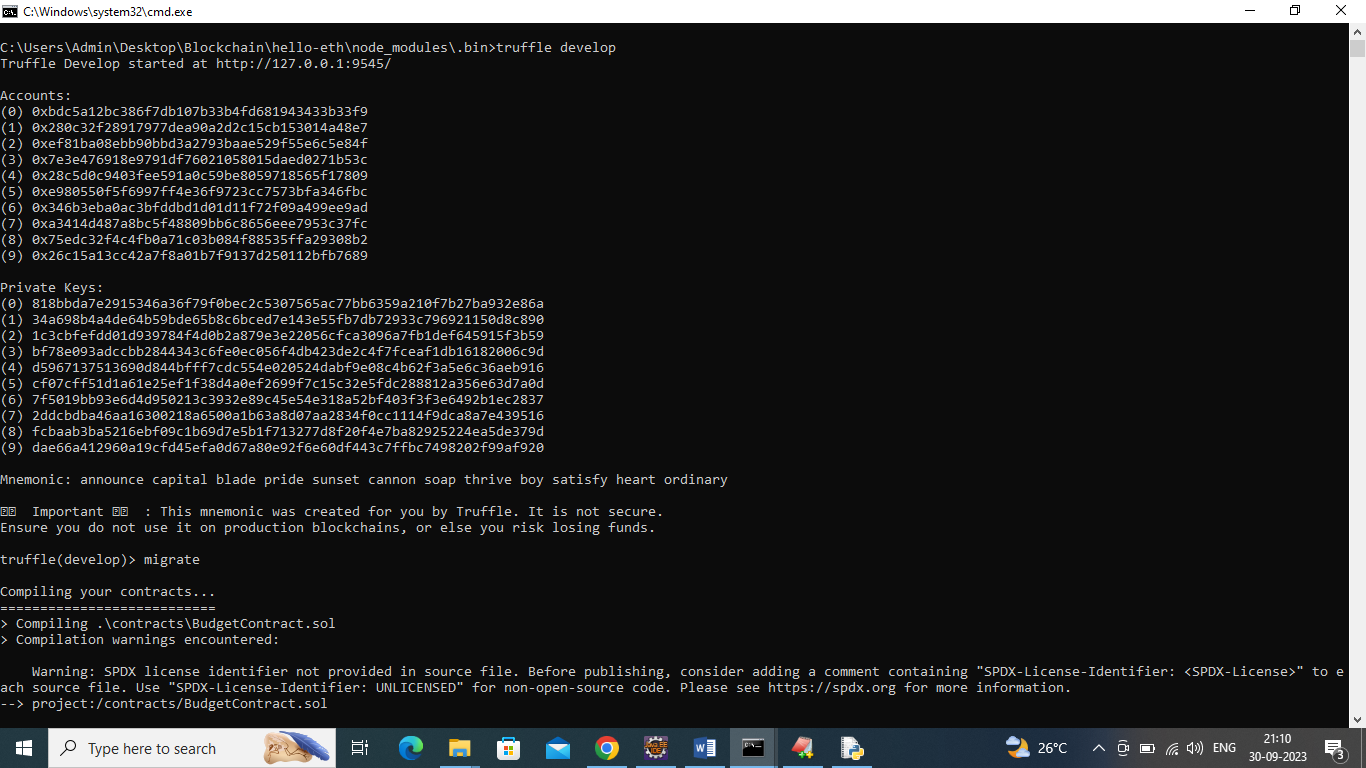
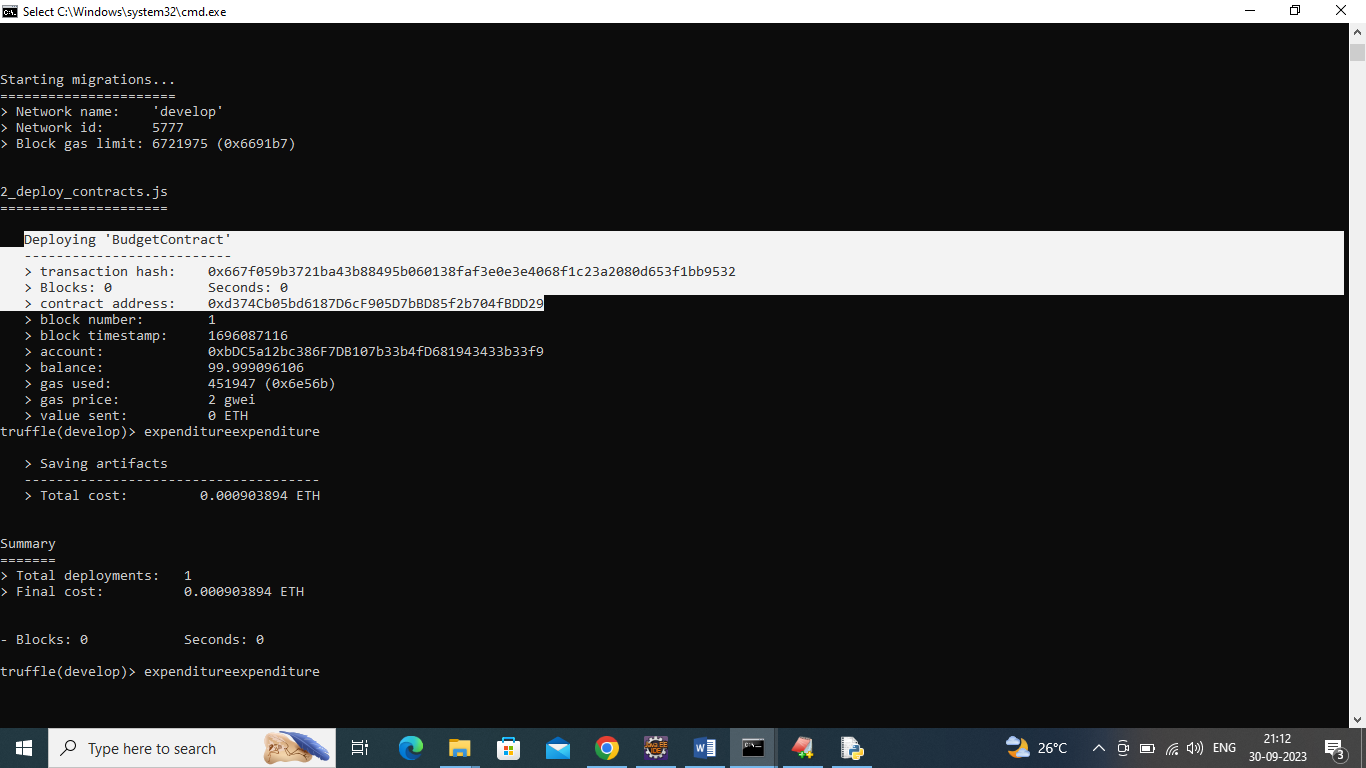
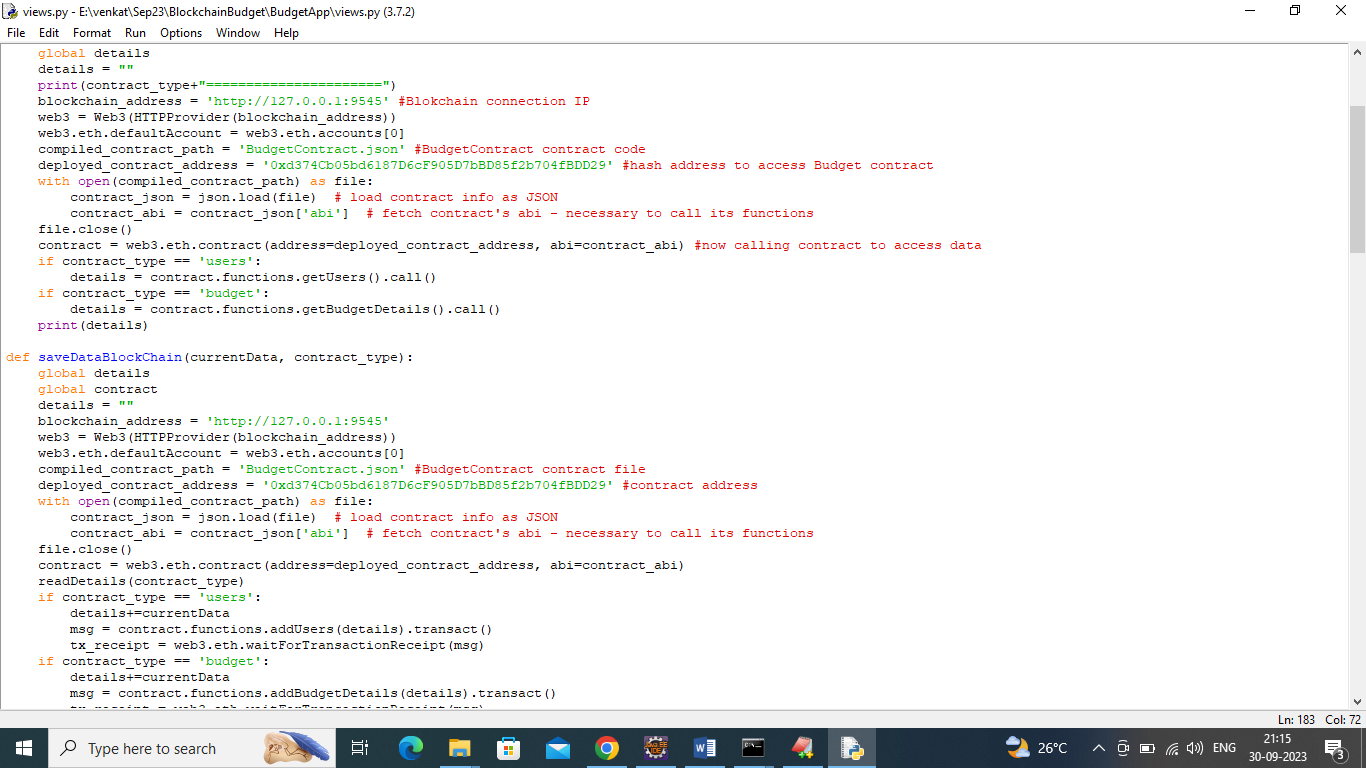
When that user fined no major problems with its accuracy, the system passers through a final acceptance test. This test confirms that the system needs the original goals, objectives and requirements established during analysis without actual execution which elimination wastage of time and money acceptance tests on the shoulders of users and management, it is finally acceptable and ready for the operation.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case Id** | **Test Case Name** | **Test Case Desc.** | **Test Steps** | | | | **Test Case Status** | **Test Priority** |
| **Step** | **Expected** | | **Actual** |
| 01 | User signup | Verify user signup or not | If user may not signup | we cannot do any further operations | we can do further operations | | High | High |
| 02 | User login | Verify user login or not | If user may not login | we cannot do any further operations | we can do further operations | | High | High |
| 03 | Add budget details | Verify budget details added or not | If budget details May not be added | we cannot do any further operations | we can do further operations | | High | High |
| 04 | Track budget | Verify budget tracked or not | If budget is not tracked | We cannot run  operation | We can Run the Operation | | High | High |

**7. SCREENSHOTS:**

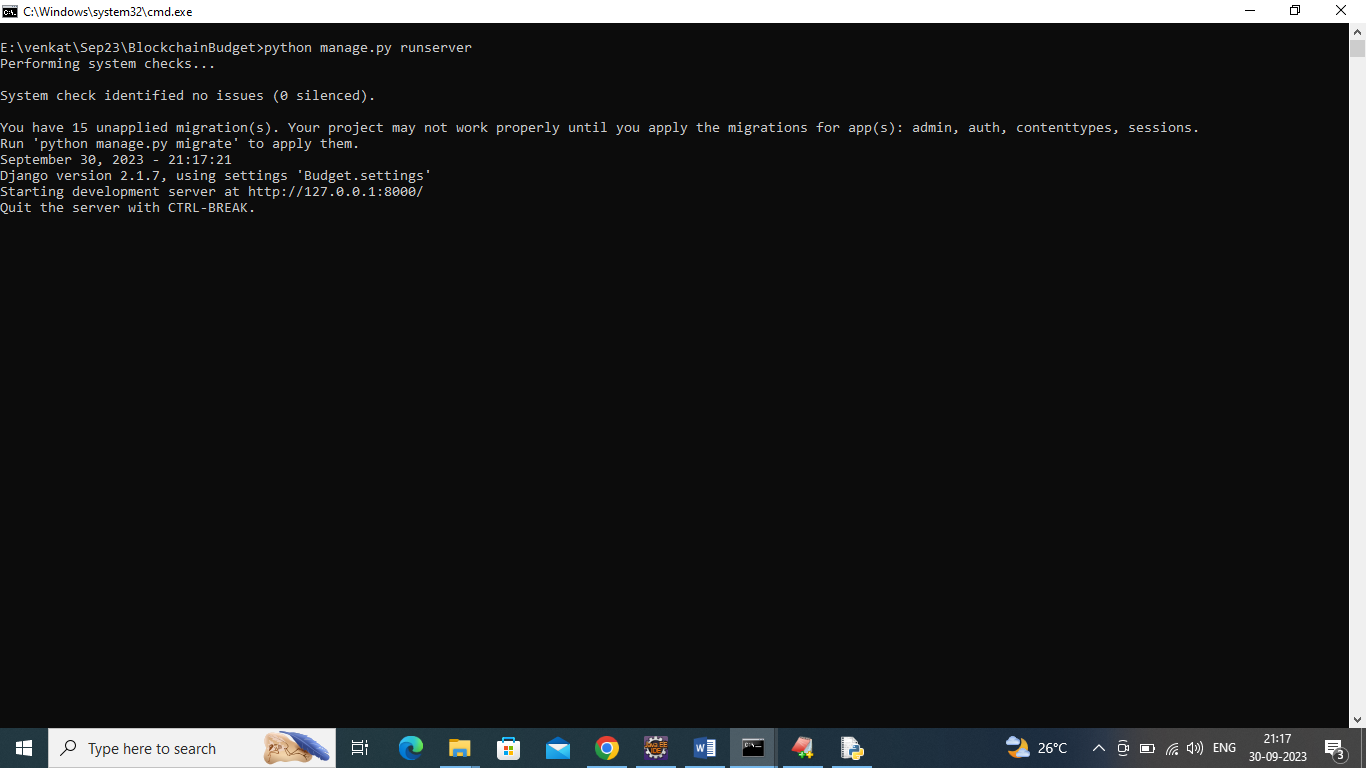


In above contract we have defined functions to save budget and user details and now we need to deploy above contract in Blockchain ETHEREUM by following below steps

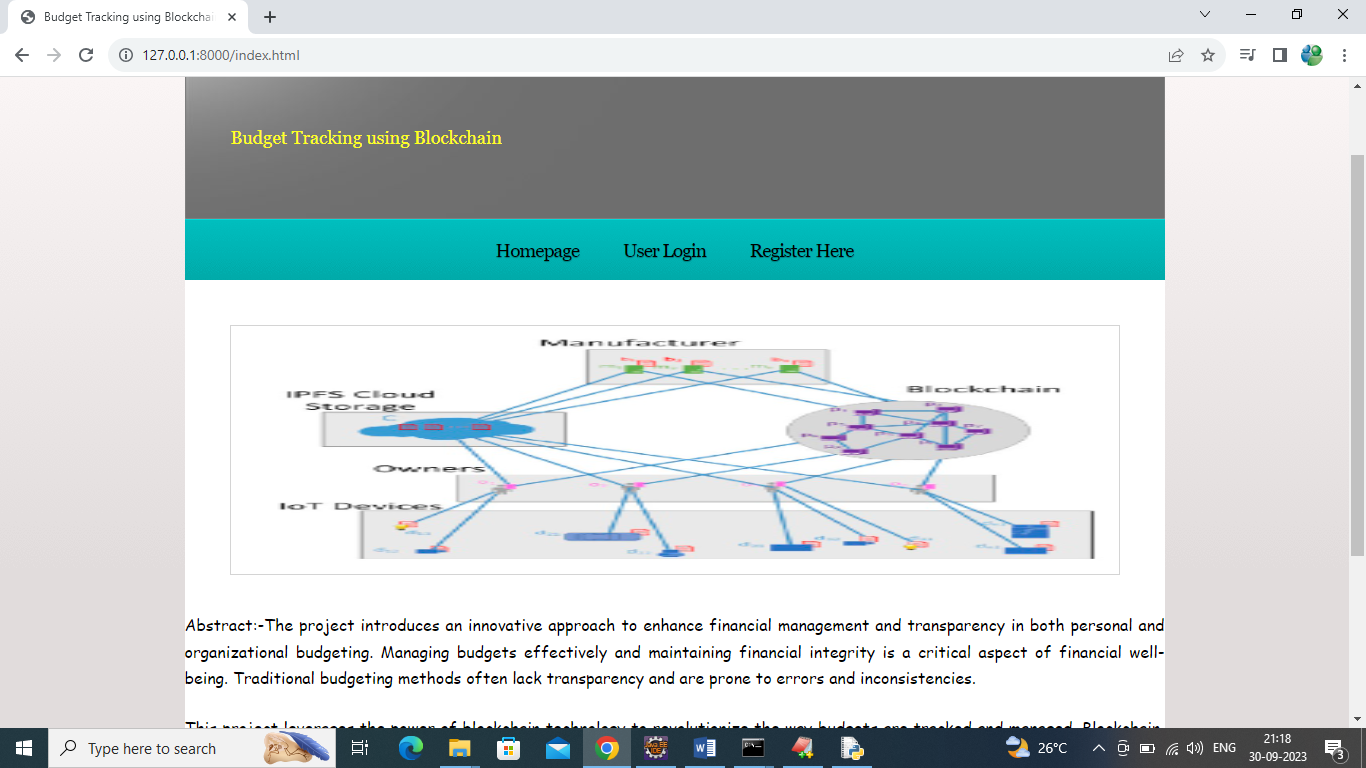
1. First go inside ‘hello-eth/nodemodules/bin’ folder and then find and double click on ‘runBlockchain.bat’ file to start ETHEREUM tool and will get below screen
2. 
3. In above screen Blockchain ETHEREUM generate some default account and private keys and now type command as ‘migrate’ and press enter key to deploy contract and get below output
4. 
5. In above screen in white colour txt we can see ‘Budget Contract’ deployed and got contract address also and this address we can specify in python code to access smart contract to store and get budget data. In below screen showing python code calling smart contract using address
6. 
7. In above screen read red colour comments to know about contract calling from python to store and get budget data. In above black screen contract deployed and let that running.

SCREEN SHOT

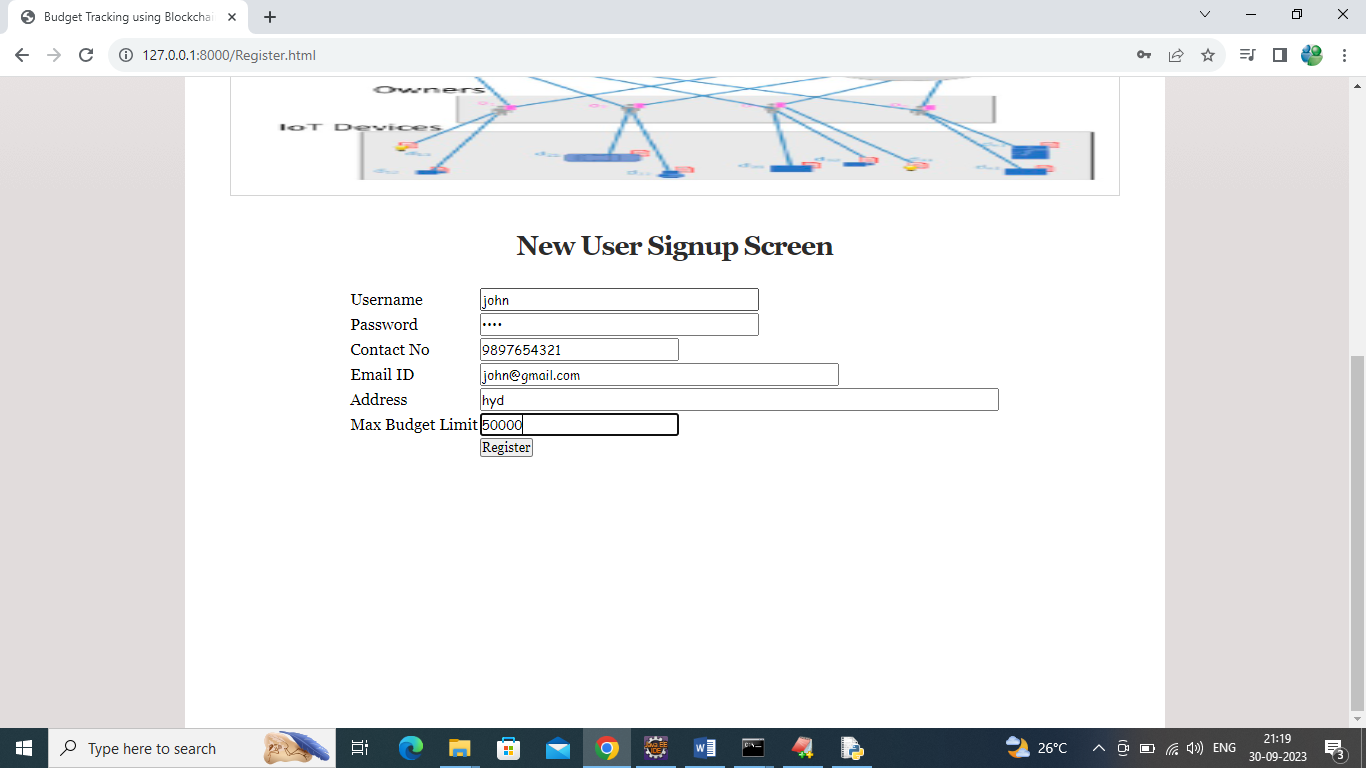
Now double click on ‘run.bat’ file to start python server and get below screen



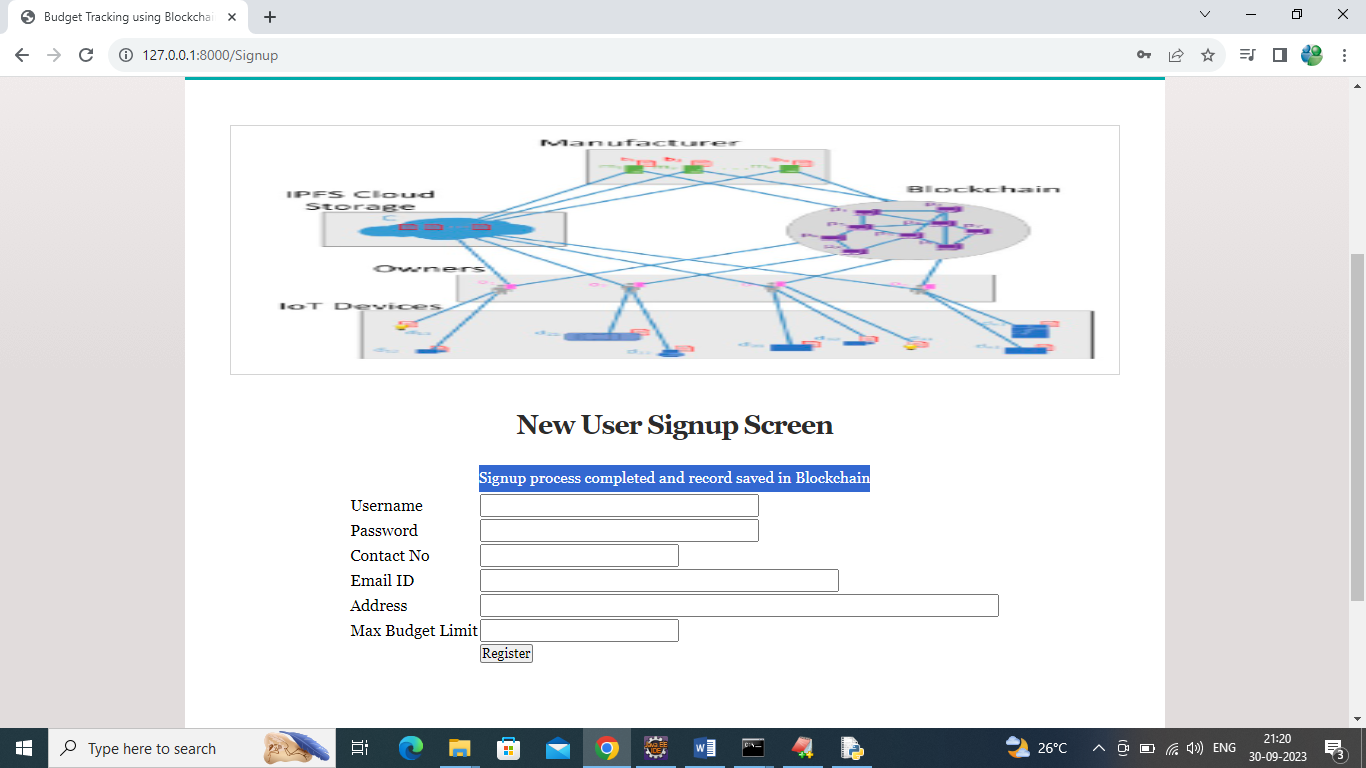
In above screen python server started and now open browser and enter URL as <http://127.0.0.1:8000/index.html> and press enter key to get below home page



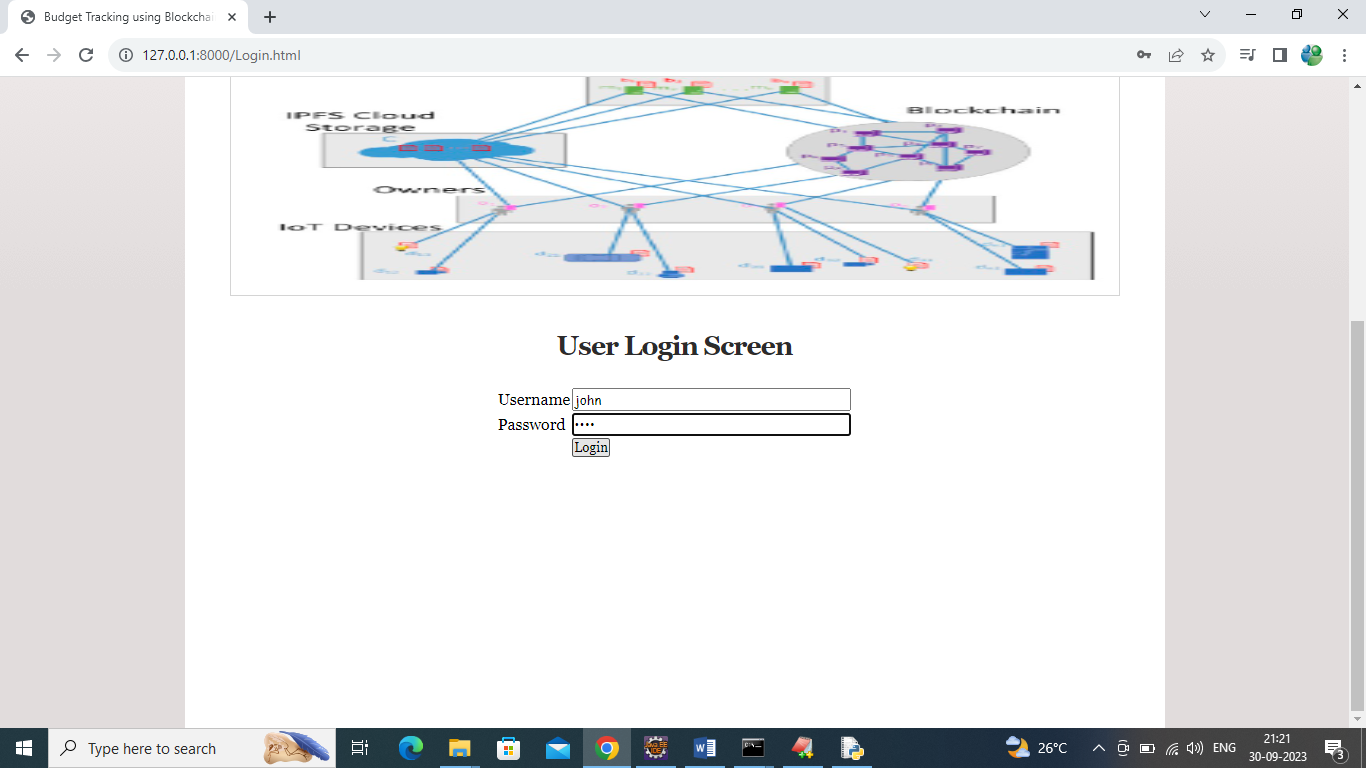
In above screen click on ‘Register Here’ link to get below signup screen



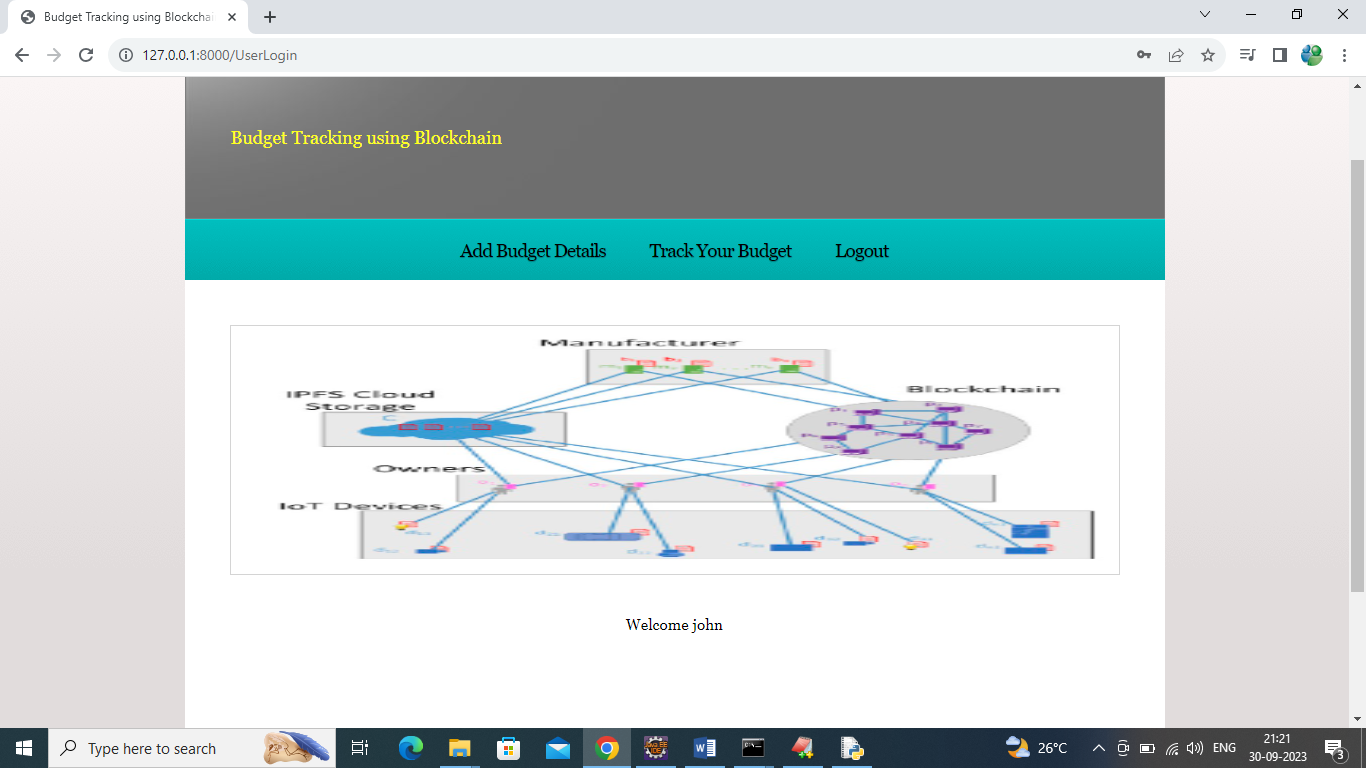
In above screen user is entering signup details with max budget limit and then press button to get below output and this limit will notify user upon tracking



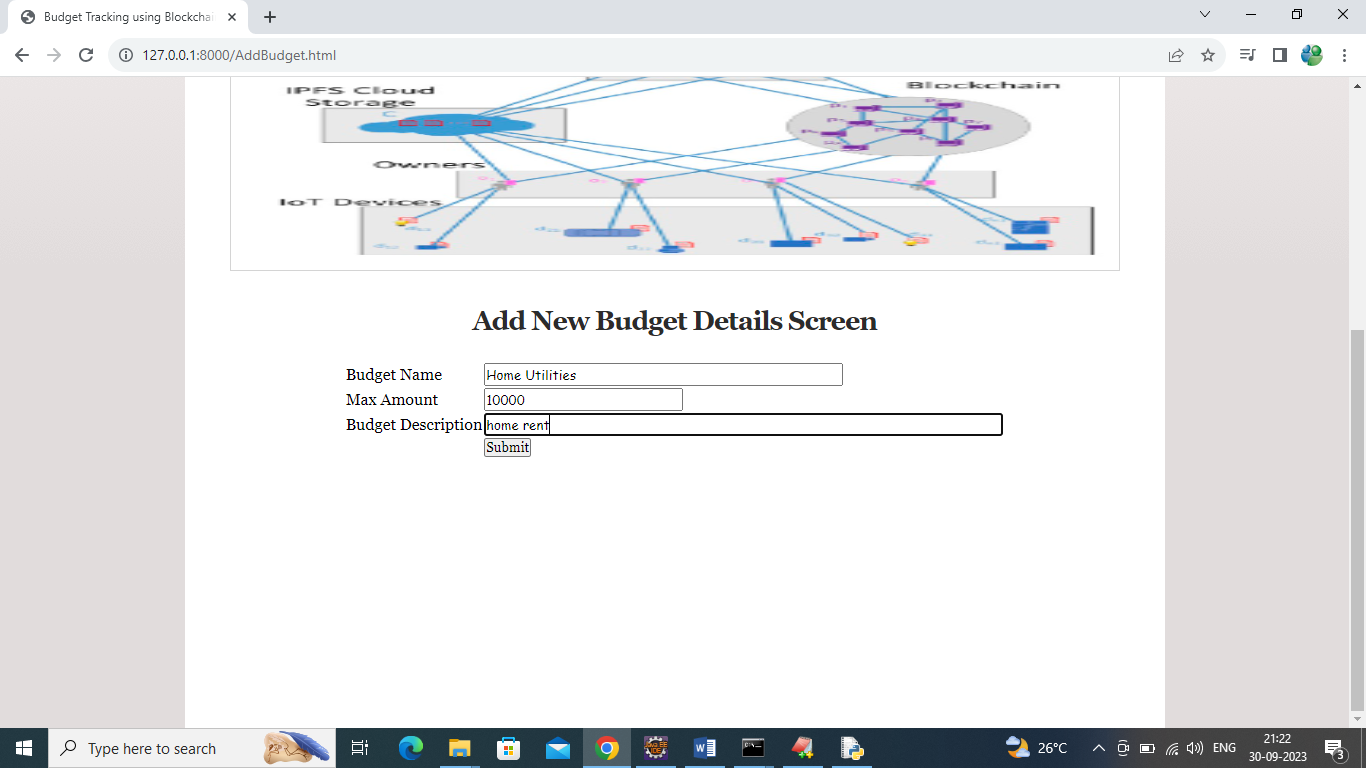
In above screen user signup details saved in Blockchain and now click on ‘User Login’ link to get below login page



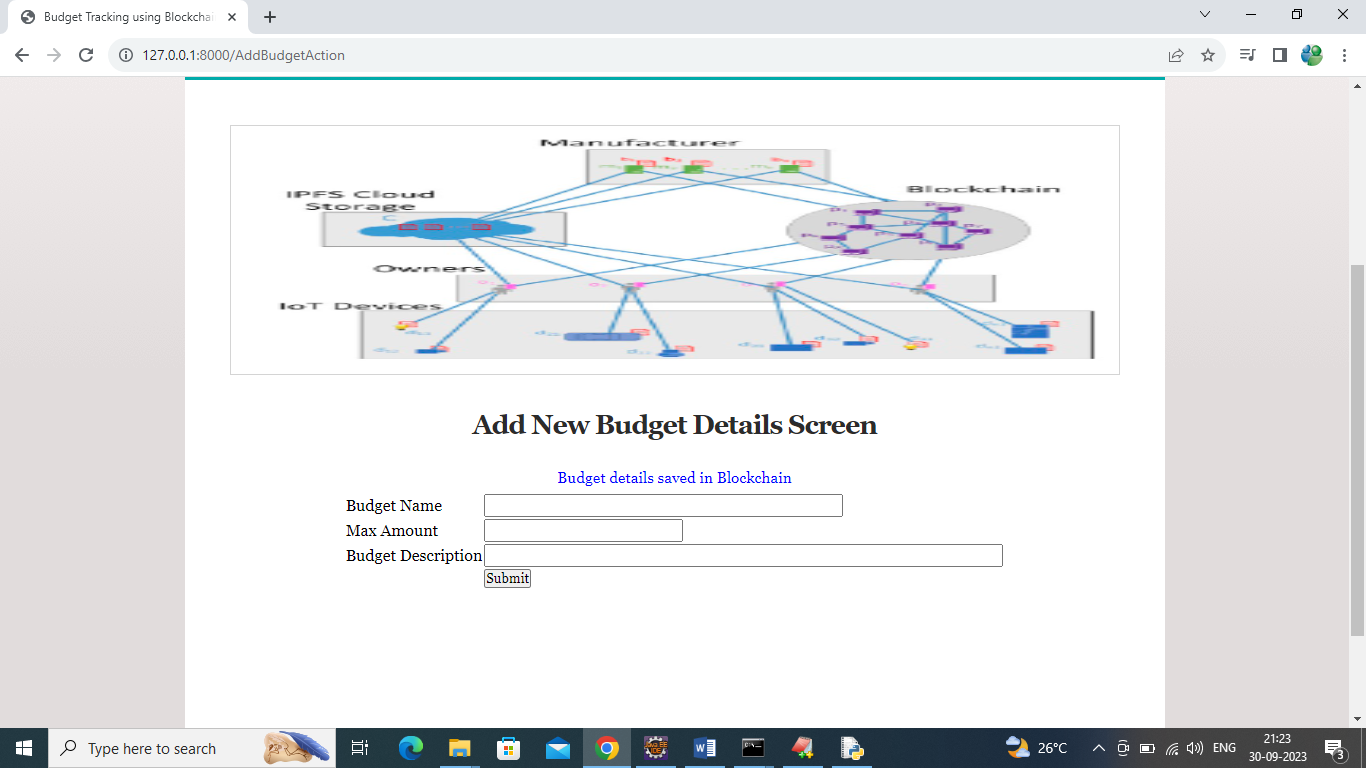
In above screen user is login and after successful login will get below page



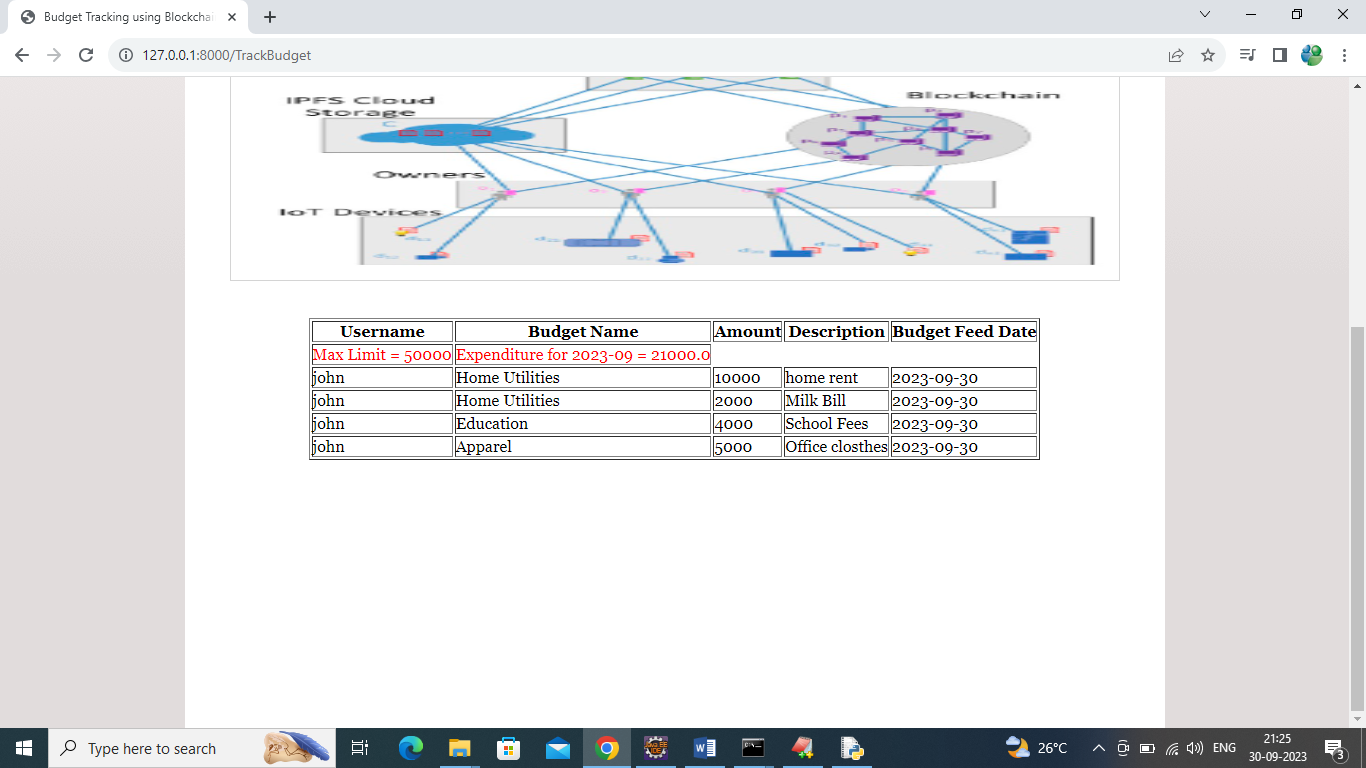
In above screen user can click on ‘Add Budget Details’ to add all expenditure of future budget details



In above screen user is entering some home utility details and then press button to save budget details and get below page



In above screen budget details added in Blockchain and similarly by using this function you can add any number of budget expenditure details. Now click on ‘Track Your Budget’ link to view all expenditure details



While tracking in above screen in red colour text user can see max budget and then can see month wise total expenditure/budget in red colour text and then in remaining rows can see each expenditure with all details. So by using above screen user can easily track or understand his budget with all his expenditure monthwise.

**8. CONCLUSION:**

All existing applications are based on single centralized servers and this server’s data can be easily tamper and there is no way to track such data tamper, sometime hackers can hack this server to crash and in such scenarios services will not be available. To overcome from above issues many applications are migrating to Decentralized Blockchain storage where data will be stored at multiple nodes in a decentralized way and if one node down then services can be access from any other working nodes.

Blockchain stored each record as block or transaction and associate each block with unique hash code, while storing new record Blockchain will verify hash code of all previous blocks and if data not tamper then all blocks will generate same hash code and verification get successful and if block data alter then it will result into incorrect hash code and data alteration will get detected. Hash code verification of Blockchain make it tamper proof and impossible for data alteration.

**9. REFERENCES:**

[1] S. Chandini, T. Poojitha, D. Ranjith, V. J. Mohammed Akram, M. S. Vani, V. Rajyalakshmi, “Online Income and Expense Tracker”, International Research Journal of Engineering and Technology (IRJET), Volume: 06 Issue: 3, e-ISSN: 2395-0056, p-ISSN: 2395- 0072 (March 2019).

[2] N. Zahira Jahan MCA., M. Phil, K. I. Vinodhini, “Personalized Expense Managing Assistant Using Android”, International Journals of Computer Techniques (IJCT), Volume: 3 Issue: 2, ISSN: 2394-2231 (March-April 2018).

[3] Mohan Prasad K, Sai Nagendra Goru Rajeev, Vamsi Desu, Albert Mayan M.J , "Automated Payroll Using GPS Tracking and Image Capture ", IOP Conference Series: Materials Science and Engineering, Vol.590, 012026 ,pp.1-6,2019 doi:10.1088/1757- 899X/590/1/012026

[4] Satpute, M. K., Kale, A., Mandal, A., & Krishnan, R. SURVEY ON CLASSIFICATION ENGINE FOR MONETARY TRANSACTIONS.(April 2020)

[5] Sabab, S. A., Islam, S. S., Rana, M. J., & Hossain, M. (2018, September). eExpense: A smart approach to track everyday expense. In 2018 4th International Conference on Electrical Engineering and Information & Communication Technology (iCEEiCT) (pp. 136-141). IEEE.

[6] Rajaprabha, M. N. (2017). Family Expense Manager Application in Android. MS&E, 263(4), 042050. [7] Kan, C., Lynch, J., & Fernbach, P. (2015). How budgeting helps consumers achieve financial goals. ACR North American Advances.

[8] <https://www.w3schools.com/>

[9]Surya.V, J. Albert Mayan," A Secure Data Sharing Mechanism In Dynamic Cloud By Using KP-ABE", Research Journal of Pharmacy and Technology , Vol 10 , Issue 1 , pp:83-86,2017

[10] Y. Anitha, R. Ranjini, S. Gomathi, “Easy App for Expanses Manager Using Android”, International Journals of Computer Techniques, Volume: 3 Issue: 2, ISSN: 2394-2231 (March April 2016).

[11] Albert Mayan J, Velmurugan A, Nitin Narayanan Kokkoori, Lokesh Koleti,"Forecasting Hospital Admissions in Emergency Department using Data Mining",Journal of Critical Reviews, Vol. 7, Issue.15,pp. 356-362,2020

[12] Velmurugan A, Ravi, T, “Allergy information ontology for enlightening people”, IEEE international conference on computing technologies and intelligent data engineering, ICCTIDE’16, pp. 1- 7, 2016, DOI: 10.1109/ICCTIDE.2016.7725329

[13] Asha Pandian, Bharathi B , Albert Mayan J,Prem Jacob , Pravin ( 2019),"A Comprehensive View of Scheduling Algorithms for MapReduce Framework in Hadoop",Journal of Computational and Theoretical Nanoscience, Vol.16, No. 8, pp. 3582-3586

[14] R.Julian Menezes, Dr.P.Jesu Jeyarin and J.Albert Mayan,"A Scholarly Audit on the Traits of Enciphering, Deciphering Algorithms bifurcated under Symmetric, Assymetric for Wired cum Wireless Environment",Journal of Advanced Research in Dynamical and Control Systems,Vol. 11,pp. 1443-1454,2019

[15] S. Dhamodaran, Albert Mayan J., N. Saibharath, N. Nagendra and M. Sundarrajan, "Spatial interpolation of meteorological data and forecasting rainfall using ensemble techniques",AIP Conference Proceedings 2207, pp.050005 ,2020

[16] Muthukumar B, Albert Mayan J, Nambiar G, Nair Daniel ,"QR Code and Biometric Based Authentication System for Trains",IOP Conference Series: Materials Science and Engineering 590 (2019) 012010,doi:10.1088/1757-899X/590/1/ 012010

[17] M.D.Kamalesh, Albert Mayan. J, Yovan Felix, Dhamodaran S and Mohana Prasad, "Automation of Blood Donation by Data Integration Using Data Mining," 2020 4th International Conference on Trends in Electronics and Informatics