International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 11 | Nov - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Govt Tender Report

kokila S

Abstract

Tendering is a lengthy and complex business process that results in several legal obligations associated with corporate contracts. Several Asian governments used an e-tendering system to carry out the tender procedure.

Government agencies and private organizations are continuously working to bring fairness in this process. The e-Tendering process brings efficiency and convenience in the overall practices which come out in the seen recently.

By facilitating the procurement process with linked parties, e-tendering attempts to promote productivity, effectiveness, and transparency in the tender process. Problems with the e-tendering system remain centralized, giving local groups complete control over the database and system, increasing the risk of fraud, collusion, and manipulation.

To avoid bidder fraud and suspicion, the highly competitive contract selection process must be supported by effective and transparent technology.

AWS S3 cloud service, AES Encryption algorithm techniques is one of the solutions that allow integrated systems without a third party because it adhere to the decentralized nature of distributed databases so that it has a copy of many users.

By adopting AWS S3 cloud service, AES Encryption algorithm techniques & QRCode technique e-tendering can reduce the source of fraud, namely database manipulation.

Introduction:

Tendering is a lengthy and complex business process that results in several legal obligations associated with corporate contracts. Several Asian governments used an e-tendering system to carry out the tender procedure. Government agencies and private organizations are continuously working to bring fairness in this process. The e-Tendering process brings efficiency and convenience in the overall practices which come out in the seen recently. By facilitating the procurement process with linked parties, e-tendering attempts to promote productivity, effectiveness, and transparency in the tender process. Problems with the e-tendering system remain centralized, giving local groups complete control over the database and system, increasing the risk of fraud, collusion, and manipulation. To avoid bidder fraud and suspicion, the highly competitive contract selection process must be supported by effective and transparent technology. AWS S3 cloud service, AES Encryption algorithm techniques is one of the solutions that allow integrated systems without a third party

because it adhere to the decentralized nature of distributed databases so that it has a copy of many users. By adopting AWS S3 cloud service, AES Encryption algorithm techniques & QRCode technique e-tendering can reduce the source of fraud, namely database manipulation.

Literature Survey:

E-Governance: A Tendering Framework Using Block chain With Active Participation of Citizens Yashita Goswami, Ankit Agrawal, Ashutosh Bhatia in 2020	E-governance powered by blockchain technology has emerged as a transformative solution, effectively addressing issues of biases and corruption. The immutability of data offered by blockchain is a key factor enabling this positive change. The framework presented in this paper not only ensures fairness but also encourages citizen participation, fostering a more interactive governance process	The solution proposed in this paper represents a generic framework that focuses on the tendering mechanism, harnessing the potential of blockchain. However, its versatility allows for expansion to encompass additional features and activities of E-governance, making it adaptable to address various other issues as well.
A Secure Network Framework for Government Tender allocation using Block Chain Logith S, Nitish PS, Raghul S, Tamilarasan T in 2020	In order to facilitate a competitive tendering process, we utilized Ethereum to create an edge computing platform. Our goal was to implement an iterative bidding method that effectively matches the most suitable contractors with tendering contracts, ultimately boosting profitability for both government borrowers and construction firms	We conducted a thorough performance assessment of the proposed model. The results were highly promising, as the suggested model outperformed its competitors across multiple tender criteria, delivering superior outcomes

1.1 Problem Statement:

In today's digital landscape, data manipulation has become a significant concern as adversaries and malicious entities exploit it to harm the public and Government bodies.

The complexity of administrative systems in many Governments often leads to highly inefficient workflows, riddled with corruption, mismanagement, and human errors. This can result in Government officials misusing their bureaucratic powers and demanding exorbitant bribes for passing tenders.

Our primary aim in developing this system is to enhance the profitability of both the government department and bidder companies. By addressing issues of data manipulation and streamlining administrative processes, we seek to create a more transparent and fairer environment for tendering activities. This will help curb corruption and ensure a level playing field for all parties involved.

1.2 Aim

The aim of this project is to develop an application. It provides security for the government e-tender by automate methodology technique with AWS S3 service.

Existing System:

The current system suffers from two significant challenges: lack of transparency and security vulnerabilities. These issues are particularly concerning for the portals, as they can result in fraud and data manipulation within the centralized database. In the event of a security breach, sensitive bid information can be leaked to competitors, causing severe financial and strategic losses for businesses.

Hardware and Software Requirements:

Hardware Requirements:

Processor : Pentium 4 +

RAM : 4GB

Hard Disk : Minimum of 80 GB.

Speed : 1.2 GHz+

Software Requirements:

OS : Windows 7 or Higher

Back End : MySQL Server

Designed Tool Kit : Visual Studio 2010

Front End : Windows Application (TCP Remoting)

Programming Language : C#

Proposed System:

We propose the development of a TCP Remoting application to enhance the security of tender data. In this system, tender details and quotation information will be securely stored in AWS S3 Services. To ensure

utmost confidentiality, each tender quotation is encrypted using the AES Encryption algorithm, then written to a QR Code Image, and finally stored in the AWS S3 service.

1.3 Objectives

The objective of this project is to create a TCP Remoting Network windows application that enhances the security of e-tendering. This is achieved by implementing advanced security measures such as the AES Encryption algorithm, AWS S3 Service, and QR Code technique.

Methodology:

Application Manager

- Login (Default Id, Password)
- Manage Department
- Add Government Employee based on Department

Government Employee

- Login
- Create Government Tender based on Department
- View Apply Tender-Bidder Company-Ratings-AWS
- Approve Tender

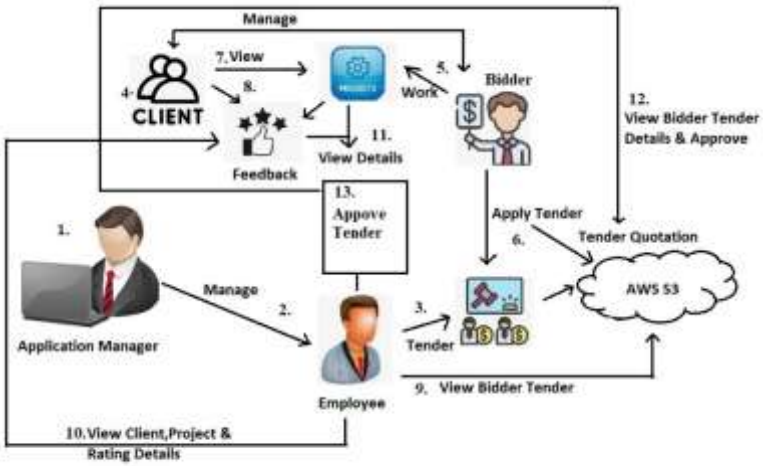
Bidder Company

- Registration
- Login
- Add Client Details
- Add Client Project Details
- Apply Government Tender AES-Encryption-AWS S3

Client Company

- Login
- View Project Work
- Post Feedback Based on Project-NLP

Architecture:



Expected outcome:

The successful implementation of our system marks a momentous achievement in revolutionizing the government tender assignment process. Throughout this project, we have highlighted the significance of AES Encryption and AWS S3 Service, showcasing their undeniable benefits in ensuring robust security and efficient data management. By creating a comprehensive end-to-end automation solution, we have taken significant strides towards optimizing the government tender workflow.

The adoption of AES Encryption algorithm stands as a crucial step in safeguarding the integrity and confidentiality of sensitive tender information. AES Encryption has played a pivotal role in fortifying our security measures, thwarting unauthorized access and ensuring that only qualified and deserving bidders gain access to tender projects. This has enhanced the transparency of the government tendering system and fostered healthy competition among bidder companies.

The integration of AWS S3 Service has transformed the storage and retrieval of data in our system. Leveraging cloud computing capabilities, AWS S3 Service has improved accessibility, scalability, and redundancy. By reducing reliance on local infrastructure, we have minimized the risk of data loss and system downtime, ensuring the continuity and reliability of our automated workflow

Chapter 2 – Literature Survey

2.1 Related Works

Title: E-Governance: A Tendering Framework Using Block chain With Active Participation of Citizens

Authors: Yashita Goswami, Ankit Agrawal, Ashutosh Bhatia

Year: 2020

E-governance powered by blockchain technology has emerged as a transformative solution, effectively addressing issues of biases and corruption. The immutability of data offered by blockchain is a key factor

enabling this positive change. The framework presented in this paper not only ensures fairness but also encourages citizen participation, fostering a more interactive governance process.

Through the implementation of blockchain, the tendering mechanism gains enhanced transparency as evaluations are conducted based on deployed business logic, minimizing biases associated with manual processes. Moreover, the technology fosters trust between bidding and tendering organizations. Once a bid proposal is submitted, it remains unchanged, instilling confidence in the tendering organization and significantly reducing the risk of fraud.

The tracking of funds within this system further promotes transparency, offering a clear view of how funds are utilized in a fair manner. The solution proposed in this paper represents a generic framework that focuses on the tendering mechanism, harnessing the potential of blockchain. However, its versatility allows for expansion to encompass additional features and activities of E-governance, making it adaptable to address various other issues as well.

Title: A Secure Network Framework for Government Tender allocation using Block Chain
Authors: Logith S, Nitish PS, Raghul S, Tamilarasan T
Year: 2020
In order to facilitate a competitive tendering process, we utilized Ethereum to create an edge computing platform. Our goal was to implement an iterative bidding method that effectively matches the most suitable contractors with tendering contracts, ultimately boosting profitability for both government borrowers and construction firms.

Additionally, we conducted a thorough performance assessment of the proposed model. The results were highly promising, as the suggested model outperformed its competitors across multiple tender criteria, delivering superior outcomes.

2.2 Comparison Tables and more about the methodology of various papers

paper	objectives	Methodologies
E-Governance: A Tendering Framework Using Block chain With Active Participation of Citizens Yashita Goswami, Ankit Agrawal,	E-governance powered by blockchain technology has emerged as a transformative solution, effectively addressing issues of biases and corruption. The immutability of data offered by blockchain is a key factor enabling this positive change. The	The solution proposed in this paper represents a generic framework that focuses on the tendering mechanism, harnessing the potential of blockchain. However, its versatility allows for expansion to encompass additional features and activities of E-governance,

Ashutosh Bhatia in 2020	framework presented in this paper not only ensures fairness but also encourages citizen participation, fostering a more interactive governance process	making it adaptable to address various other issues as well.
A Secure Network Framework for Government Tender allocation using Block Chain Logith S, Nitish PS, Raghul S, Tamilarasan T in 2020	In order to facilitate a competitive tendering process, we utilized Ethereum to create an edge computing platform. Our goal was to implement an iterative bidding method that effectively matches the most suitable contractors with tendering contracts, ultimately boosting profitability for both government borrowers and construction firms	We conducted a thorough performance assessment of the proposed model. The results were highly promising, as the suggested model outperformed its competitors across multiple tender criteria, delivering superior outcomes

SOFTWARE REQUIREMENT ANALYSIS

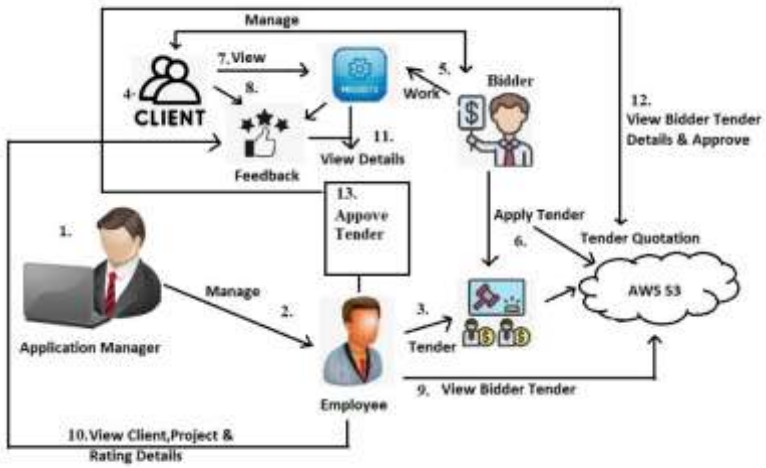
INTRODUCTION TO SRS

The introduction of the Software Requirements Specification (SRS) provides an overview of the entire SRS with purpose, scope, definitions, acronyms, abbreviations, references and overview of the SRS. The aim of this document is to gather, analyse, and give an in-depth insight of the complete "A Secure Network Framework for Government Tender Allocation Based on AWS Service" by defining the problem statement in detail.

PURPOSE

The Purpose of the Software Requirements Specification is to provide the technical, Functional and non-functional features, required to develop a web application App. The entire application designed to provide user flexibility for finding the shortest and/or time saving path. In short, the purpose of this SRS document is to provide a detailed overview of our software product, its parameters and goals. This document describes the project's target audience and its user interface, hardware and software requirements. It defines how our client, team and audience see the product and its functionality.

Architecture:



TECHNOLOGY:

C# Language

C# is a versatile and robust programming language developed by Microsoft, widely utilized for creating various software applications. It seamlessly combines the power and efficiency of C++ with the user-friendliness of Visual Basic, making it a favored choice among developers. As an object-oriented language, C# focuses on creating and manipulating objects to construct programs. It supports essential object-oriented programming concepts such as encapsulation, inheritance, and polymorphism.

With C#, developers can build reusable and modular code by defining classes, serving as blueprints for objects. The language boasts a rich set of features, including strong typing, automatic memory management through garbage collection, and an extensive standard library.

The syntax of C# is both robust and easy to read, making it suitable for beginners and experienced developers alike. Sharing similarities with other C-style languages, switching between languages becomes more accessible for developers. C# offers a diverse array of built-in data types, allowing for flexible data manipulation, such as integers, floating-point numbers, characters, Booleans, and strings. Control flow statements like if-else, switch, and loops empower developers to govern the execution flow within their programs.

C# supports exception handling, enabling developers to gracefully catch and manage runtime errors, ensuring the stability and reliability of applications. Additionally, C# includes powerful features like lambda expressions, LINQ (Language Integrated Query), generics, delegates, events, and attributes, enhancing code expressiveness and efficiency.

Developers can utilize integrated development environments (IDEs) like Microsoft Visual Studio to build C# applications, providing a comprehensive set of tools, debuggers, and code editors. C# benefits from a vibrant and supportive community, offering abundant resources, tutorials, and libraries that expedite the development process.

In conclusion, C# is a potent, expressive, and widely adopted language for software development. Its blend of simplicity, performance, and strong support for object-oriented programming makes it a preferred choice for building a wide range of applications, from desktop software to web and mobile applications.

ASP.NET Framework

ASP.NET, a robust and widely adopted web development framework from Microsoft, forms the core of my project report. This report delves into the various aspects of ASP.NET, highlighting its key features and capabilities. ASP.NET empowers developers to create dynamic and interactive web applications using the .NET framework. The framework is built on the Common Language Runtime (CLR), offering advantages such as memory management and enhanced security.

ASP.NET provides several components that simplify web development, including Web Forms, MVC (Model-View-Controller) architecture, and Web API for creating RESTful services. Web Forms facilitates a drag-and-drop approach to build visually appealing and responsive web user interfaces. In contrast, MVC follows a structured pattern, separating the application into model, view, and controller components, ensuring a more modular and maintainable codebase. Web API empowers developers to construct web services consumed by various clients, making it ideal for developing APIs and implementing microservices architecture.

The framework comes equipped with a wide range of libraries and controls that expedite development, encompassing data controls for seamless database interaction, authentication and authorization mechanisms for user management, and caching mechanisms for optimizing performance. Additionally, ASP.NET seamlessly integrates with other Microsoft technologies, including SQL Server for efficient database management and IIS (Internet Information Services) for hosting web applications.

Moreover, ASP.NET is language-agnostic, supporting multiple programming languages, primarily C# and VB.NET. This flexibility allows developers to leverage their preferred language skills. The framework also offers seamless integration with Visual Studio, a powerful IDE equipped with features like IntelliSense, debugging tools, and project management capabilities.

To conclude, ASP.NET stands as a comprehensive and versatile platform for building web applications. Its diverse components, extensive libraries, and seamless integration with Microsoft technologies make it the go-to choose for developers seeking efficient and scalable solutions for web development projects.

Amazon Simple Storage Services (S3)

Amazon Simple Storage Service (S3) is an exceptionally scalable and reliable cloud storage solution provided by Amazon Web Services (AWS). Let me elaborate on the key features and benefits of AWS S3 in this paragraph.

AWS S3 empowers users to store and retrieve vast amounts of data from anywhere on the web. Its user-friendly and intuitive interface allows for easy data upload, download, and management. S3 is designed to ensure high availability and durability, safeguarding data against hardware failures or disasters. This is accomplished through automatic data replication across multiple servers and facilities within an AWS region, ensuring data accessibility even if one server or facility experiences issues.

S3 caters to various use cases with its different storage classes. The standard storage class provides high-performance and immediate data access, making it ideal for frequently accessed data. The infrequent access storage class offers a cost-effective solution for less frequently accessed data while maintaining fast retrieval times. Additionally, Glacier and Glacier Deep Archive storage classes are tailored for long-term data retention, featuring lower costs and retrieval times.

Scalability is a standout feature of S3, capable of storing virtually limitless data, making it suitable for both small businesses and large enterprises. The service also offers granular access controls, enabling users to define data access permissions. Data transfers over the internet are secure, facilitated by SSL/TLS encryption, and S3 seamlessly integrates with

S3's seamless integration with other AWS services allows users to build highly scalable and reliable applications. It serves as a versatile storage solution for websites, mobile applications, data backup and restore, content distribution, big data analytics, and more. S3 provides additional features such as versioning, event notifications, and lifecycle management policies to further enhance data management capabilities.

In conclusion, AWS S3 is a versatile and dependable cloud storage service, offering scalable, durable, and secure storage for a wide range of applications. Its impressive features, including high availability, scalability, diverse storage classes, access controls, and seamless integration with other AWS services, make it the preferred choice for organizations seeking reliable and cost-effective data storage solutions in the cloud.

General Constraints

The results generated have to be entered in to the system and any error or any value entered out of the boundary will not be understood by the system. In any case if the database crashes, the whole information collected and the results generated will be of no use.

3.3 Specific Requirements

3.3.1 External Interface Requirements

This section provides a detailed description of all inputs into and outputs from the system. It also gives a description of the hardware, software and communication interfaces and provides basic prototypes of the user interface.

Methodologies:

The government staff added based on department wise, his credentials are sent to his mail address using SMTP, Login to the application and add the tender details, Bidder Company register & login to the application and view for the tender details and bid for the tender based on the department wise. The tender quotation details secure using AES Encryption algorithm, QRCode image & store in AWS S3 service. Bidder Company should maintain previous work details, Company profile and client company details. Client Company will post their feedback and Ratings related to work to the bidder company. The Rating point is calculated based on NLP Concept. Government staff verifies the details provided by the bidder and bidder tender quotation details under AES Encryption algorithm, QRCode image & store in AWS S3 service. Government staff verifies the bidder company details, quotation amount, client previous work, client rating, company profile and declares the winner the bidder company can view the selected winner on that tender category.

Hardware Requirements

Processor	:	Pentium 4 +
RAM	:	4GB
Hard Disk	:	Minimum of 80 GB.
Speed	:	1.2 GHz+

Software Requirements

OS	:	Windows 7 or Higher
Back End	:	MySQL Server
Designed Tool Kit	:	Visual Studio 2010
Front End	:	Windows Application (TCP Remoting)
Programming Language	:	C#

System Design

The system design document is very important and necessary part in the project deployment. The main reason to design an any project to understand the architecture working and to understand the project flow to the stakeholders. the system design is developed to meet our project requirement and to plan accordingly. In our project we are going to design and high level or level system so that it can helps to achieve the goals of the project.

The design document helps to analyse the hardware and software requirement of the project.in our project our domain is data security. When it comes to the implementation of the project there are some different concepts and the different methods need to followed. When it comes to machine learning and deep learning there are several steps and procedure need to be followed as mentioned below.

Methodology:

Application Manager

- Login (Default Id, Password)
- Manage Department
- Add Government Employee based on Department

Government Employee

- Login
- Create Government Tender based on Department
- View Apply Tender-Bidder Company-Ratings-AWS
- Approve Tender

Bidder Company

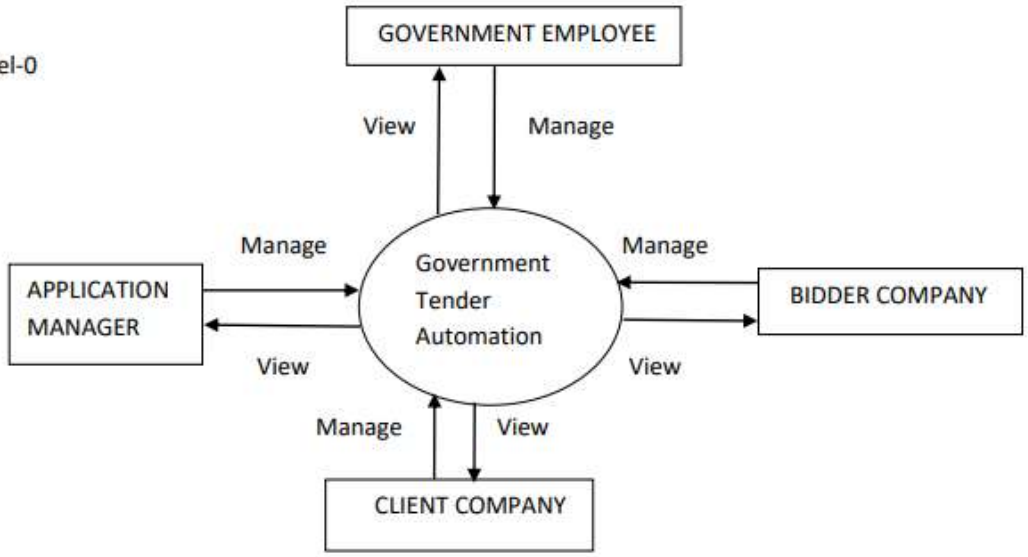
- Registration
- Login
- Add Client Details
- Add Client Project Details
- Apply Government Tender AES-Encryption-AWS S3

Client Company

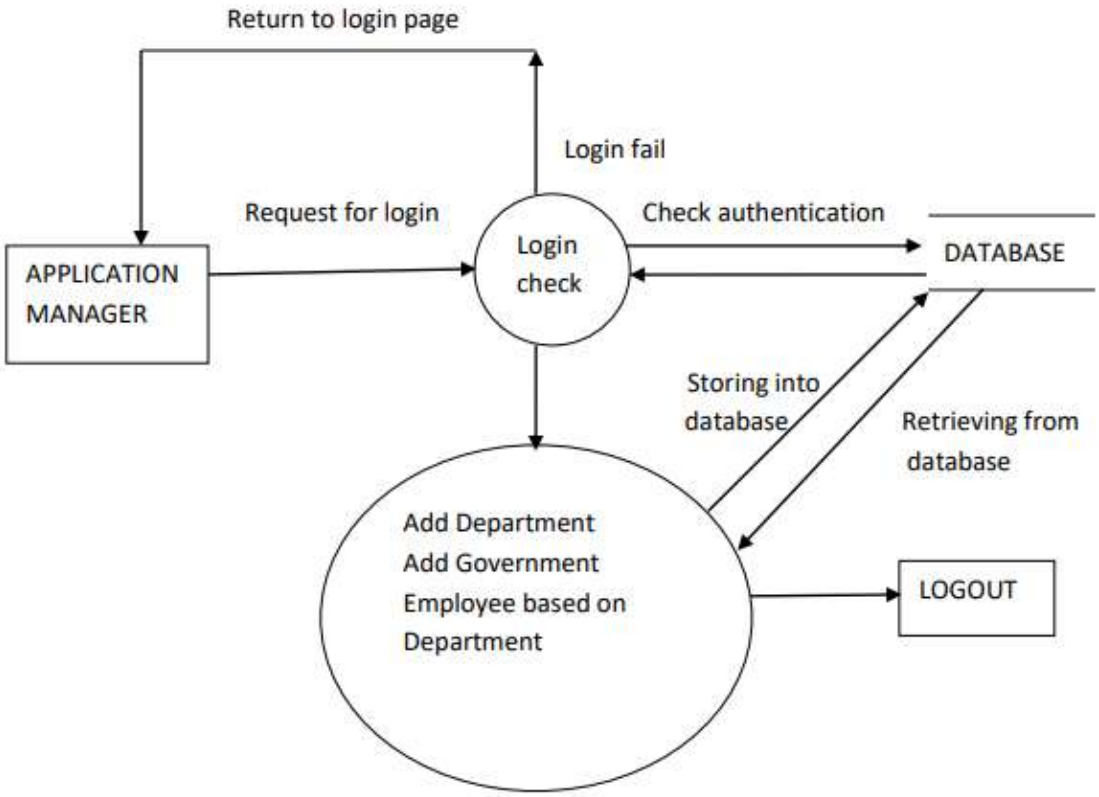
- Login
- View Project Work
- Post Feedback Based on Project-NLP

Cfd:Context Flow Diagram

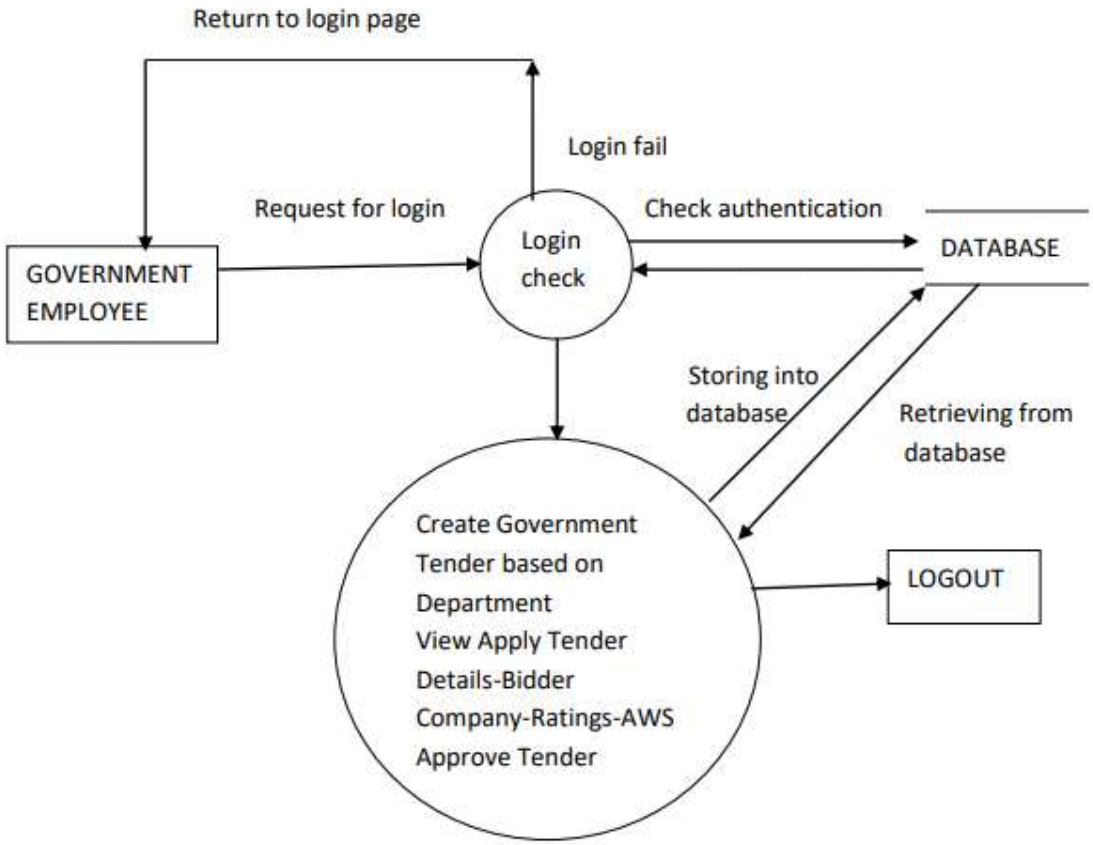
Level-0



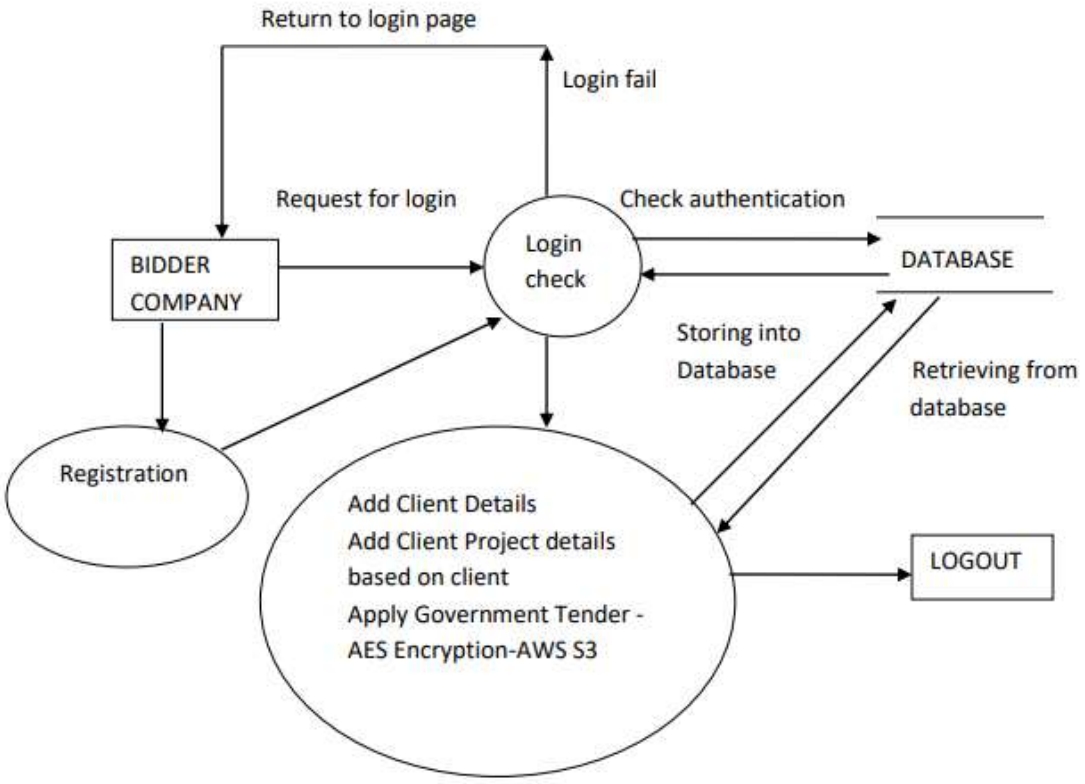
Dfd:Level 1:Application Manager



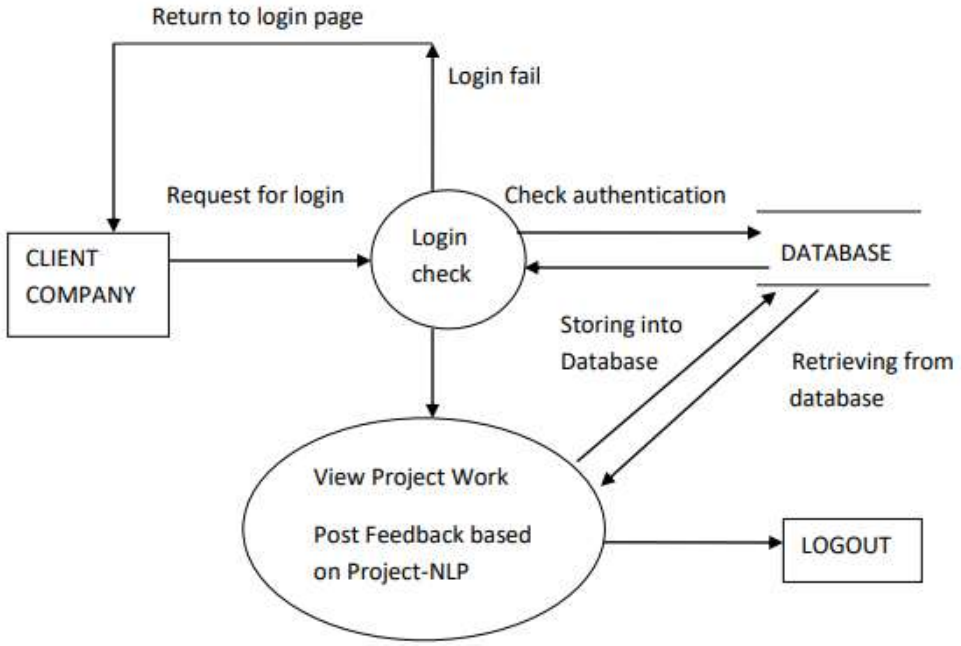
DFD :Level1 Government Employee



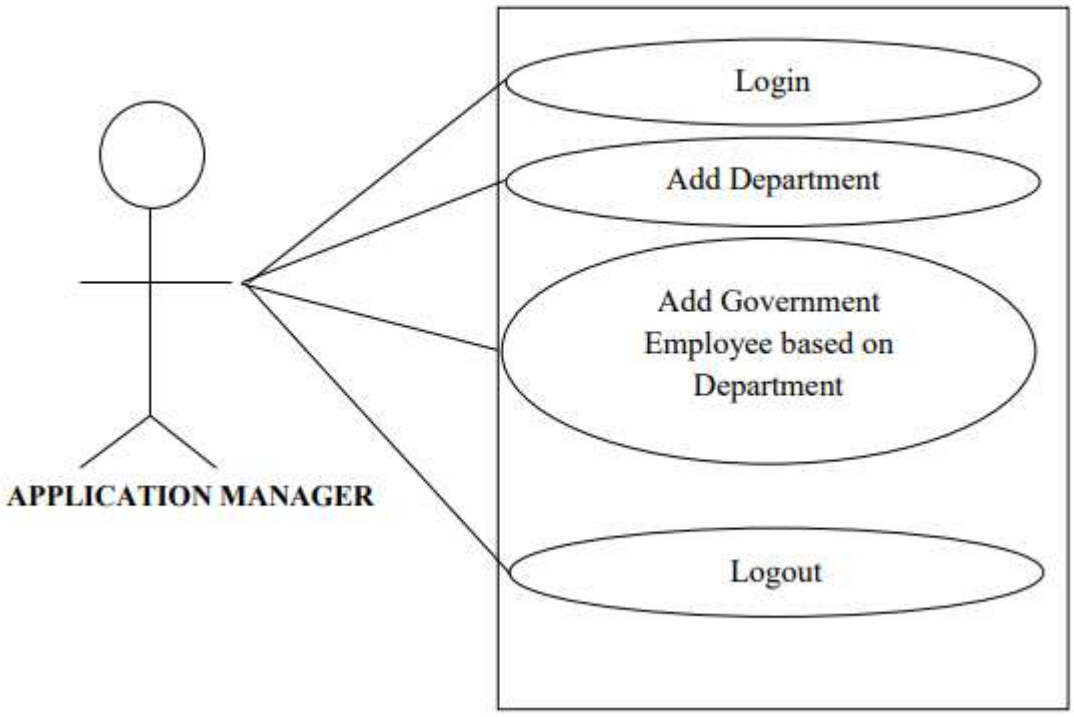
Dfd:Level 1:Bidder Company

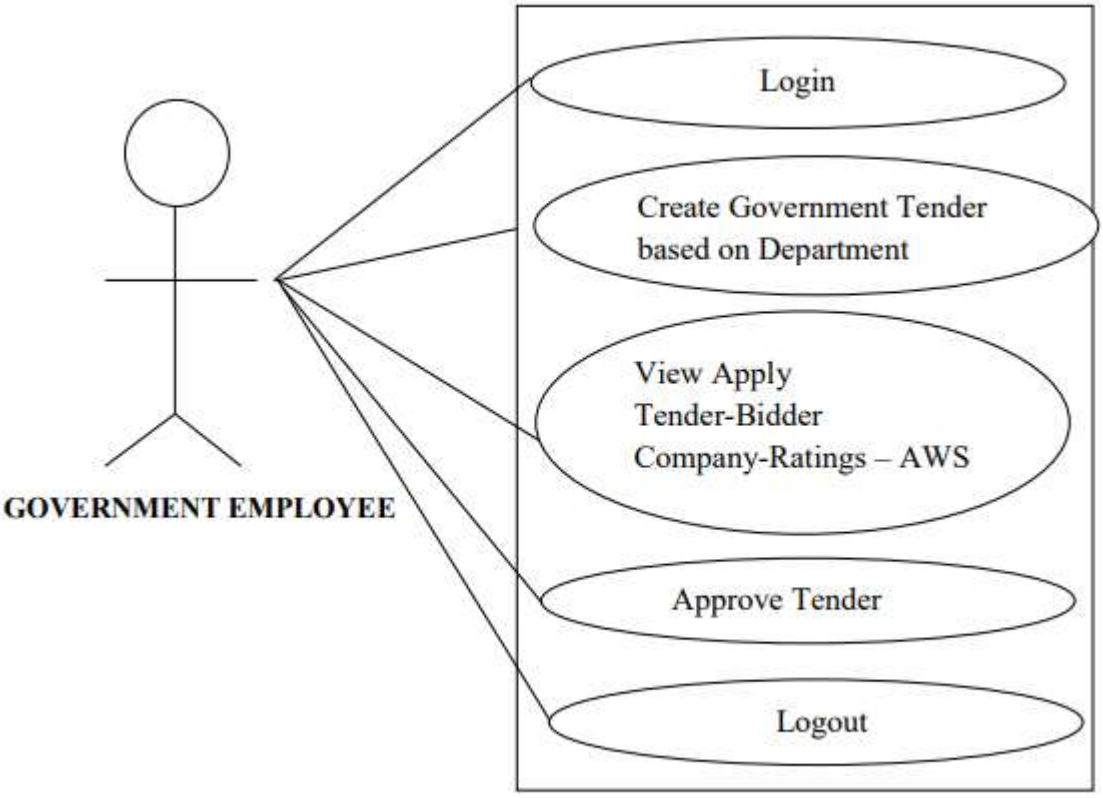


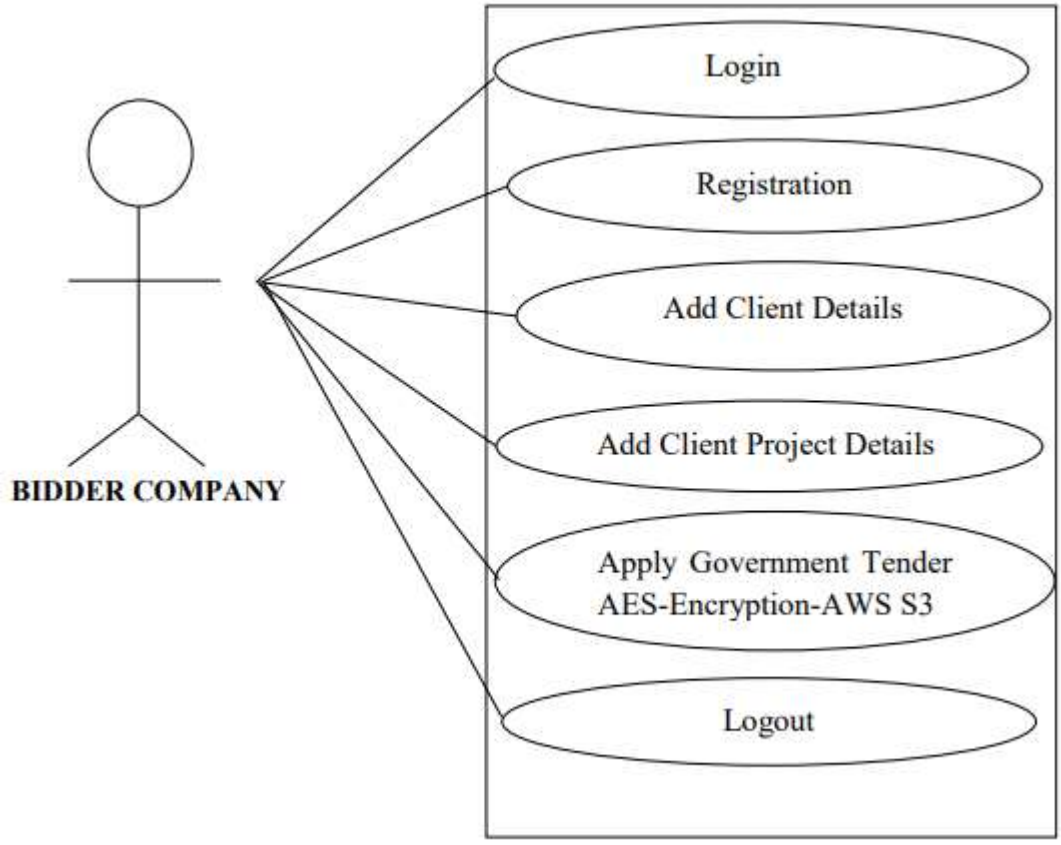
Dfd:Level 1:Client Company

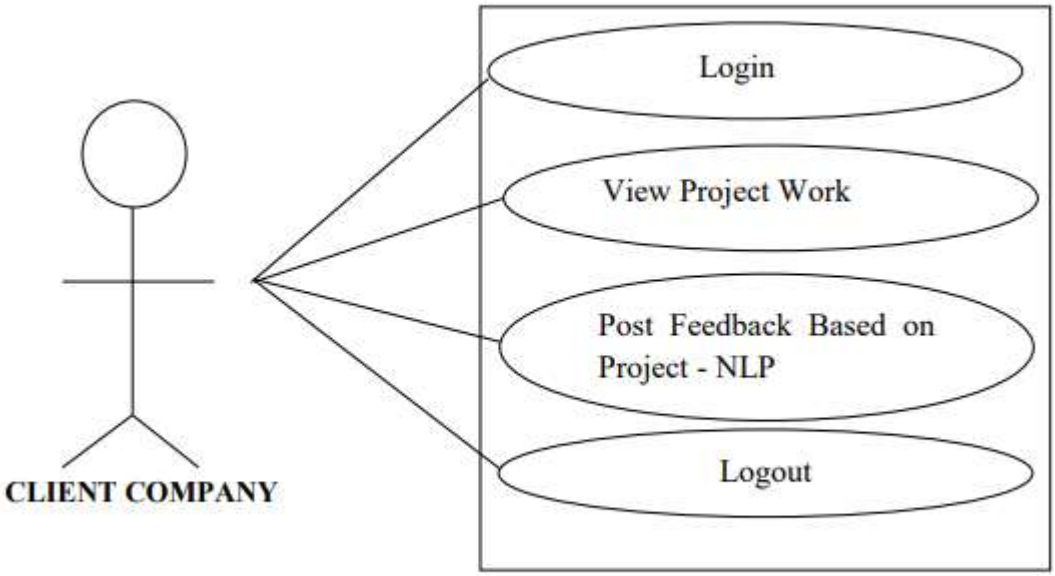


Use case Diagram

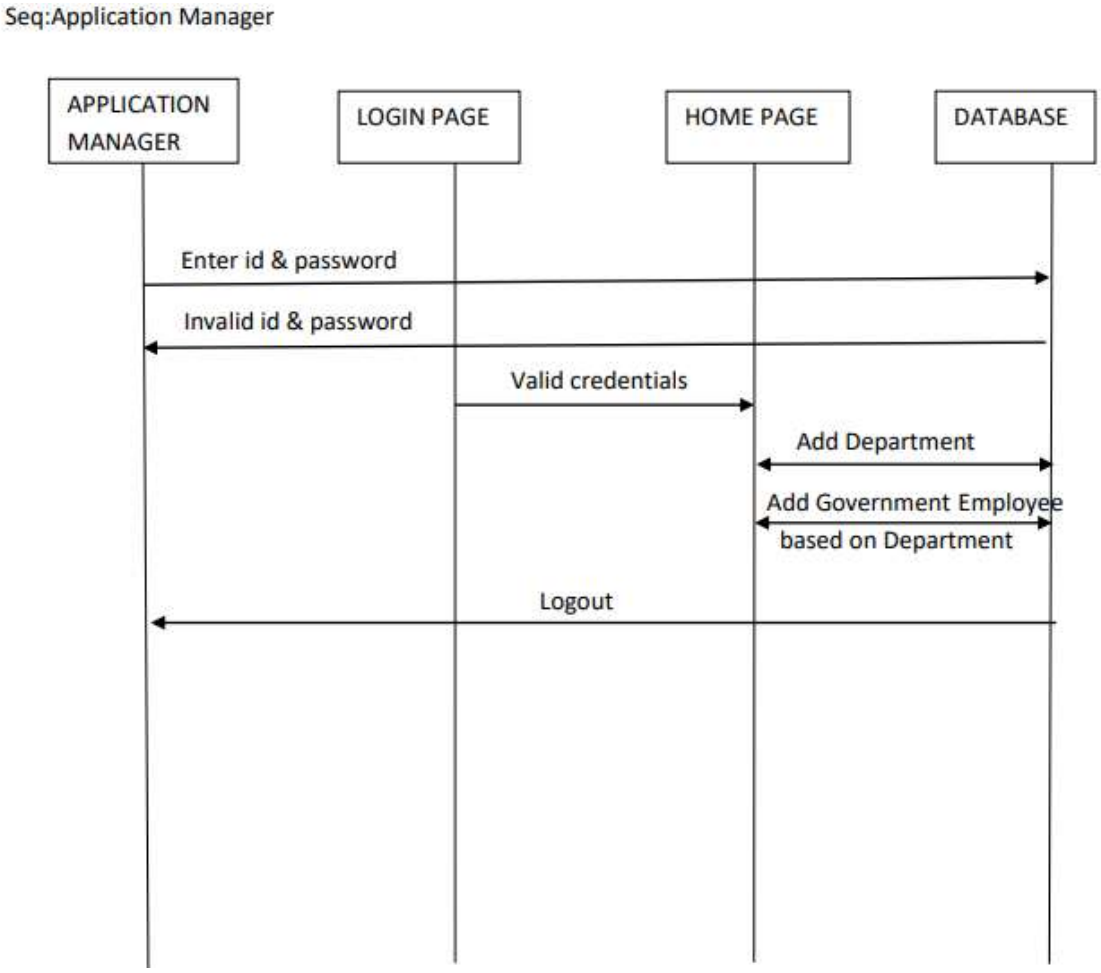



Login
Create Government Tender based on Department
View Apply Tender-Bidder Company-Ratings – AWS
Approve Tender
Logout
GOVERNMENT EMPLOYEE

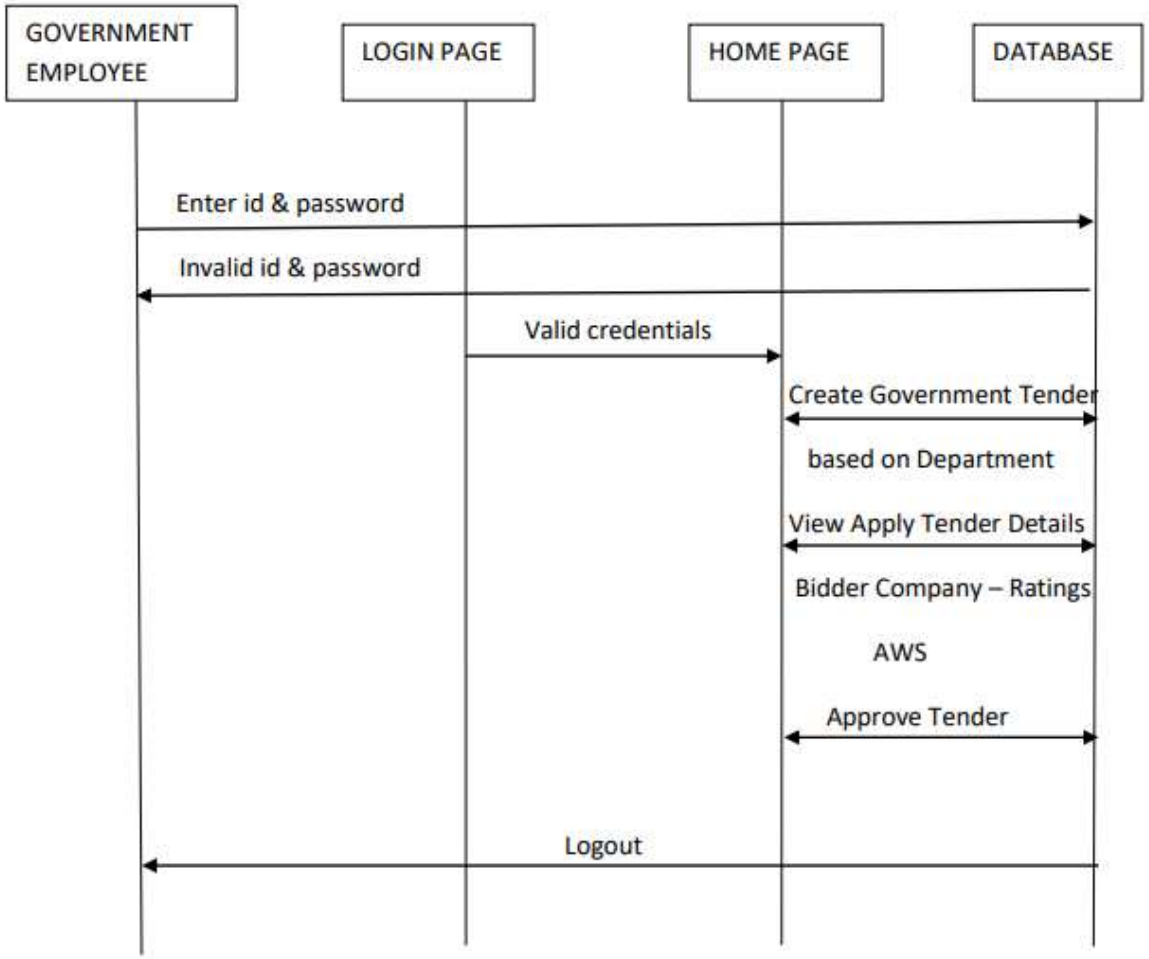

Login
Registration
Add Client Details
Add Client Project Details
Apply Government Tender AES-Encryption-AWS S3
Logout
BIDDER COMPANY



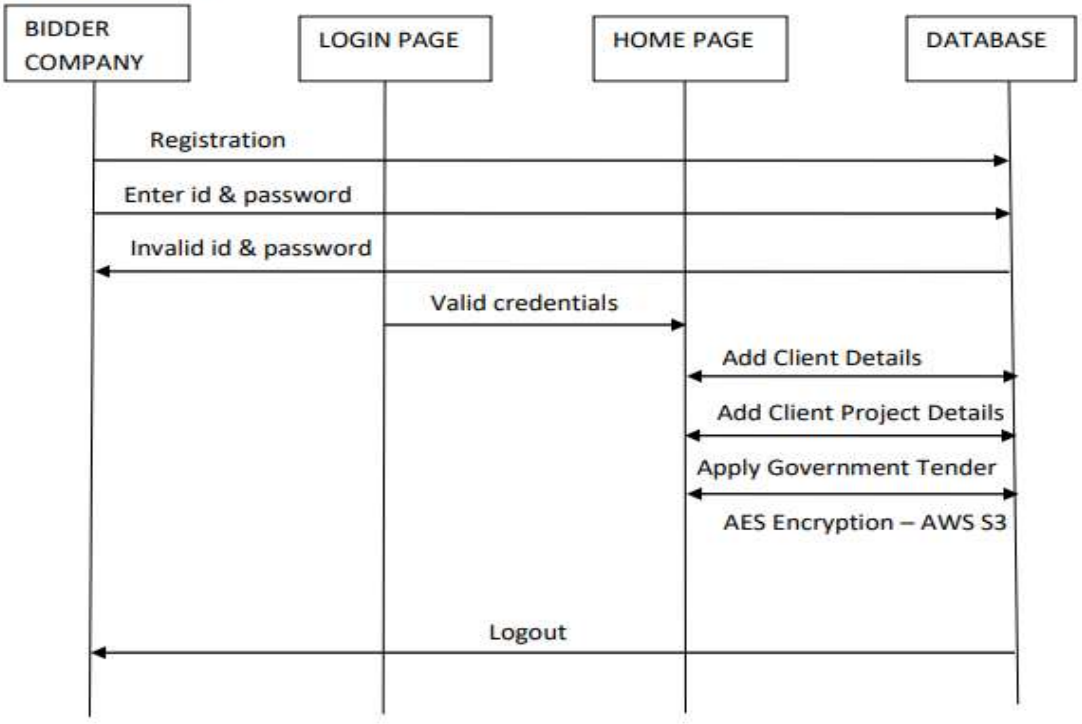
Sequence Diagram

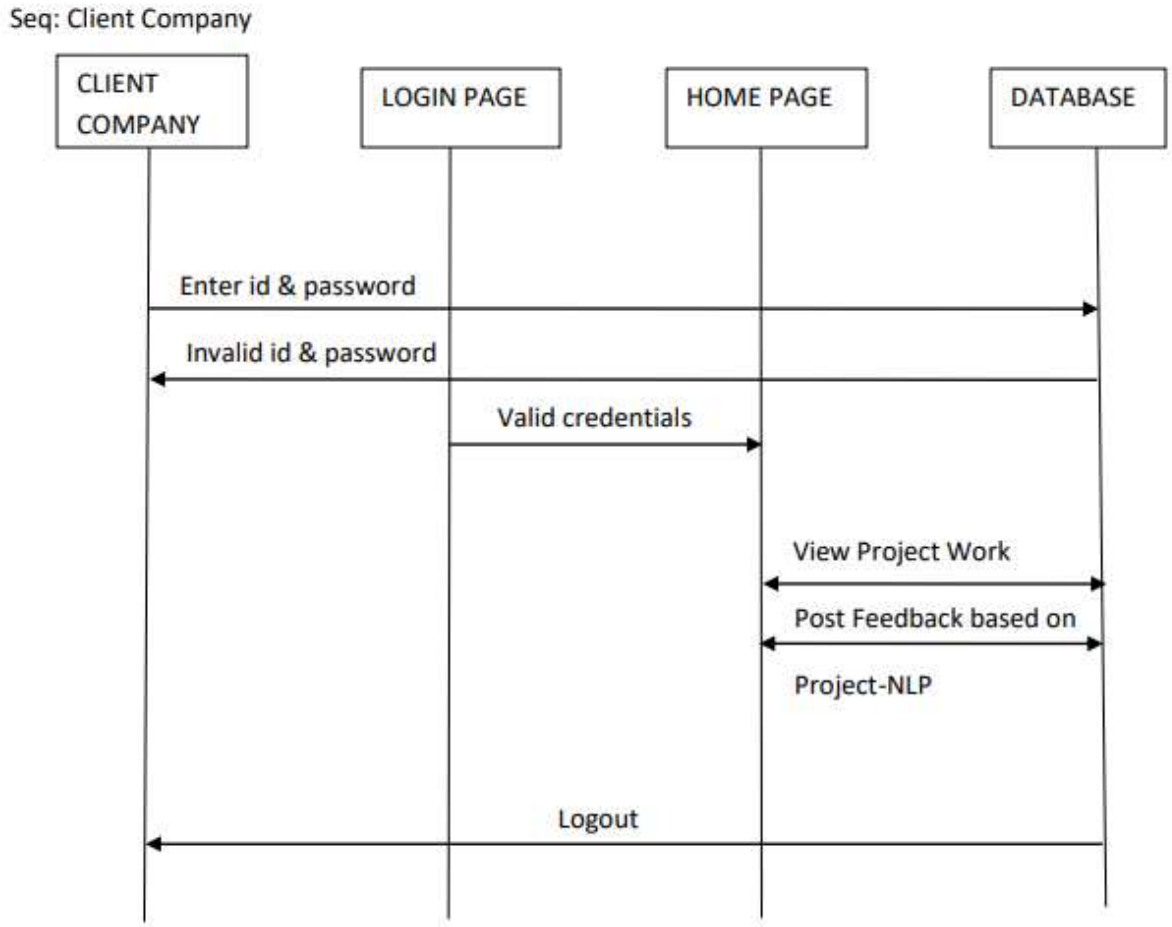


Seq: Government Employee



Seq: Bidder Company





Implementation

IMPLEMENTATION STEPS

Here is a revised version of the implementation process for A Secure Network Framework for Government Tender Allocation Based on AWS Service:

AWS Account Setup:

Create an AWS account if one does not already exist. This will provide the necessary infrastructure and services required for the system.

MySQL Database Configuration and Deployment:

Set up a MySQL database instance. Configure the required tables to store tender details, bidder details, client & client project details, tender quotation details and client feedback.

AES Encryption Integration:

Develop a module that utilizes the AES algorithm to encrypt tender quotation data. Integrate this module with the system to automatically encrypt the data before storing it in the QR code.

QR Code Generation and Storage:

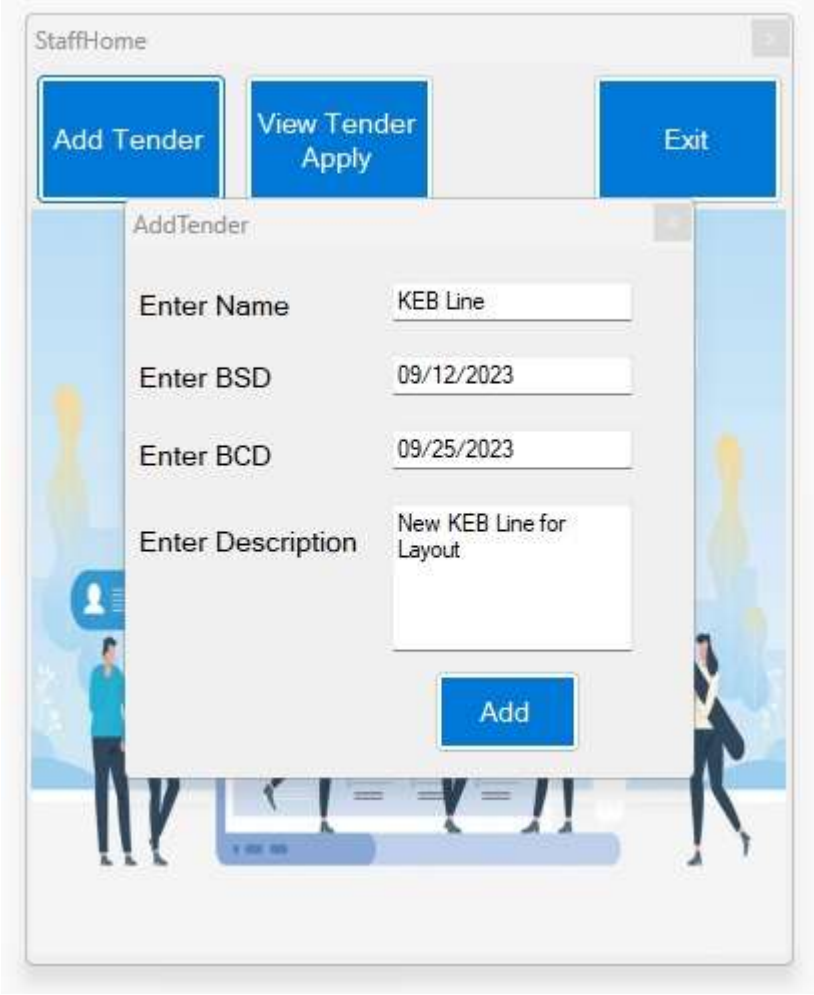
Implement a QR code generation module that converts the encrypted tender quotation data into a QR code image. Store the generated QR code images in an S3 bucket, ensuring appropriate access control mechanisms are in place.

Access Control Implementation:

Implement access control mechanisms that permit only authorized personnel to retrieve and decrypt vaccine ingredients data from the S3 bucket. This can be achieved by providing access keys only to authorized individuals or through the use of AWS Identity and Access Management (IAM) policies.

Tender Allocation Module:

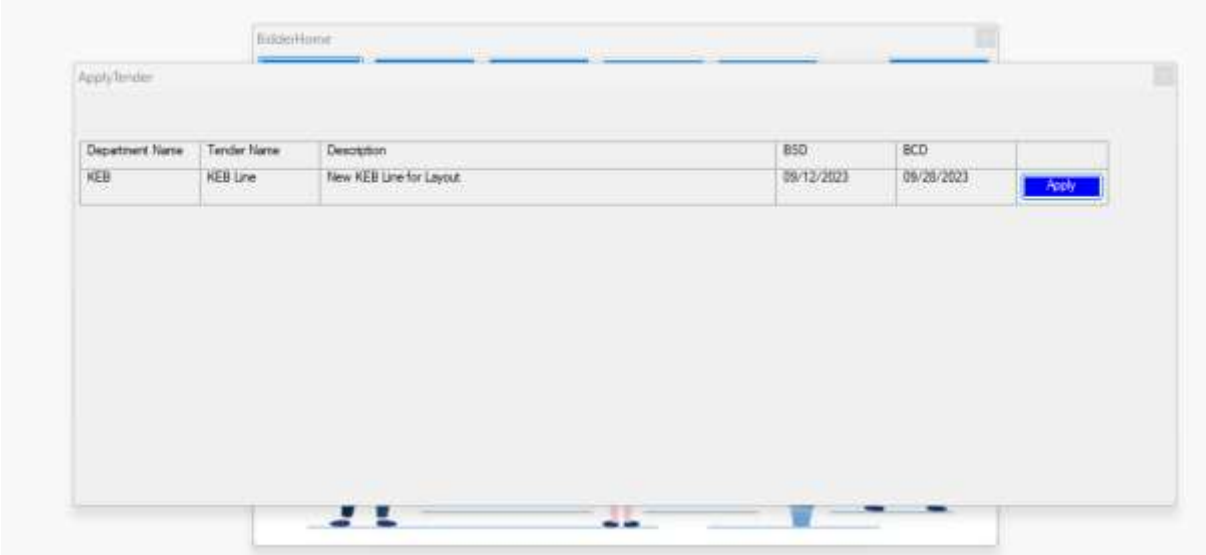
This module functions as an e-learning platform. The staff of a particular department initiates a new tender, and bidders apply for the tender by submitting their quotations and other details. The project utilizes AES algorithm for storing and encrypting the quotation details.



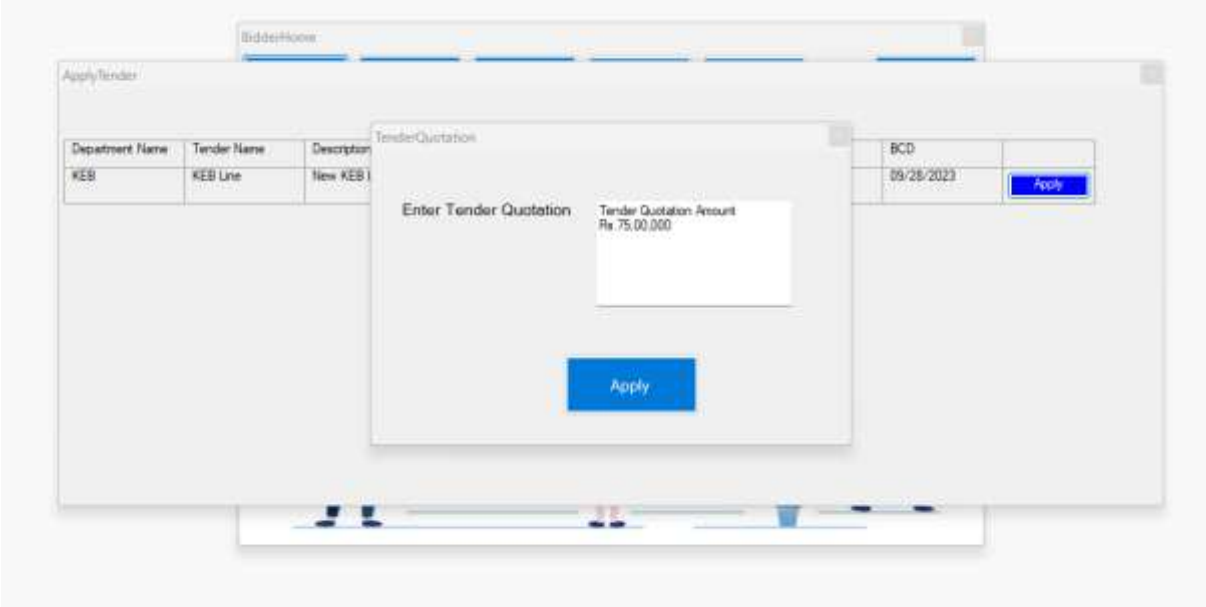
Government Staff create Tender

Bidder Approval Process:

The encrypted quotation is converted into a QR code image and stored in the AWS S3 bucket. Only authorized personnel can access the quotation details by entering the OTP sent to their email address. They can then approve or reject the bidder for the tender.



Bidder view Tender details & apply tender



Input Tender quotation amount, using AES Encryption algorithm encrypt tender quotation data, encrypt data write to QRCode image & in AWS S3 service store QRCode image. Using Shamir's algorithm split the encryption key & store in database along with AWS S3 QRCode image path.

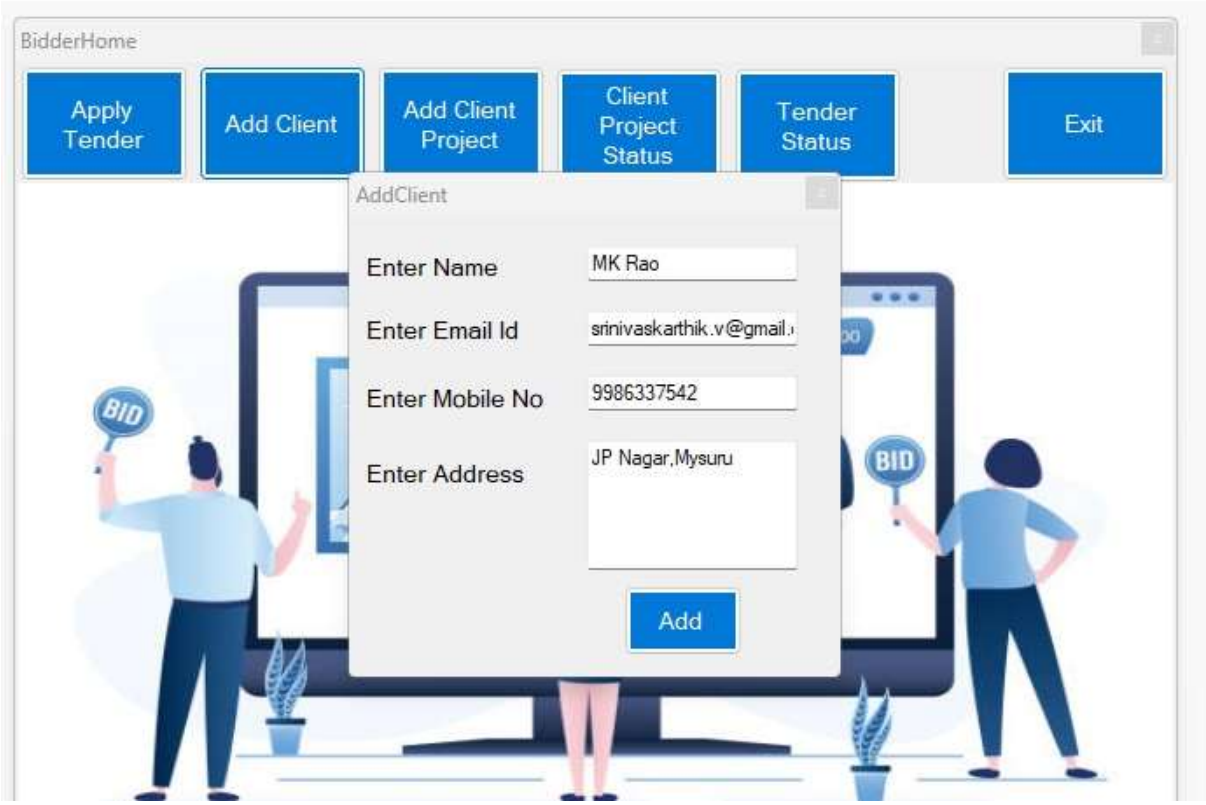
```
AmazonS3Client _s3ClientObj = null;
        static string filename;
        public static string url = "tcp://" + Login.ServerIP + ":5050/RemotingServer";
        GTService.MyConnection loginobj =
(GTService.MyConnection)Activator.GetObject(typeof(GTService.MyConnection), url);
        private void btnApply_Click(object sender, EventArgs e)
        {
```

```
        Random rnd = new Random();
        Shamir objsh = new Shamir();
        int key = rnd.Next(1000, 9999);
        string attributedata = objsh.AttributeValue(key);
        attributedata = attributedata.Remove(0, 1);
        string DataKey = attributedata;


        string EncryptData = AESCryptoClass.EncryptData(txtTQ.Text, key.ToString());

        Random rnd1 = new Random();
        var QCwriter = new BarcodeWriter();
        QCwriter.Format = BarcodeFormat.QR_CODE;
        var result = QCwriter.Write(EncryptData);
        string v = rnd1.Next(1000, 9999).ToString();
        filename = Login.UserID + "_" + v + ".jpg";

        string filepath = Application.StartupPath + "/File/" + filename;
        var barcodeBitmap = new Bitmap(result);

        using (MemoryStream memory = new MemoryStream())
        {
           using (FileStream fs = new FileStream(filepath,
              FileMode.Create, FileAccess.ReadWrite))
           {
              barcodeBitmap.Save(memory, ImageFormat.Jpeg);
              byte[] bytes = memory.ToArray();
              fs.Write(bytes, 0, bytes.Length);
           }
        }



        ////Amazon AWS
        // Set up your AWS credentials
        BasicAWSCredentials credentials = new BasicAWSCredentials("AKIA2PAQROQSYOJ52K4Z",
"DlyNZC62y0RQx1HSm+65xS7Dwfosd4xD1VDo6Mn3");

        // Create a new Amazon S3 client
        AmazonS3Client s3Client = new AmazonS3Client(credentials, Amazon.RegionEndpoint.USEast1);
        TransferUtility fileTransferUtility = new TransferUtility(s3Client);
        fileTransferUtility.Upload(Application.StartupPath + "/File/" + filename, "tender2023", filename);
        File.Delete(filepath);
        string _fpath = "tender2023" + "/" + filename;
        string res = loginobj.TenderApply(int.Parse(ApplyTender.TId), int.Parse(Login.UserID), DataKey,
_fpath);
        if (res == "1")
        {


           txtTQ.Text = "";
           MessageBox.Show("Tender Apply Successfully", "Message", MessageBoxButtons.OK,
MessageBoxIcon.Information);
```

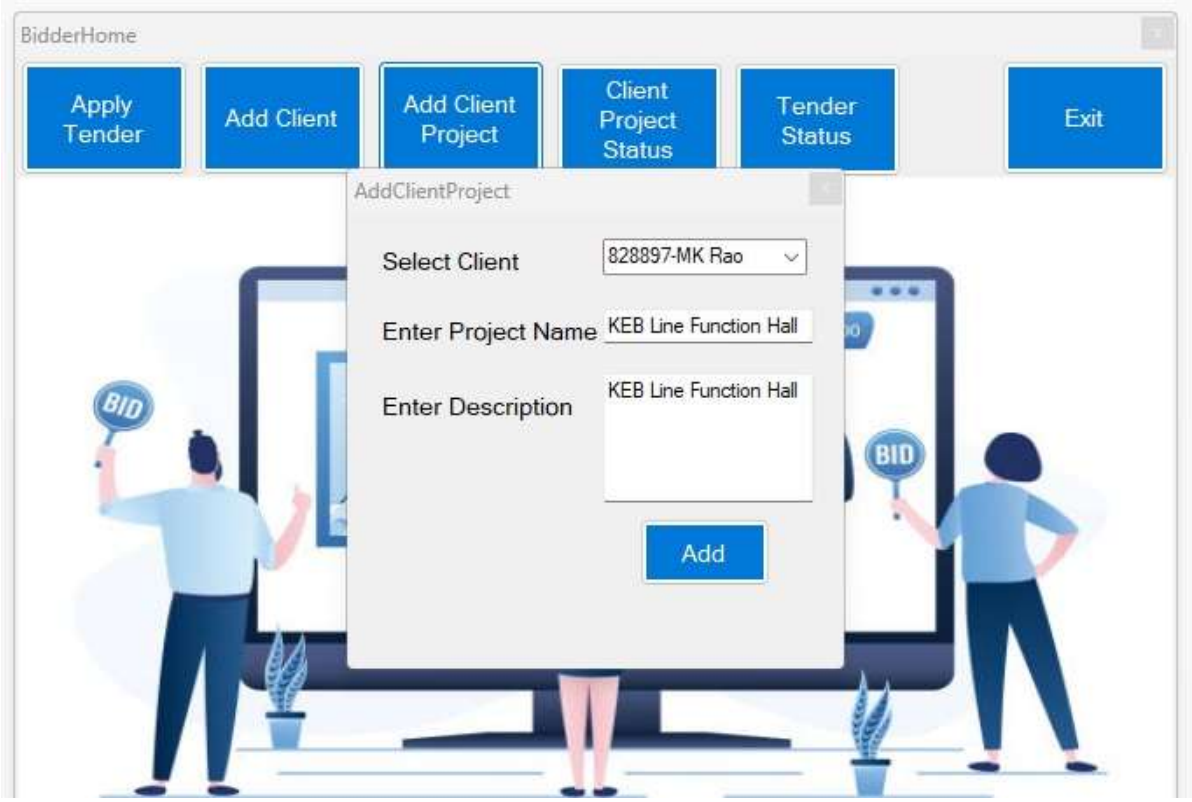
```
        }

        else if (res == "2")
        {
           // txtName.Text = txtCrimePlace.Text = "";
            MessageBox.Show("Tender Already Applied!!...", "Warning", MessageBoxButtons.OK,
MessageBoxIcon.Exclamation);

        }
```

Client Management and Feedback:

After approval, the bidding company can add clients and proceed with their work related to the tender. Once the work is completed, clients can provide feedback on the services. NLP concepts are used to convert client comments into ratings.



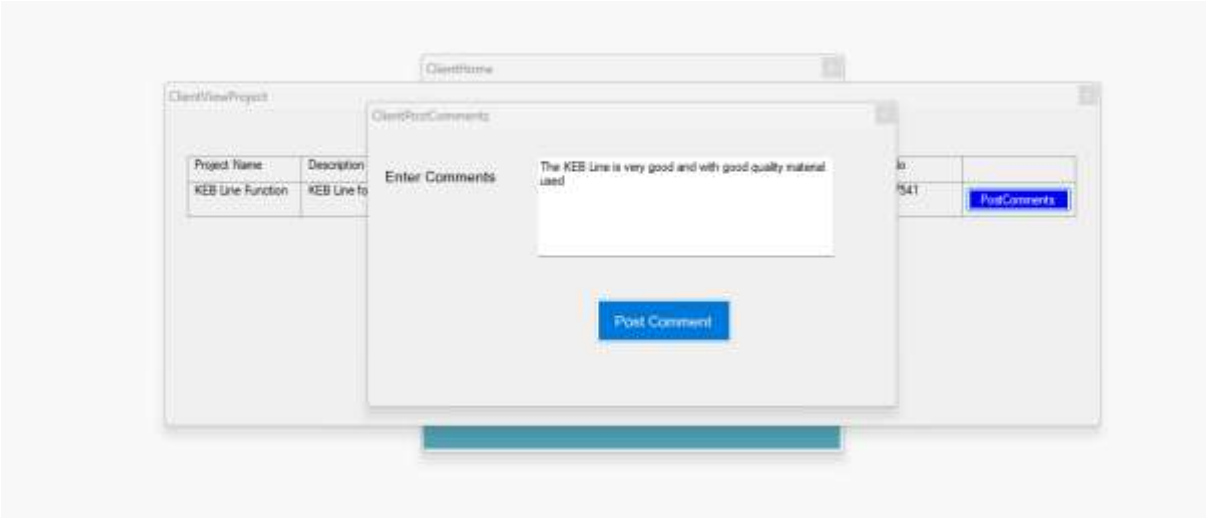
Add Client Details



Client Project Details

Rating and Bidding:

The ratings reflect on the bidder company profiles, which can be beneficial when bidding for future projects. This feedback system helps improve the overall bidding process and ensures a transparent.



Client Post Comments on Bidder Project work. The Comment process to NLP concept evaluate positive point & negative points.

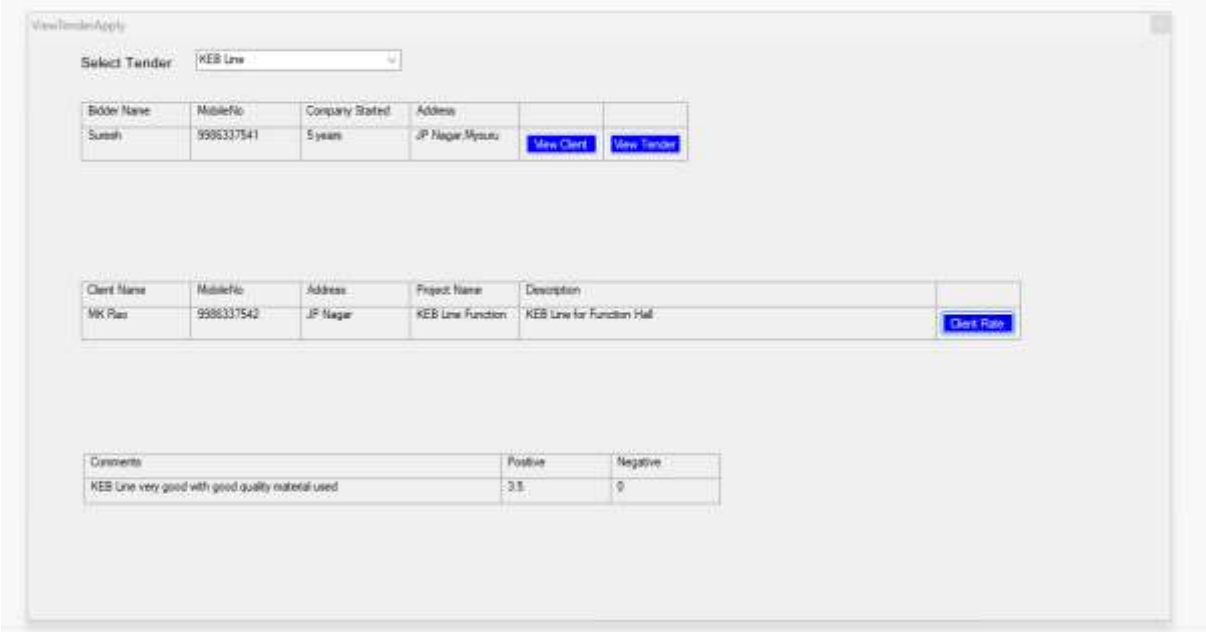
Natural Language Processing (NLP) is a field of artificial intelligence that focuses on the interaction between computers and human language. Its goal is to enable computers to understand and interpret human language in

a manner similar to humans. NLP involves various linguistic and cognitive concepts such as grammar, semantics, syntax, and pragmatics. NLP encompasses several fundamental tasks and activities. Tokenization involves breaking down text into smaller units like words or phrases. Part-of-speech tagging assigns grammatical labels to words in a sentence. Named Entity Recognition (NER) identifies and categorizes named entities like names of people, places, and dates. Sentiment analysis determines the emotional tone of text as positive, negative, or neutral. Other important tasks in NLP include machine translation for automated language translation, text classification for categorizing text.

The Client Company post comments on bidder project work. The Comments process to NLP in that filter articles, filter pronouns, filter prepositions, filter conjunctions & based on positive, negative keywords evaluate points i.e., calculate positive points & negative points in posted comments.

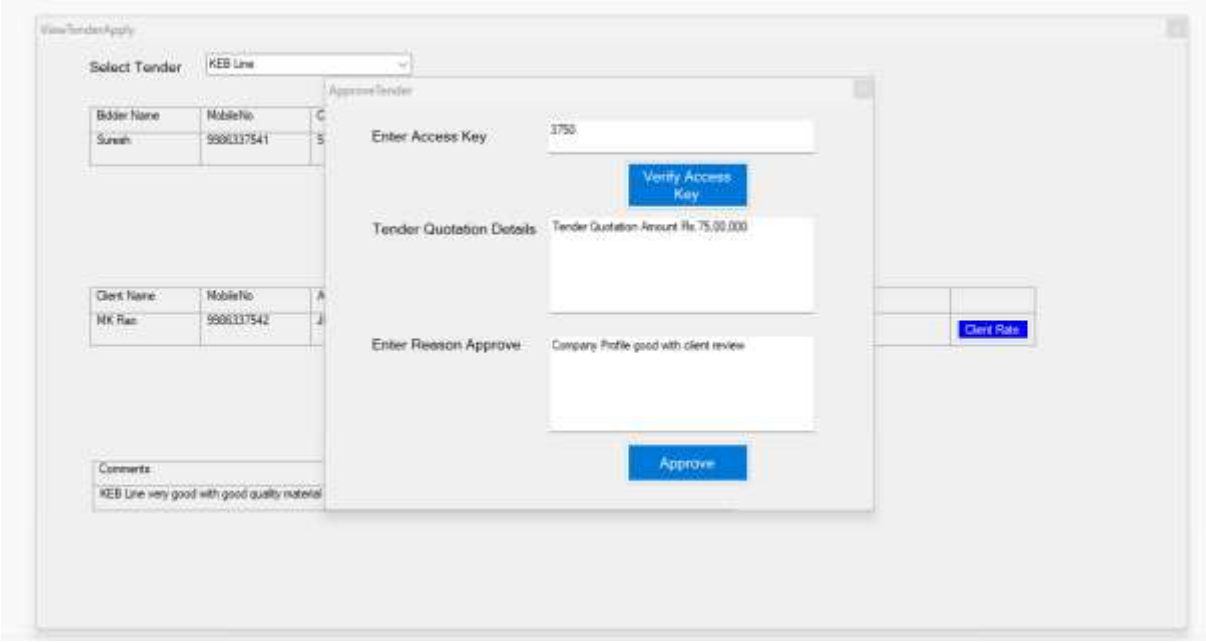
```
public string NLPmethod(string sentence)
    {
      NLP.Tokenizer tok = new NLP.Tokenizer();
      NLP.CleanSentence objcleansentence = new NLP.CleanSentence();
      string tokens = tok.btnGo_Click(sentence);
      NLP.CleanSentence cs = new NLP.CleanSentence();
      string articles = cs.RemoveArticles(tokens);
      string pronouns = cs.RemovePronouns(articles);
      string prepositions = cs.RemovePrepositions(pronouns);
      string conjunction = cs.RemoveConjunctions(prepositions);
      return conjunction;

    }

comment = NLPmethod(txtComments.Text.ToLower());
      NLP.PosNeg objposneg = new NLP.PosNeg();

      hneg = objposneg.Negativemethod();
      hpos = objposneg.Positiveemethod();
      foreach (DictionaryEntry d in hneg)
```

```
{
    string key = d.Key.ToString();
    if (comment.Contains(key))
    {
        nwordcount += 1;
        ncount += int.Parse(d.Value.ToString());
    }
}
float ntotalstars = ncount / nwordcount;
if (float.IsNaN(ntotalstars))
    ntotalstars = 0;

foreach (DictionaryEntry d in hpos)
{
    string key = d.Key.ToString();
    if (comment.Contains(key))
    {
        pwordcount += 1;
        pcount += int.Parse(d.Value.ToString());
    }
}
float ptotalstars = pcount / pwordcount;
if (float.IsNaN(ptotalstars))
    ptotalstars = 0;
```



Government staff view bidder details, client details, client project and client feedback

Government staff view tender quotation details & approve tender



Coding Part:

```
AmazonS3Client _s3ClientObj = null;
      static string filename;
      public static string url = "tcp://" + Login.ServerIP + ":5050/RemotingServer";
      GTService.MyConnection loginobj =
(GTService.MyConnection)Activator.GetObject(typeof(GTService.MyConnection), url);
      private void ApproveTender_Load(object sender, EventArgs e)
      {
         btnApprove.Enabled = false;
      }

      private void btnApprove_Click(object sender, EventArgs e)
```

```
        {
            string res = loginobj.UpdateTenderReason(int.Parse(ViewTenderApply.ATId), txtReason.Text);
            if (res == "1")
            {
                txtQuotation.Text = txtReason.Text = "";
                MessageBox.Show("Tender Approved Successfully", "Message", MessageBoxButtons.OK,
MessageBoxIcon.Information);

            }

            else
            {
                // txtName.Text = txtCrimePlace.Text = "";
                MessageBox.Show("Tender Approved Error!!...", "Warning", MessageBoxButtons.OK,
MessageBoxIcon.Exclamation);

            }
        }

        private void btnVAK_Click(object sender, EventArgs e)
        {
            if (txtAccessKey.Text == ViewTenderApply.Accesskey)
            {
                ////Amazon AWS
                btnApprove.Enabled = true;
                DataTable tab = new DataTable();
                tab = loginobj.GetBidderTender_ATId(int.Parse(ViewTenderApply.ATId));
                _s3ClientObj = new AmazonS3Client("AKIA2PAQROQSYOJ52K4Z",
"DlyNZC62y0RQx1HSm+65xS7Dwfosd4xD1VDo6Mn3", Amazon.RegionEndpoint.USEast1);

                string filepath = Application.StartupPath + "/File/" +
tab.Rows[0]["FilePath"].ToString().Split('/')[1];
                if (File.Exists(filepath))
                {
                    File.Delete(filepath);
                }
                GetObjectResponse _responseObj = _s3ClientObj.GetObject(new GetObjectRequest() {
BucketName = tab.Rows[0]["FilePath"].ToString().Split('/')[0], Key =
tab.Rows[0]["FilePath"].ToString().Split('/')[1] });
                _responseObj.WriteResponseStreamToFile(filepath);

                GetDecryptData objdk = new GetDecryptData();
                string result = objdk.GetData(tab.Rows[0]["KeyVal"].ToString());
                var QCreader = new BarcodeReader();
                string QCfilename = Path.Combine(filepath);
                var QCresult = QCreader.Decode(new Bitmap(QCfilename));
                txtQuotation.Text = AESCryptoClass.Decrypt(QCresult.Text, result);
            }
            else
            {
                // txtName.Text = txtCrimePlace.Text = "";
                MessageBox.Show("Invalid Access Key!!...", "Warning", MessageBoxButtons.OK,
MessageBoxIcon.Exclamation);
```

```
    }
}
```

TESTING

INTRODUCTION

Software testing serves several important objectives, which are as follows:

- Defect Detection: One of the primary objectives of software testing is to identify defects and errors that may have been introduced during the software development process.
- Defect Prevention: Testing helps in preventing defects by catching and fixing issues early in the development lifecycle, reducing the likelihood of defects reaching the end-users.
- Meeting Business and User Requirements: Testing ensures that the end result of the software aligns with the business and user requirements, ensuring that the software functions as intended.
- Compliance with Requirements: Software testing verifies that the software satisfies both the Business Requirement Specification (BRS) and the System Requirement Specifications (SRS).
- Customer Confidence: By providing a quality product through rigorous testing, software testing helps build customer confidence in the software's reliability and performance.

Software testing is conducted to confirm that the finished software behaves as expected based on the defined requirements and specifications. The ultimate goal is not to find every possible software bug, but to discover situations that could have adverse effects on users, usability, and maintainability. Various types of testing are performed at different levels, from module-level to application-level, depending on the purpose of testing and the software's specific requirements.

LEVELS OF TESTING

Unit Testing

Unit Testing is a crucial level of the software testing process where individual units or components of a software system are thoroughly tested. The primary objective is to verify that each unit of the software functions as intended.

Unit testing focuses on verifying the smallest unit of software design, which is the software component or module. It involves testing important control paths within the module to uncover any errors or issues. Unit testing is a type of white-box testing, where the internal structure and code implementation of the module are examined.

The unit testing process includes the following steps:

- Testing Module Interface: The module's interface is tested to ensure that data flows correctly into and out of the program during execution.
- Testing Local Data Structure: The local data structure of the module is tested to ensure that data stored temporarily remains intact during the execution.
- Boundary Condition Testing: Testing is performed to ensure that the module operates accurately at established boundaries or limits.
- Testing Independent Control Paths: All independent paths through the control structure are tested to ensure that all statements in the module are executed at least once.
- Error Handling Path Testing: The module's error-handling paths are tested to ensure that error scenarios are handled appropriately.

Unit testing is crucial as it helps identify and address issues at the granular level, ensuring the reliability and functionality of individual units before they are integrated into the larger system. This approach aids in delivering a high-quality and robust software product.

Integration Testing

Integration Testing is a crucial phase in the software testing process, where individual units are combined and tested as a group to identify faults in their interaction. This level of testing is a logical extension of unit testing. At its simplest form, two already-tested units are combined into a component, and their interface is thoroughly tested. A component, in this context, refers to an integrated aggregate of more than one unit. The primary objective is to test different combinations of units and gradually expand the process to include modules from other groups. Ultimately, all the modules making up a process are tested together. If any errors are detected during the combination of units, they are likely related to the interface between the units. This method simplifies the analysis process by reducing the number of possibilities.

In this software, the bottom-up integration testing approach has been utilized. This approach starts with the smallest and lowest-level modules and progressively proceeds one module at a time. For each module, tests were conducted, and the results were diligently recorded.

System Testing

System Testing is a critical stage in the software testing process where a fully integrated system or software is thoroughly tested. The primary objective of this test is to assess the system's compliance with the specified requirements.

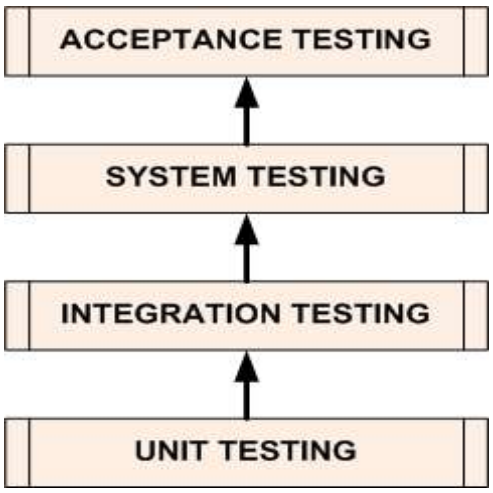
Acceptance Testing

Acceptance Testing, also known as User Acceptance Testing (UAT), is a significant phase in the software testing process aimed at assessing a system's acceptability. The primary purpose of this test is to

evaluate whether the system complies with the specified business requirements and is suitable for delivery.

User Acceptance Testing (UAT) is typically conducted by end-users or on their behalf to ensure that the software functions in accordance with the Business Requirement Document. UAT focuses on several key aspects:

- Ensuring all functional requirements are met.
- Verifying that all performance requirements are achieved.
- Assessing other aspects such as transportability, compatibility, and error recovery to ensure they meet the necessary criteria.
- Confirming that the acceptance criteria specified by the user are satisfactorily addressed.



TYPES OF TESTING

White box testing

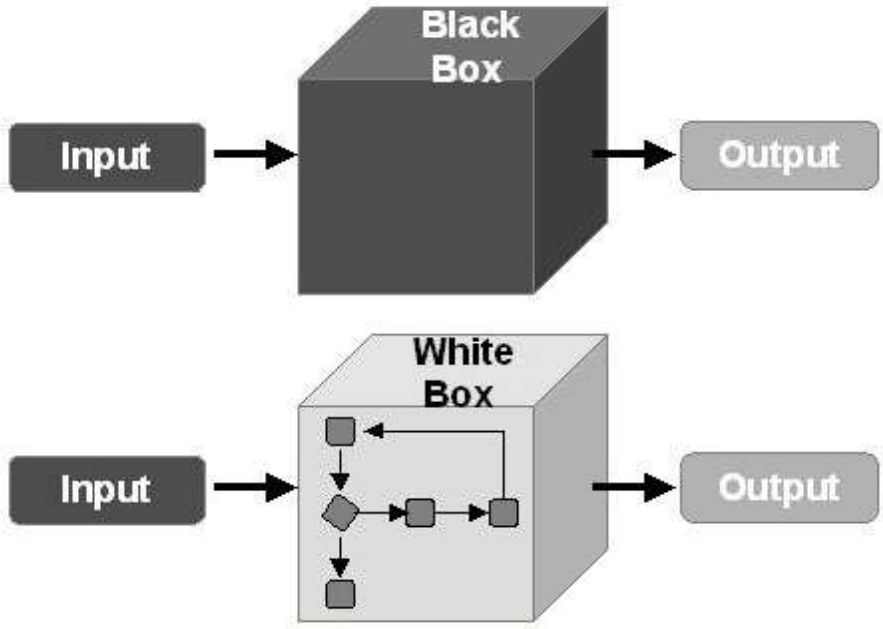
White-box testing is a software testing approach where the tester has access to the internal structure, design, and implementation details of the item being tested. This knowledge allows the tester to choose specific inputs to exercise various paths through the code and determine appropriate outputs. White-box testing requires programming expertise and deep implementation knowledge.

The name "white-box" comes from the analogy that the software program is like a transparent box, enabling the tester to see inside and understand its internal workings. This type of testing involves examining the internal software and code to ensure comprehensive coverage of code statements, branches, paths, and conditions. White-box testing is also known as structural testing or Glass box testing.

Black box testing

Black-box testing is a software testing technique that does not consider the internal system design. Instead, tests are based solely on requirements and functionality. The name "black box" comes from the analogy that the software program is like an opaque box, and the tester cannot see its internal workings.

During black-box testing, the internal mechanism of the system is ignored, and the focus is placed on the output generated in response to various inputs and the execution of the system. This approach is also referred to as functional testing, as it aims to ensure that the system functions correctly according to its specified requirements without delving into its internal implementation details.



Regression Testing

The purpose of regression testing is to verify that recent program or code changes have not negatively impacted existing features. It involves selecting either a full or partial set of previously executed test cases that are re-executed to ensure that the existing functionalities continue to work as expected. This testing is crucial to ensure that new code changes do not cause any unintended side effects on the existing functionalities and to confirm that the old code remains functional even after introducing new changes.

During the program development process, regression testing is a standard practice, often performed by specialized code testing experts in larger companies. Test department coders develop test scenarios and exercises, forming a test bucket to assess new units of code after they are written. Before releasing a new version of a software product, the old test cases are run against the new version to ensure that all the previous capabilities still function correctly. This precaution is essential because modifying or adding new code to a program can potentially introduce errors into sections of the code that were not intended to be changed.

TEST REPORTS

The users test the developed system when changes are made according to the needs. The testing phase involves the testing of the developed system using various kinds of data. An elaborate testing of data is prepared and system is tested using the test data. Test cases are used to check for outputs with different set of inputs.

Test Scenario 1: Enter proper credentials, login as Application Manager and check for successful login.

Test Case 1:

Step	Description	Input	Expected result	Actual result	Status
1	Open the application	N/A	'A Secure Network Framework for Government Tender Allocation Based On AWS Service' home page must be displayed	'A Secure Network Framework for Government Tender Allocation Based On AWS Service' homepage is displayed	Pass
2	Click on login	N/A	Login form must be displayed	Login form is displayed	Pass
3	Enter Application Manager Id and password,	Application Manager Id: Application	Navigate Successfully to Application Manager home page	Application Manager successfully navigated to home page	Pass

		Manager Id			

	Click on login	Password: 123			
4	Click on Add Department	Add Department	Department Details Should store in database & display successfully message	Department Details stored in database & display successfully message	Pass
5	Click on Add Employee based on Department	Add Employee based on Department	Add Employee based on Department Should store in database & display successfully message	Add Employee based on Department stored in database & display successfully message	Pass

Test Scenario 2: Enter proper credentials, login as Government Employee and check for successful login.

Test Case 2:

Step	Description	Input	Expected result	Actual result	Status
1	Open the application	N/A	'A Secure Network Framework for Government Tender Allocation Based On AWS Service' home page must be displayed	'A Secure Network Framework for Government Tender Allocation Based On AWS Service' homepage is displayed	Pass
2	Click on login	N/A	Login form must be displayed	Login form is displayed	Pass
3	Enter Employee Id and password,	Employee Id: Employee Id	Navigate Successfully to Employee home page	Employee successfully navigated to home page	Pass
	Click on login	Password: 123			

4	Click on Add Tender	Add Tender Details	Add Tender Details Should store in database & display successfully message	Add Tender Details stored in database & display successfully message	Pass
5	Click on Tender Bid	View Tender Bid Details	View Tender Bidder ,Client, Project Work & Client Feedback Details Should display successfully	View Tender Bidder ,Client, Project Work & Client Feedback Details displayed successfully	Pass
6	Click on Approve Tender	Approve Tender	View Tender Details – AES-AWS S3 & should be able to view Tender Quotation details Successfully & Approve Tender with Reason Successfully	View Tender Details- AES-AWS S3 & Successfully view Tender Quotation details & Approve Tender with Reason Successfully	Pass

Test Scenario 3: Enter proper credentials, login as Bidder Company and check for successful login.

Test Case 3:

Step	Description	Input	Expected result	Actual result	Status
1	Open the application	N/A	'A Secure Network Framework for Government Tender Allocation Based On AWS Service' homepage must be displayed	'A Secure Network Framework for Government Tender Allocation Based On AWS Service' home page is displayed	Pass
2	Click on Bidder Company Registration	Input basic Details	Based on Bidder Company Details should successfully register to application	Bidder Company Successfully register to application with details	Pass
3	Click on login	N/A	Login form must be displayed	Login form is displayed	Pass

4	Enter Bidder Company Id and password, Click on login	Bidder Company Id: Seed Company Password: 123	Navigate Successfully to Bidder Company home page	Bidder Company successfully navigated to home page	Pass
5	Click on Add Client & Client Project Details	Add Client & Client Project Details	Add Client & Client Project details should store in database & display successful message	Add Client & Client Project details stored in database & display successful message	Pass
6	Click on Update Project Status	Update Project Status	Update Project Status should update to database & display successful message	Update Project Status updated to database & display successful message	Pass
7	Click on Apply Tender	Bidder Id & Tender Quotation Amount	Tender Quotation Amount-AES-AWS S3 & File path store to database Successfully	Tender Quotation Amount – AES-AWS S3 & File Path stored to database Successfully	Pass

8	Click on View Tender Winner	View Tender Winner	Tender Winner Details Should display Successfully	Tender Winner Details display Successfully	Pass

Test Scenario 4: Enter proper credentials, login as Client Company and check for successful login.

Test Case 4:

Step	Description	Input	Expected result	Actual result	Status
1	Open the application	N/A	'A Secure Network Framework for Government Tender Allocation Based On AWS Service' home page must be displayed	'A Secure Network Framework for Government Tender Allocation Based On AWS Service' home page is displayed	Pass
2	Click on login	N/A	Login form must be displayed	Login form is displayed	Pass

3	Enter Client Id and password, Click on login	Client Id: Client Id Password: 123	Navigate Successfully to Client home page	Client successfully navigated to home page	Pass

4	Click on Post Comment to Project Work	Post Comment Details	Post Comment details (NLP Based) should store in database & display successful message	Post Comment details (NLP Based) stored in database & display successful message	Pass

Chapter- 5 Conclusion

Conclusion:

The successful implementation of our system marks a momentous achievement in revolutionizing the government tender assignment process. Throughout this project, we have highlighted the significance of AES Encryption and AWS S3 Service, showcasing their undeniable benefits in ensuring robust security and efficient data management. By creating a comprehensive end-to-end automation solution, we have taken significant strides towards optimizing the government tender workflow.

The adoption of AES Encryption algorithm stands as a crucial step in safeguarding the integrity and confidentiality of sensitive tender information. AES Encryption has played a pivotal role in fortifying our security measures, thwarting unauthorized access and ensuring that only qualified and deserving bidders gain access to tender projects. This has enhanced the transparency of the government tendering system and fostered healthy competition among bidder companies.

The integration of AWS S3 Service has transformed the storage and retrieval of data in our system. Leveraging cloud computing capabilities, AWS S3 Service has improved accessibility, scalability, and

redundancy. By reducing reliance on local infrastructure, we have minimized the risk of data loss and system downtime, ensuring the continuity and reliability of our automated workflow.

Our primary objective was to optimize the government tender process, benefiting both the government and participating bidder companies. Through the incorporation of AES Encryption and AWS S3 Service, we have created a streamlined and fortified system, amplifying the overall profitability and efficiency of government tenders. The enhanced security measures protect the government's interests, while the merit-based selection process empowers bidder companies to showcase their capabilities and secure opportunities that align with their expertise.

In conclusion, the successful implementation of AES Encryption and AWS S3 Service has brought a paradigm shift to the government tendering process. Our end-to-end automation has simplified the workflow, reduced manual intervention, and streamlined operations. By deploying a secure and transparent selection mechanism, we have elevated the credibility and trustworthiness of the entire process, instilling confidence among both government entities and bidders alike.

As we look to the future, we envision further advancements in our system, embracing cutting-edge technologies to continuously enhance the efficiency and security of government tender assignments. The collaboration between technology and governance has never been more vital, and our project serves as a testament to the transformative power of innovation in shaping a stronger and more progressive society. With AES Encryption and AWS S3 Service at its core, our system represents a beacon of progress, fostering growth and prosperity through a fair and secure tendering landscape.

5.3 Future Scope of our study

- It can be further expanded to include more features and activities of E- governance to solve other issues also
- Can able to track the complete details of the tender work status

REFERENCES:

[1]. Vikas Hassija, Vinay Chamola, Dara Nanda Gopala Krishna, Neeraj Kumar, and Mohsen Guizani “A Blockchain and Edge Computing-based Secure Framework for Government Tender Allocation” IEEE 2020

[2]. Dhawal Mali, Divya Mogaveera, Parth Kitawat, Mohd.Jawwad “Blockchain-based e-Tendering System ” IEEE 2020

[3]. Pranamya Jain, Siddhesh Gangan, Siddhesh Rane, Yash Jain, and Dhiren Patel “A Stake-Based Blockchain Solution for Tender Procurement System” 2019

[4]. C. D. Clack, V. A. Bakshi, and L. Braine, “Smart contract templates: essential requirements and design options,” arXiv preprint arXiv:161204496, 2016.

[5]. J. A. Jaoude and R. G. Saade, “Blockchain applications–usage in different domains,” IEEE Access, vol. 7, pp. 45 360–45 381, 2019.

[6]. H. Hou, “The application of blockchain technology in e-government in china,” in 2017 26th International Conference on Computer Communication and Networks (ICCCN). IEEE, 2017, pp. 1–4. [7]. B. K. Mohanta, S. S. Panda, and D. Jena, “An overview of smart contract and use cases in blockchain technology,” in 2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT), July 2018, pp. 1–4.

[8]. Ambegaonker, Ajeenkkya, Utkarsh Gautam, and Radha Krishna Rambola. "Efficient approach for Tendering by introducing Blockchain to maintain Security and Reliability." 2018 4th International Conference on Computing Communication and Automation (ICCCA). IEEE, 2018.

[9]. Betts, Martin, et al. "Towards secure and legal e-tendering." Journal of Information Technology in Construction 11 (2006): 89- 102

[10]. Wood, Gavin. "Ethereum: A secure decentralised generalised transaction ledger." Ethereum project yellow paper 151.2014 (2014): 1-32.

[11]. McDermott P (2010) Building open government. Gov. Inf. Q. 27(4):401–413. Special Issue: Open/Transparent Government

[12]. J. Huang, L. Kong, G. Chen, M.-Y. Wu, X. Liu, and P. Zeng, “Towards secure industrial iot: Blockchain system with credit-based consensus mechanism,” IEEE Transactions on Industrial Informatics, 2019