

**University of Lucknow**



**Goel Institute of Higher Studies**

**Mahavidyalaya**

**College Code: 1164**

**Summer Training Project Report**

“**GENARLIZED BOT FOR STOCKS**”

**UNDER GUIDANCE OF SUBMITTED BY**

**Mr. Arun Singh Yadav**

Assistant Professor, Project Guide,

Department of Computer Science,

G.I.H.S. Mahavidyalaya,

**Ashish kumar Shukla**

2111640580017

**Manshi Shukla**

2111640580034

**Neha Dubey**

211640580037

Department of Computer Science,

G.I.H.S. Mahavidyalaya

B.C.A. Vth Semester

December 2023



**University of Lucknow**



**Goel Institute of Higher Studies**

**Mahavidyalaya**

**College Code: 164**

**Summer Training Project Report**

“”

**UNDER GUIDANCE OFSUBMITTED BY**

**Mr. Arun Singh Yadav**

Assistant Professor, Project Guide,

Department of Computer Science,

G.I.H.S. Mahavidyalaya,

**Manshi Shukla**

2111640580034

**Ashish kumar Shukla**

2111640580017

**Neha Dubey**

211640580037

Department of Computer Science,

G.I.H.S. Mahavidyalaya

B.C.A. Vth Semester

December 2023



**University of Lucknow**



**Goel Institute of Higher Studies**

**Mahavidyalaya**

**College Code: 164**

**Summer Training Project Report**

“**STUDENT GRIEVANCE SYSTEM**”

**UNDER GUIDANCE OFSUBMITTED BY**

**Mr. Arun Singh Yadav**

Assistant Professor, Project Guide,

Department of Computer Science,

G.I.H.S. Mahavidyalaya,

**Ashish kumar Shukla**

2111640580017

**Manshi Shukla**

2111640580034

**Neha Dubey**

211640580037

Department of Computer Science,

G.I.H.S. Mahavidyalaya

B.C.A. Vth Semester

December 2022



**University of Lucknow**



**Goel Institute of Higher Studies**

**Mahavidyalaya**

**College Code: 164**

**Summer Training Project Report**

“**STUDENT GRIEVANCE SYSTEM**”

**UNDER GUIDANCE OFSUBMITTED BY**

**Mr. Arun Singh Yadav**

Assistant Professor, Project Guide,

Department of Computer Science,

G.I.H.S. Mahavidyalaya,

**Neha Dubey**

2111640580037

**Manshi Shukla**

2111640580034

**Ashish kumar Shukla**

2111640580017

Department of Computer Science,

G.I.H.S. Mahavidyalaya

B.C.A. Vth Semester

December 2023



**DEPARTMENT OF COMPUTER SCIENCE**

**G.I.H.S. MAHAVIDYALAYA**

**CERTIFICATE**

This is to certify that the summer training project entitled **“Generalized Bot For Stock”** has been developed by the team comprising of **Ashish kumar Shukla, Manshi Shukla** and **Neha Dubey** under my supervision in for the partial fulfillment for the award of degree “**Bachelor of Computer Application (BCA)”** for the session 2022-23.

**Mr. Arun Singh Yadav**

Assistant Professor,Project Guide

Department of B.C.A ,

G.I.H.S.Mahavidyalaya,

Lucknow

**Dr. Santosh Pandey**

Principal,

G.I.H.S.Mahavidyalaya,

Lucknow

**CERTIFICATEPAGE**



any organization. If the organization is an academic institution, then this issue

becomes more sensitive and important. Students are the mostvulnerable entities at

educational institutions often fail to express and sometimes fail to seek proper

support for the issues they face arising at numerous levels. Thus, on analyzing the

prevailing state of redressed mechanisms of grievances at some of the prestigious

colleges of Madhya Pradesh, it came as a revelation that none of them had a

completely formulated grievance redressal mechanism to address the arising con-

ﬂicts in the lives of the students. In wake of the above-mentioned problem as an

implication, a prototype of grievance redressal has been worked out which could

comply well with the solution provision for the arising conﬂicts for students. In this

paper, we focus on the development and the execution of the above-mentioned

prototype which could be incorporated to adhere to the grievance redressal for

students. This paper puts deep insight into incorporating all those problem are

any organization. If the organization is an academic institution, then this issue

becomes more sensitive and important. Students are the mostvulnerable entities at

educational institutions often fail to express and sometimes fail to seek proper

support for the issues they face arising at numerous levels. Thus, on analyzing the

prevailing state of redressed mechanisms of grievances at some of the prestigious

colleges of Madhya Pradesh, it came as a revelation that none of them had a

completely formulated grievance redressal mechanism to address the arising con-

ﬂicts in the lives of the students. In wake of the above-mentioned problem as an

implication, a prototype of grievance redressal has been worked out which could

comply well with the solution provision for the arising conﬂicts for students. In this

paper, we focus on the development and the execution of the above-mentioned

prototype which could be incorporated to adhere to the grievance redressal for

students. This paper puts deep insight into incorporating all those problem areas

which were found on the basis of the analysi**PREFACE**

**PREFACE**

A grievance is a discontent or dispute which could arise at any level in any organization. If the organization is an academic institution, then this issue becomes more sensitive and important. Students are the most vulnerable entities at educational institutions often fail to express and sometimes fail to seek proper support for the issues they face arising at numerous levels. In colleges if anyone wants to complaint about something we need to write it down on a paper and submit it in complaint box. So for solving this problem we are creating a ‘Online student grievance cell’ for Student Grievances. Here anyone related to college or any other person can just easily login and submit his complaint. This complaint letter will be solved by respective Principal or Admin of this system. In this system the facility of managing User and their data related to their Complaints and their profile is also easy managed. This data can be easily viewed and modified whenever required. This Complaint Management System also provides the functionality of providing a monthly report which will used to maintain information about how many complaints are solved in a particular month. It also provides the validation of Users.

**Ashish kumar Shukla**

2111640580017

**Manshi Shukla**

2111640580034

**Neha Dubey**

211640580037

Department of Computer Science,

G.I.H.S. Mahavidyalaya

**ACKNOWLEDEGEMENT**

We would like to express my special thanks of gratitude to our principal sir **Dr. Santosh Pandey** as well as our project guide**Mr. Arun Singh Yadav** who gave us the golden opportunity to do this wonderful project which also helped us in doing a lot of research and we come to know about so many new things. We are really thankful to them.

Secondly, we would like to thank my teachers **Mr virendra Singh, Mr. Apoorva ,Mr. K.K.Shukla, and Mr. shivansh ojha, Mr. Hemant Pandey, Mrs .Priya Dixit** who helped us a lot in finalizing this project within the limited time frame.

Also, we would like to thank our **Parents** for moral support and Seniors  **Vishal,**and **Himanshu verma**for their Support in this project.

We are making this project to increase our knowledge.

Thanks again to all who helped me.

WithRegards

Ashish kumar Shukla

2111640580017

Manshi Shukla

2111640580034

Neha Dubey

211640580037

Department of Computer Science,

G.I.H.S. Mahavidyalaya

**TABLE OF CONTENT**

|  |  |  |
| --- | --- | --- |
| Serial | Content | Page No |
|  | Title Page |  |
|  | Certification of Originality |  |
|  | Certification of Summer Training |  |
|  | Abstract/Preface |  |
|  | Acknowledgement |  |
|  | Title of Figure |  |
|  |  |  |
| 1. | **[ INTRODUCTION ]** | **1** |
|  |  |  |
| 1.1 | Introduction | 1 |
| 1.2 | Background | 3 |
| 1.3 | Purpose | 3 |
| 1.4 | Existing System | 4 |
|  |  |  |
| 2. | **[ OBJECTIVE AND ADVANTAGES ]** | **5** |
|  |  |  |
| 2.1 | Objective and Scope of Proposed System | 6 |
| 2.2 | Advantages Of Proposed System | 7 |
|  |  |  |
| 3 | **[ TOOLS AND CONFIGURATION ]** | **8** |
|  |  |  |
| 3.1 | Software Requirement | 9 |
| 3.2 | Hardware Requirement | 9 |
|  |  |  |
| 4 | **[ TECHNOLOGY USED ]** | **10** |
|  |  |  |
| 4.1 | Editor( Visual Studio Code) | 11 |
| 4.2 | Working Technology(HTML ,CSS ,Java Script ,Python(django as a framework) ,MS SQL | 13 |
| 4.3 | SSMS | 20 |
|  |  |  |
| 5 | **[ SOFTWARE ENGINEERING MODEL USED ]** | **22** |
|  |  |  |
| 5.1 | Iterative model | 23 |
|  |  |  |
| 6 | **[ SYSTEM ANALYSIS ]** | **26** |
|  |  |  |
| 6.1 | Investigation Phase | 27 |
| 6.2 | Software Requirement Specification | 28 |
|  |  |  |
| 7 | **[ FEASIBILITY STUDY ]** | **29** |
|  |  |  |
| 7.1 | Operational Feasibility | 30 |
| 7.2 | Technical Feasibility | 31 |
| 7.3 | Economical Feasibility | 31 |
|  |  |  |
| 8 | **[ CONCEPTUAL MODULES** ] | **32** |
|  |  |  |
| 8.1 | Data flow Diagram | 33 |
| 8.2 | Entity Relationship Diagram | 36 |
|  |  |  |
| 9 | **[ MODULES ]** | **40** |
|  |  |  |
| 9.1 | Student Registration module | 41 |
| 9.2 | Student Login module | 41 |
| 9.3 | Status module | 41 |
| 9.4 | Admin register module | 41 |
| 9.5 | Admin login module | 41 |
| 9.6 | Committee member module | 41 |
| 9.7 | Grievance module | 41 |
|  |  |  |
| 10 | **[ SNAPSHOTS AND CODING ]** | **42** |
|  |  |  |
| 11 | **[ TESTING ]** | **49** |
|  |  |  |
| 11.1 | Black box testing | 50 |
| 11.2 | White box testing | 50 |
| 11.3 | Unit testing | 50 |
| 11.4 | Integration testing | 51 |
| 11.5 | System testing | 51 |
|  |  |  |
| 12 | **[ FUTURE SCOPE ]** | **52** |
|  |  |  |
| 13 | **[ CONCLUSION ]** | **54** |
|  |  |  |
| 14 | **[ BIBLIOGRAPHY ]** | **56** |
|  |  |  |

A grievance is a discontent or dispute which could arise at any level in

any organization. If the organization is an academic institution, then this issue

becomes more sensitive and important. Students are the mostvulnerable entities at

educational institutions often fail to express and sometimes fail to seek proper

support for the issues they face arising at numerous levels. Thus, on analyzing the

prevailing state of redressed mechanisms of grievances at some of the prestigious

colleges of Madhya Pradesh, it came as a revelation that none of them had a

completely formulated grievance redressal mechanism to address the arising con-

ﬂicts in the lives of the students. In wake of the above-mentioned problem as an

implication, a prototype of grievance redressal has been worked out which could

comply well with the solution provision for the arising conﬂicts for students. In this

paper, we focus on the development and the execution of the above-mentioned

prototype which could be incorporated to adhere to the grievance redressal for

students. This paper puts deep insight into incorporating all those problem areas

which were found on the basis of the

A grievance is a discontent or dispute which could arise at any level in

any organization. If the organization is an academic institution, then this issue

becomes more sensitive and important. Students are the mostvulnerable entities at

educational institutions often fail to express and sometimes fail to seek proper

support for the issues they face arising at numerous levels. Thus, on analyzing the

prevailing state of redressed mechanisms of grievances at some of the prestigious

colleges of Madhya Pradesh, it came as a revelation that none of them had a

completely formulated grievance redressal mechanism to address the arising con-

ﬂicts in the lives of the students. In wake of the above-mentioned problem as an

implication, a prototype of grievance redressal has been worked out which could

comply well with the solution provision for the arising conﬂicts for students. In this

paper, we focus on the development and the execution of the above-mentioned

prototype which could be incorporated to adhere to the grievance redressal for

students. This paper puts deep insight into incorporating all those problem areas

which were found on the basis of the analysis ph

A grievance is a discontent or dispute which could arise at any level in

any organization. If the organization is an academic institution, then this issue

becomes more sensitive and important. Students are the mostvulnerable entities at

educational institutions often fail to express and sometimes fail to seek proper

support for the issues they face arising at numerous levels. Thus, on analyzing the

prevailing state of redressed mechanisms of grievances at some of the prestigious

colleges of Madhya Pradesh, it came as a revelation that none of them had a

completely formulated grievance redressal mechanism to address the arising con-

ﬂicts in the lives of the students. In wake of the above-mentioned problem as an

implication, a prototype of grievance redressal has been worked out which could

comply well with the solution provision for the arising conﬂicts for students. In this

paper, we focus on the development and the execution of the above-mentioned

prototype which could be incorporated to adhere to the grievance redressal for

students. This paper puts deep insight into incorporating all those problem areas

which were found on the basis of the analysis ph

A grievance is a discontent or dispute which could arise at any level in

any organization. If the organization is an academic institution, then this issue

becomes more sensitive and important. Students are the mostvulnerable entities at

educational institutions often fail to express and sometimes fail to seek proper

support for the issues they face arising at numerous levels. Thus, on analyzing the

prevailing state of redressed mechanisms of grievances at some of the prestigious

colleges of Madhya Pradesh, it came as a revelation that none of them had a

completely formulated grievance redressal mechanism to address the arising con-

ﬂicts in the lives of the students. In wake of the above-mentioned problem as an

implication, a prototype of grievance redressal has been worked out which could

comply well with the solution provision for the arising conﬂicts for students. In this

paper, we focus on the development and the execution of the above-mentioned

prototype which could be incorporated to adhere to the grievance redressal for

students. This paper puts deep insight into incorporating all those problem areas

which were found on the basis of the analysis ph

A grievance is a discontent or dispute which could arise at any level in

any organization. If the organization is an academic institution, then this issue

becomes more sensitive and important. Students are the mostvulnerable entities at

educational institutions often fail to express and sometimes fail to seek proper

support for the issues they face arising at numerous levels. Thus, on analyzing the

prevailing state of redressed mechanisms of grievances at some of the prestigious

colleges of Madhya Pradesh, it came as a revelation that none of them had a

completely formulated grievance redressal mechanism to address the arising con-

ﬂicts in the lives of the students. In wake of the above-mentioned problem as an

implication, a prototype of grievance redressal has been worked out which could

comply well with the solution provision for the arising conﬂicts for students. In this

paper, we focus on the development and the execution of the above-mentioned

prototype which could be incorporated to adhere to the grievance redressal for

students. This paper puts deep insight into incorporating all those problem areas

which were found on the basis of the analysis ph

**INTRODUCTION**

**GENERALIZED BOT FOR STOCKS**

* 1. **INTRODUCTION**
  2. **FUNCTION**

**Stocks management** is the practice of ordering, storing, tracking, and controlling inventory.

The purpose of stock management software is to maintain an optimal stock level, track goods during transport between locations, receive new items, manage warehouse processes such as picking, packing, and shipping, prevent product obsolescence and spoilage, and ensure your products are never out of stock.

Since we know that a company or an organization or any firm who imports some product in their stock and use them in their company to produce a finished (or new ) product. or a firm who imports a number of products in his company and sell it in many parts but there is not a generalized bot for stock system by this, there is no record of product in the stock **Generalized Bot For Stocks** is a system.Similarly this website bot for stock is the online facility to managed the stock inventory system.

|  |
| --- |
|  |

* 1. **BACKGROUND**

The development of a bot for stocks involves a multidisciplinary approach, combining elements of finance, programming, data analysis, and artificial intelligence. Here's a brief background on the key aspects involved in a bot for stocks project:

1. **Understanding Financial Markets:**
   * A foundational understanding of financial markets, including stocks, bonds, commodities, and derivatives, is crucial. Developers need to grasp concepts such as market dynamics, trading strategies, risk management, and market indicators.
2. **Programming Languages:**
   * Proficiency in programming languages is essential for building a stock bot. Common languages for this purpose include Python, Java, C++, and others. Python is particularly popular due to its extensive libraries for data analysis and machine learning.
3. **Data Sources and APIs:**
   * Access to real-time and historical market data is critical. Developers often use financial APIs (Application Programming Interfaces) provided by stock exchanges, financial news services, and data providers to fetch data for analysis and decision-making.
4. **Risk Management:**
   * Effective risk management is crucial in trading. Developers need to build features that help users manage risk, including setting stop-loss orders, position sizing, and other risk control mechanisms.
5. **User Interface (UI) and User Experience (UX):**
   * The bot's interface should be user-friendly, providing a seamless experience for traders and investors. A well-designed UI allows users to input parameters, receive alerts, and visualize data easily.
6. **Security:**
   * Given the sensitivity of financial data, security is paramount. Developers must implement robust security measures to protect user information and ensure the integrity of transactions.
7. **Backtesting:**
   * Before deploying a trading bot in the live market, it's crucial to conduct thorough backtesting. This involves testing the bot's strategy against historical data to evaluate its performance and refine parameters.
8. **Continuous Improvement and Maintenance:**

The financial markets are dynamic, and trading strategies may need adjustments over time. Developers should build mechanisms for continuous improvement, updates, and maintenance of the bot.

* 1. **PURPOSE**

In purpose system, we have design a three user interfaces system which will manage the security and it will provide AI based feature for the user .So that the inventory system of any company or organization will be easy to maintain and convenient to use.

So the interface to make AI based inventory management system we have focused on the following.

* The system will help to keep record the location of stored stocks.
* Automatic alert in the situation of under-stocks.
* Inventory for rejected product so that LRM (line rejected note ) can be managed .

**1.5 EXISTING SYSTEM**

An Inventory Management System (IMS) is a system used by businesses and organizations to efficiently track, manage, and control their inventory or stock of products, materials, and goods. It plays a crucial role in ensuring that an organization has the right amount of inventory on hand to meet customer demand while minimizing carrying costs, preventing stock-outs, and reducing the risk of overstocking. Here is some background information on Inventory Management Systems:

* System keep record of all these product details and streamlines and automate all the processes with respect to **historical prospective.**
* The primary objectives of an Inventory Management System include optimizing stock levels, reducing holding costs, ensuring product availability, minimizing the risk of stockouts, and preventing overstock situations that tie up capital.
* IMS can manage different types of inventory, including raw materials, work-in-progress (WIP), finished goods, spare parts, and consumables.

.

**OBJECTIVES**

**AND**

**ADVANTAGES**

**2.BJECTIVES AND SCOPE OF PROPOSED SYSTEM**

**2.1.2 Objectives:**

To manage and organize required WIP(work in progress ) for an industrial firm.So that we can have Acknowledgment and holistic information for a particular inventory .By this project we can have following advantage -

* **Inventory optimization** - Inventory optimization is a critical aspect of supply chain management that involves efficiently managing and controlling the inventory levels of products within an organization. The primary goal is to strike a balance between ensuring product availability to meet customer demand and minimizing holding costs.
* **Cost reduction (such as holding cost and maintaining )-** Cost reduction is a critical aspect of business management aimed at decreasing expenses and improving overall profitability. Implementing effective cost reduction strategies requires a comprehensive approach and careful consideration of various aspects of the business.
* **Improve cash flow-** Improving cash flow in the context of inventory management involves optimizing processes related to purchasing, holding, and selling inventory.
* **Order fulfillment**- Order fulfillment refers to the process of receiving, processing, and delivering customer orders. It involves various stages, from the placement of an order to the delivery of the product or service to the customer. Efficient order fulfillment is crucial for customer satisfaction and can significantly impact a company's reputation
* **Security maintain** - Security management is a critical aspect when developing and operating a bot for stocks, especially given the sensitive nature of financial information and the potential impact of security breaches.
* **Reduction of dead stock**- Dead stock, also known as excess inventory or obsolete inventory, refers to products that have not been sold and are unlikely to be sold in the future. Managing and reducing dead stock is essential for improving cash flow, minimizing holding costs, and optimizing overall inventory performance.
* Automate routine tasks, such as reorder point calculations and order generation, to reduce manual effort and minimize human errors.

**2.1.2 Scope:**

The future scope of bots for stocks is likely to be influenced by ongoing advancements in technology, changes in financial markets, and evolving user expectations. Here are several potential trends and areas of development for bots in the stock market:

1. **Artificial Intelligence and Machine Learning:**
   * Continued integration of advanced AI and machine learning algorithms for more sophisticated analysis, pattern recognition, and predictive modeling.
   * AI-driven bots may evolve to dynamically adapt to market conditions and improve decision-making capabilities.
2. **Natural Language Processing (NLP):**
   * Enhanced natural language processing capabilities for improved communication between users and bots.
   * Bots may become more conversational, understanding and responding to user queries and instructions in a more human-like manner.
3. **Algorithmic Trading Evolution:**
   * Further development of algorithmic trading strategies, including high-frequency trading algorithms, to capitalize on market inefficiencies.
   * Integration of deep learning techniques for more complex and adaptive trading strategies.
4. **Risk Management and Compliance:**
   * Improved risk management features, including better tools for assessing and managing portfolio risk.
   * Enhanced compliance functionalities to ensure that bots adhere to regulatory requirements.
5. **Blockchain Integration:**
   * Increased integration of blockchain technology for transparent and secure transactions.
   * Development of decentralized finance (DeFi) bots that operate on blockchain platforms.
6. **Personalization and User Experience:**
   * More personalized user experiences, with bots providing tailored investment recommendations based on individual user profiles and preferences.

User-friendly interfaces designed to make financial information and investment strategies more accessible to a wider audience.

**2.2ADVANTAGES OF PROPOSED SYSTEM**

* A purpose-built system or bot for stocks offers several advantages for investors, traders, and financial professionals. Here are some key benefits:
* **Automated Trading:**
* **Efficiency:** Automated execution of trades based on predefined criteria eliminates the need for manual intervention, allowing for faster and more efficient trading.
* **Consistency:** Bots can execute trades consistently and without emotional bias, adhering strictly to predefined rules.
* Market Analysis and Insights:
* **Data Processing:** Bots can rapidly process large volumes of financial data, conducting complex analyses and providing insights that might be impractical for humans to perform manually.
* **Real-time Monitoring**: Bots can continuously monitor market conditions in real time, identifying trends, patterns, and potential opportunities or risks.
* **Risk Management:**
* **Dynamic Adjustments:** Bots can dynamically adjust trading strategies based on real-time market conditions and risk parameters.
* **Stop-Loss and Take-Profit Orders:** Automated placement of stop-loss and take-profit orders helps manage risk and protect investments.
* **Historical Testing:** Bots allow for backtesting of trading strategies using historical data, helping users evaluate strategy performance under different market conditions.
* **Optimization:** Users can refine and optimize trading strategies based on historical performance, improving the likelihood of success in future trades.
* **Time Efficiency:**
* **24/7 Monitoring:** Bots can operate 24/7, monitoring markets around the clock and reacting to market changes in real time.
* **Instantaneous Execution:** Automated execution ensures that trades are executed instantly when predetermined conditions are met.
* **Diversification:**
* **Multi-Asset Management:** Bots can handle the simultaneous management of multiple assets, enabling diversification across various financial instruments.
* **Portfolio Balancing:** Bots can automatically rebalance portfolios to maintain desired asset allocations.
* **Reduced Human Error:**
* **Elimination of Emotional Decision-Making:** Automated systems remove the emotional aspects of trading, preventing impulsive decisions and minimizing the impact of human error.
* **Consistent Execution:** Bots execute trades consistently according to predefined rules, reducing the risk of errors caused by fatigue or cognitive biases.
* **Cost Efficiency:**
* **Lower Transaction Costs:** Automated trading can lead to lower transaction costs due to the elimination of manual order placement.
* **Operational Efficiency:** Bots reduce the need for extensive human resources in monitoring and executing trades.
* **Speed of Execution:**
* **High-Frequency Trading (HFT):** Bots can execute trades at speeds far beyond

**Tools**

**AND**

**Configuration**

**SOFTWARE AND HARDWARE REQUIREMENT**

**3.1SOFTWARE REQUIREMENT**

|  |  |
| --- | --- |
| Programmer | Client |
| * Python (Django as a frame work) * MS SQL * Visual Studio Code * MS Word 2010 | Windows 10, Windows 11 |

**3.2 HARDWAREREQUIREMENT**

|  |  |
| --- | --- |
| Programmer | Client |
| * Intel 3.0 ghz or higher processor * 4 GB RAM * 20 GB HDD Space | * Intel 3.0 ghz or higher processor * 2 GB RAM * 2 GB HDD Space |

**TECHNOLOGY**

**USED**

**4.1 EDITOR**

**Visual Studio Code** Visual Studio Code is a lightweight but powerful source code editor which runs on your desktop and is available for Windows, macOS and Linux. It comes with built-in support for JavaScript, TypeScript and Node.js and has a rich ecosystem of extensions for other languages (such as C++, C#, Java, Python, PHP, Go) and runtimes (such as .NET and Unity).

Features include support for [debugging](https://en.wikipedia.org/wiki/Debugging), [syntax highlighting](https://en.wikipedia.org/wiki/Syntax_highlighting), [intelligent code completion](https://en.wikipedia.org/wiki/Intelligent_code_completion), [snippets](https://en.wikipedia.org/wiki/Snippet_(programming)), [code refactoring](https://en.wikipedia.org/wiki/Code_refactoring), and embedded [Git](https://en.wikipedia.org/wiki/Git). Users can change the [theme](https://en.wikipedia.org/wiki/Theme_(computing)), [keyboard shortcuts](https://en.wikipedia.org/wiki/Keyboard_shortcut), preferences, and install [extensions](https://en.wikipedia.org/wiki/Plug-in_(computing)) that add additional functionality.

|  |
| --- |
| [Visual Studio Code 1.35 icon.svg](https://en.wikipedia.org/wiki/File:Visual_Studio_Code_1.35_icon.svg) |
| [Visual Studio Code Insiders running on Windows 10](https://en.wikipedia.org/wiki/File:VS_Code_(Insiders).png) |
| Visual Studio Code Insiders running on Windows 10   |  |  | | --- | --- | | Visual Studio Code Insiders running on Windows 10 | | | [**Developer(s)**](https://en.wikipedia.org/wiki/Software_developer) | [Microsoft](https://en.wikipedia.org/wiki/Microsoft) | | **Initial release** | April 29, 2015; 5 years ago | | [**Stable release**](https://en.wikipedia.org/wiki/Software_release_life_cycle) | 1.53.2 (11 February 2021; 10 days ago) | | [**Preview release**](https://en.wikipedia.org/wiki/Software_release_life_cycle#BETA) | 1.53.0-insider | | [**Repository**](https://en.wikipedia.org/wiki/Repository_(version_control)) | * [github.com/microsoft/vscode](https://github.com/microsoft/vscode) | | **Written in** | [TypeScript](https://en.wikipedia.org/wiki/TypeScript), [JavaScript](https://en.wikipedia.org/wiki/JavaScript), [HTML](https://en.wikipedia.org/wiki/HTML), and [CSS](https://en.wikipedia.org/wiki/CSS) | | [**Operating system**](https://en.wikipedia.org/wiki/Operating_system) | [Windows 7](https://en.wikipedia.org/wiki/Windows_7) or later, [OS X 10.10](https://en.wikipedia.org/wiki/OS_X_10.10) or later, [Linux](https://en.wikipedia.org/wiki/Linux) | | | [**Platform**](https://en.wikipedia.org/wiki/Computing_platform) | [IA-32](https://en.wikipedia.org/wiki/IA-32), [x86-64](https://en.wikipedia.org/wiki/X86-64), [ARM64](https://en.wikipedia.org/wiki/ARM64) | | [**Size**](https://en.wikipedia.org/wiki/File_size) | * Windows: 40.8–68.3 MB * Linux: 46.5–66.6 MB * macOS: 67.5 MB | | **Available in** | English (US), Simplified Chinese, Traditional Chinese, French, German, Italian, Portuguese (Brazil), Japanese, Korean, Russian, Spanish | | [**Type**](https://en.wikipedia.org/wiki/Software_categories#Categorization_approaches) | [Source code editor](https://en.wikipedia.org/wiki/Source_code_editor), [debugger](https://en.wikipedia.org/wiki/Debugger) | | [**License**](https://en.wikipedia.org/wiki/Software_license) | * **Source code:**[MIT License](https://en.wikipedia.org/wiki/MIT_License) * **Binaries built by Microsoft:**[Proprietary software](https://en.wikipedia.org/wiki/Proprietary_software) | | **Website** | [code.visualstudio.com](https://code.visualstudio.com) | |

**4.2WORKING TECHNOLOGY**

**HTML:**



HTML or Hyper Text Markup Language is the standard markup language used to create web pages. HTML was created in 1991 by Tim Berners-Lee at CERN in Switzerland. It was designed to allow scientists to display and share their research. HTML is written in the form of HTML elements consisting of tags enclosed in angle brackets (like <html>). HTML tags most commonly come in pairs like <h1> and </h1>, although some tags represent empty elements and so are unpaired, for example <img>. The first tag in a pair is the start tag, and the second tag is the end tag (they are also called opening tags and closing tags).

|  |
| --- |
| **HTML Structure**  <html>  <head>  <title>Page title</title>  </head>  <body>  <h1>This is a heading</h1>  <p>This is a paragraph.</p>  <p>This is another paragraph.</p>  </body>  </html> |

The purpose of a web browser (Chrome, Edge, Firefox, Safari) is to read HTML documents and display them correctly.

A browser does not display the HTML tags, but uses them to determine how to display the document:

HTML elements form the building blocks of all websites. HTML allows images and objects to be embedded and can be used to create interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. It can embed scripts written in languages such as Java Script which affect the behavior of HTML web pages. HTML is descriptive markup language. Library of various markup languages is defined in various browser.

**CSS:**



CSS tutorial or CSS 3 tutorial provides basic and advanced concepts of CSS technology. Our CSS tutorial is developed for beginners and professionals. The major points of CSS are given below:

• CSS stands for Cascading Style Sheet.

• CSS is used to design HTML tags.

• CSS is a widely used language on the web.

• HTML, CSS and JavaScript are used for web designing. It helps the web designers to apply

Style on HTML tags.

Cascading Style Sheets (CSS) is a style sheet language used for describing the look and formatting of a document written in a markup language. While most often used to style web pages and user interfaces written in HTML and XHTML, the language can be applied to any kind of XML document, including plain XML, SVG and XUL. CSS is a cornerstone specification of the web and almost all web pages use CSS style sheets to describe their presentation.

CSS is designed primarily to enable the separation of document content from document presentation, including elements such as the layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple pages to share formatting, and reduce complexity and repetition in the structural content (such as by allowing for table less web design).

CSS can also allow the same markup page to be presented in different styles for different rendering methods, such as on-screen, in print, by voice (when read out by a speech-based browser or screen reader) and on Braille-based, tactile devices. It can also be used to allow the web page to display differently depending on the screen size or device on which it is being viewed. While the author of a document typically links that document to a CSS file, readers can use a different style sheet, perhaps one on their own computer, to override the one the author has specified. With plain HTML you define the colors and sizes of text and tables throughout your pages. If you want to change a certain element you will therefore have to work your way through the document and change it. With CSS you define the colors and sizes in "styles". Then as you write your documents you refer to the styles. Therefore: if you change a certain style it will change the look of your entire site. Another big advantage is that CSS offers much more detailed attributes than plain HTML for defining the look and feel of your site.

**JAVASCRIPT:**



JavaScript (JS) is a dynamic computer programming language. It is most commonly used as part of web browsers, whose implementations allow client-side scripts to interact with the user, control the browser, communicate asynchronously, and alter the document content that is displayed. It is also being used in server-side network programming (with Node.js), game development and the creation of desktop and mobile applications.

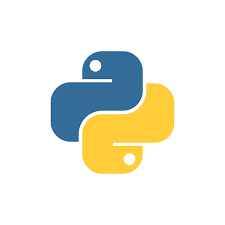
JavaScript is a prototype-based scripting language with dynamic typing and has first-class functions. Its syntax was influenced by C. JavaScript copies many names and naming conventions from Java, but the two languages are otherwise unrelated and have very different semantics. The key design principles within JavaScript are taken from the self and Scheme programming languages. It is a multi-paradigm language, supporting object-oriented, imperative, and functional programming styles.

The application of JavaScript in use outside of web pages—for example, in PDF documents, site-specific browsers, and desktop widgets is also significant. Newer and faster JavaScript VMs and platforms built upon them (notably Node.js) have also increased the popularity of JavaScript for server-side web applications. On the client side, JavaScript was traditionally implemented as an interpreted language but just-in-time compilation is now performed by recent (post-2012) browsers. JavaScript was formalized in the ECMA Script language standard and is primarily used as part of a web browser (client-side JavaScript). This enables programmatic access to objects within a host environment.

JavaScript is the most popular programming language in the world. It is the language for HTML, for the Web, for computers, servers, laptops, tablets, smart phones, and more. You can use JavaScript to:

* Change HTML elements
* Delete HTML elements
* Create new HTML elements
* Copy and clone HTML elements

**PYTHON**



Python is a high-level, general-purpose programming language known for its simplicity, readability, and versatility. It was created by Guido van Rossum and first released in 1991. Guido van Rossum starts working on Python at the Centrum Wiskunde & Informatica (CWI) in the Netherlands. The first Python version, Python 0.9.0, is released in February 1991.

Python has gained popularity for its readability, simplicity, and versatility. It has become one of the most widely used programming languages, particularly in fields such as web development, data science, machine learning, and automation. The commitment to maintaining backward compatibility and the strong Python community contribute to its ongoing success.

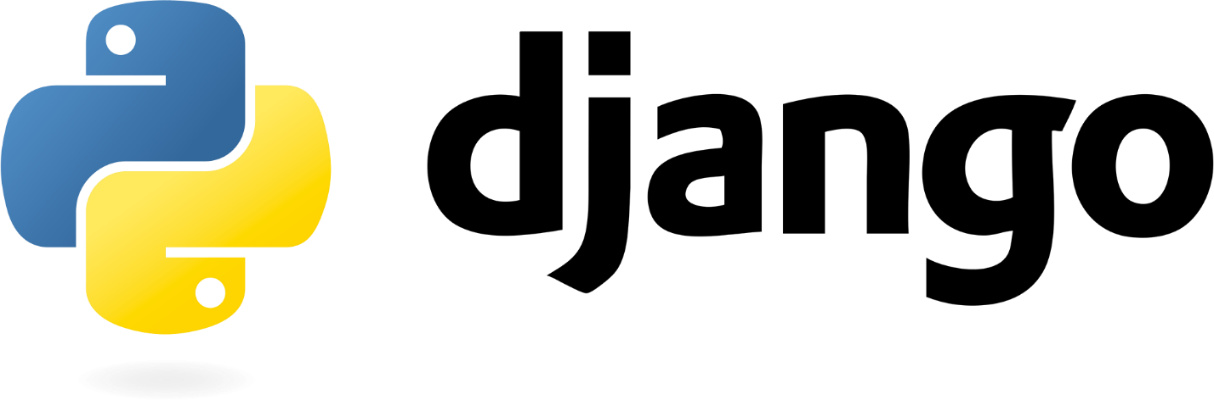
**Advantages of python-**

Using Python as a backend language for web development has several advantages, contributing to its popularity in this domain.

1. **Django and Flask Frameworks:**
   * Python has powerful web frameworks like Django and Flask that simplify and streamline the web development process. Django, in particular, follows the "Don't Repeat Yourself" (DRY) principle and encourages rapid development with a clean, pragmatic design.
2. **Readability and Maintainability:**
   * Python's clean and readable syntax helps developers write code that is easy to understand and maintain. This is crucial for large and complex web applications where multiple developers may collaborate.
3. **Extensive Libraries and Modules:**
   * Python's standard library and third-party packages provide a wide range of tools and utilities for web development. This reduces the need to write code from scratch and accelerates development.
4. **Community Support:**
   * Python has a large and active community of developers, including a dedicated web development community. This community support ensures that developers have access to resources, tutorials, and solutions to common problems.
5. **Scalability:**
   * Python can scale well for web applications, and its frameworks provide features and best practices to handle scalability challenges. As the application grows, Python-based web frameworks offer tools for managing increased traffic and data.
6. **Database Support:**
   * Python supports various databases, both relational and non-relational. ORM (Object-Relational Mapping) tools like Django's ORM simplify database interactions and make it easy to switch between different database systems.
7. **Security:**
   * Python web frameworks emphasize security features, helping developers build secure web applications. Features like protection against common web vulnerabilities and secure authentication mechanisms are built into frameworks like Django.
8. **Asynchronous Programming:**
   * With the introduction of asynchronous features in recent Python versions and frameworks like FastAPI, developers can build efficient and scalable web applications by handling multiple tasks concurrently.
9. **Integration Capabilities:**
   * Python can seamlessly integrate with other technologies and services, making it suitable for building web applications that need to interact with external APIs, services, or data sources.

Overall, Python's combination of readability, robust frameworks, extensive libraries, and community support makes it a compelling choice for backend development in web applications

**DJANGO**



Django is a high-level, open-source web framework for the Python programming language. It follows the Model-View-Controller (MVC) architectural pattern, but in Django terminology, it's referred to as the Model-View-Template (MVT) pattern. Django's primary goal is to simplify and accelerate the process of building robust, maintainable web applications.

Django is a powerful and feature-rich web framework that simplifies the development of web applications by providing a high-level, Pythonic interface for common tasks. It's an excellent choice for building scalable and maintainable web applications.

**Advantages of Django**

1. **Rapid Development:**
   * Django follows the "Don't Repeat Yourself" (DRY) principle, which promotes code reusability. It includes a high-level ORM system, automatic admin interface, and built-in templates, allowing developers to build applications quickly and efficiently.
2. **Model-View-Template (MVT) Architecture:**
   * Django's MVT architecture provides a clear separation of concerns, making it easier to organize and maintain code. Models represent the data, views handle the presentation and business logic, and templates manage the user interface.
3. **Object-Relational Mapping (ORM):**
   * Django's ORM system allows developers to interact with databases using Python code, abstracting away the need for raw SQL queries. This simplifies database operations, enhances code readability, and reduces the risk of SQL injection.
4. **Admin Panel:**
   * Django includes an automatic admin interface generated based on the application's models. This admin panel allows for easy content management, user authentication, and other administrative tasks without writing additional code.
5. **Built-in Security Features:**
   * Django is designed with security in mind. It provides built-in protection against common web vulnerabilities, such as SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF). Developers can focus on building secure applications without having to implement these features manually.
6. **Scalability:**
   * Django applications can scale effectively to handle increased traffic and data. The framework provides tools and best practices for optimizing performance and managing scalability challenges.
7. **Community and Documentation:**
   * Django has a large and active community of developers. This community contributes to the framework's development, provides support, and shares a wealth of knowledge. The official Django documentation is comprehensive, making it easy for developers to learn and reference.
8. **Versatility and Flexibility:**
   * Django is a versatile framework that supports a variety of web development tasks. It can be used for building content management systems (CMS), e-commerce platforms, social networking sites, and more. Additionally, Django can be easily integrated with other technologies and services.
9. **Middleware Support:**
   * Django's middleware system allows developers to process requests globally before they reach the view or after they leave the view. This is useful for implementing cross-cutting concerns such as authentication, security, and caching.
10. **Testing Framework:**
    * Django encourages the writing of tests with its testing framework. Developers can create unit tests, functional tests, and integration tests to ensure the reliability and stability of their applications.

**SOFTWARE ENGINEERING**

**MODEL**

**USED**

**SOFTWARE ENGINEERING MODEL USED**

**5.1 PROTOTYPE MODEL**

The Prototyping Model is one of the most popularly used Software Development Life Cycle Models (SDLC models). This model is used when the customers do not know the exact project requirements beforehand. In this model, a prototype of the end product is first developed, tested and refined as per customer feedback repeatedly till a final acceptable prototype is achieved which forms the basis for developing the final product.

In this process model, the system is partially implemented before or during the analysis phase thereby giving the customers an opportunity to see the product early in the life cycle. The process starts by interviewing the customers and developing the incomplete high-level paper model. This document is used to build the initial prototype supporting only the basic functionality as desired by the customer. Once the customer figures out the problems, the prototype is further refined to eliminate them. The process continues till the user approves the prototype and finds the working model to be satisfactory.

There are 2 approaches for this model:

**1. Rapid Throwaway Prototyping**

This technique offers a useful method of exploring ideas and getting customer feedback for each of them. In this method, a developed prototype need not necessarily be a part of the ultimately accepted prototype. Customer feedback helps in preventing unnecessary design faults and hence, the final prototype developed is of a better quality.

**2. Evolutionary Prototyping**

In this method, the prototype developed initially is incrementally refined on the basis of customer feedback till it finally gets accepted. In comparison to Rapid Throwaway Prototyping, it offers a better approach which saves time as well as effort. This is because developing a prototype from scratch for every iteration of the process can sometimes be very frustrating for the developers.

**Advantages**

* The customers get to see the partial product early in the life cycle. This ensures a greater level of customer satisfaction and comfort.
* New requirements can be easily accommodated as there is scope for refinement.
* Missing functionalities can be easily figured out.
* Errors can be detected much earlier thereby saving a lot of effort and cost, besides enhancing the quality of the software.
* The developed prototype can be reused by the developer for more complicated projects in the future.
* Flexibility in design.

**Disadvantages**

* Costly with respect to time as well as money.
* There may be too much variation in requirements each time the prototype is evaluated by the customer.
* Poor Documentation due to continuously changing customer requirements.
* It is very difficult for the developers to accommodate all the changes demanded by the customer.
* There is uncertainty in determining the number of iterations that would be required before the prototype is finally accepted by the customer.
* After seeing an early prototype, the customers sometimes demand the actual product to be delivered soon.
* Developers in a hurry to build prototypes may end up with sub-optimal solutions.
* The customer might lose interest in the product if he/she is not satisfied with the initial prototype.

**Use**

The Prototyping Model should be used when the requirements of the product are not clearly understood or are unstable. It can also be used if requirements are changing quickly. This model can be successfully used for developing user interfaces, high technology software-intensive systems, and systems with complex algorithms and interfaces. It is also a very good choice to demonstrate the technical feasibility of the product.

Flow chart

**SYSTEM**

**ANALYSIS**

**SYSTEM ANALYSIS**

System analysis is the process of studying the student grievance cell system and procedures, generally referred to as complain systems, to see how they can operate and whether improvement is needed. This may involve examining data movement and storage, machines and technology used in the system, programs that control the machines, people providing inputs, doing the  processing and receiving the outputs.

**6.1INVESTIGATION PHASE**

The investigation phase is also known as the fact-finding stage or the analysis of the current system. This is a detailed study conducted with the purpose of wanting to fully understand the existing system and to identify the basic information requirements. Various techniques may be used in fact-finding and all fact obtained must be recorded. A thorough investigation was done in every effected aspect when determining whether the proposed system is feasible enough to be implemented.

**6.1.1 Investigation:-**

 As it was essential for us to find out more about the present system, we used the following methods to gather the information: -

1. Observation: - Necessary to see the way the system works first hand.
2. Document sampling: - These are all the documents that are used in the system. They are necessary to check all the data that enters and leaves the system.
3. Questionnaires: - These were conducted to get views of the other employees who are currently employed in the system.

**6.1.2 Analysisofthe investigationstrengthsofsystem :**

1. No complex equipment: - The equipment that is used is very simple and no special skills have to be mastered to be able to operate the system. Therefore no training is required for the employees.

2. Low cost: - There is little money spent in maintaining the present system other than buying the necessary office equipment.

**6.1.3 Constraints and Limitations :**

The constraints and limitation within a system are the drawbacks that occur during the implementation of the system. These limitations and constraints can crop up in almost every system; the most important fact is to find a way to overcome these problems. Software design is the first of three technical activities  – design, code generation, and test that are required to build and verify the software. Each activity transforms information in manner that ultimately results in validated computer software. The design task produces a data design, an architectural design, an interface design and component design. The design of an information system produces the details that clearly describe how a system will meet the requirements identified during system analysis. The system design  process is not a step by step adherence of clear procedures and guidelines. When we started working on system design, we face different types of problems; many of these are due to constraints imposed by the user or limitations of hardware and software available. Sometimes it was quite difficult to enumerate that complexity of the problems and solutions thereof since the   12 variety of likely problems is so great and no solutions are exactly similar however the following consideration we kept in mind during design phase.

**6.2 SOFTWARE REQUIREMENT SPECIFICATION**

The software requirement specification is produced at the culmination of the analysis task. The function and performance allocated to software as part of system engineering are refined by establishing a complete information description, a detailed functional description, a representation of system behavior, an indication of performance requirement and design constraints appropriate validation criteria, and other information pertinent to requirement. The introduction to software requirements specification states the goals and objectives of the software, describing it in the context of the computer based system. The Information Description provides a detailed description of the problem that the software must solve. Information content, flow and structure are documented. A description of each function required to solve the problem is presented in the Functional Description. Validation Criteria is probably the most important and ironically the most often neglected section of the software requirement specification. Software requirement specification can be used for different purpose.

**FEASIBILITY**

**STUDY**

**FEASIBILITY STUDY**

Prior to stating whether the system we have to develop is feasible or not we believe that we should emphasize on what is implied by the word “Feasibility”. Feasibility is the measure of how beneficial or practical the development of the system will be to the organization. It is a  preliminary survey for the systems investigation. It aims to provide information to facilitate a later in-depthinvestigation.

**TYPES OF FEASIBILITY-**

There are various measures of feasibility that helps to decide whether a particular project is feasible or not. These measures include –

1**. Operational Feasibility**

2. **Technical Feasibility**

3**. Economical Feasibility**

 Each of these types will be explained in detail throughout the project report.

**7.1 OPERATION FEASIBILITY**

A proposed system is beneficial only if it can be turned into an information system that will meet the operational requirements of an organization. A system often fails if it does not fit within existing operations and if users resist the change. Important issues a systems developer must look into are:

* Will the new system be used if implemented in an organization?
* Are there any major barriers to implementation or is proposed system accepted without destructive resistance?

The whole purpose of computerizing the Complaint Management is to handle the work much more accurately and efficiently with less time consumption. There will be additional work to be completed, because now the cellular company will have to maintain database of both their employees as well as their Customers.

Another important fact to be regarded is the security control, which is handled by the system. Since data regarding each Customer and the Organization is confidential, security is a key issue. Information falling into the wrong hands could jeopardize the entire organization. Unlike in semi-computerized systems

The proposed system offers adequate control to protect the organization against fraud and embezzlement and guarantees the accuracy and Security of data and information. This is handled  by the system providing individuals with separate login names and passwords.

The new system is user-friendlier, which enables the end-user to complete his/her work efficiently and accurately with interest. After taking the above fact into consideration we can state the operating of the proposed system within the organization is feasible.

* In this phase of the feasibility study the following two main topics
* Technical Performance Aspect
* Acceptance within the organization

Technical performance aspect is explained in the technical feasibility report and there is no new information is needed in this to explain it again, but as for the acceptance within the organization the following points are important and those are explained according to the topics

**Whether the system provides right information to the right place?**

In the current system which is the semi computerized system the Information may be lost in the process of sending from one place to another. This is mainly due to human interaction in the process of the transferring information from one place to another.

**Whether the new system affects the current users in the system?**

The new proposed system will affect the users in the following areas:-

* Accuracy
* Efficiency
* Productivity
* Robustness
* Lesser time consuming

**7.2 TECHNICAL FEASIBILITY**

Based on the outline design of the system requirements in terms f inputs, output, Procedures, the technical issues raised during technical feasibility include:

**1**. Does the necessary technology exist to do what is proposed?

**2**. Does the proposed equipment have the technical capacity to hold the data required to use in the new system?

**3**. Adequate responses provided by the proposed system?

4. Is the system flexible enough to facilitate expansion?

5. Is there any technical guarantee of accuracy, reliability, ease of access and data security?

The system developer’s task is to view needed capabilities in light of currently available technology. Our site works hand in hand with high technology. A database has to be maintained in order to update and backup data whenever required. To create databases we use SQL server. After taking the above facts into consideration we can state that the new proposed system is technically feasible.

**7.3 ECONOMICAL FEASIBILITY**

In making recommendations a study of the economics of the proposed system should be made. Even though finding out the costs of the proposed project is difficult we assume and estimate the costs and benefits as follows.

* Costs associated with the development of the system.
* Costs associated with operating the system

**CONCEPTUAL**

**MODULE**

**CONCEPTUAL MODULE**

**8.1DATA FLOW DIAGRAM**

A data flow diagram shows the way information flows through a process or system. It includes data inputs and outputs, data stores, and the various subprocesses the data moves through. DFDs are built using standardized symbols and notation to describe various entities and their relationships.

Data flow diagrams visually represent systems and processes that would be hard to describe in a chunk of text. You can use these diagrams to map out an existing system and make it better or to plan out a new system for implementation. Visualizing each element makes it easy to identify inefficiencies and produce the best possible system.

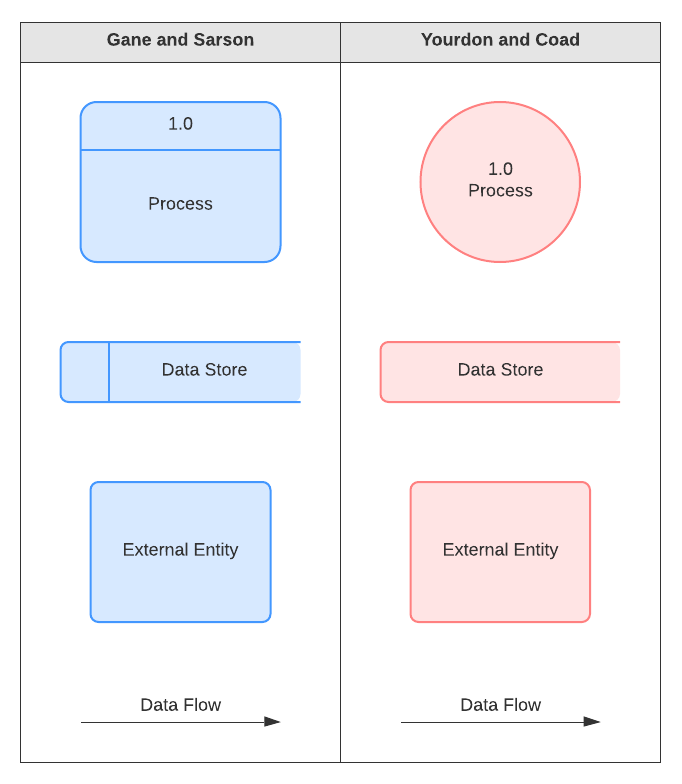
**8.1.1 Symbols and notations-**

**External Entity** – Also known as actors, sources or sinks, and terminators, external entities produce and consume data that flows between the entity and the system being diagrammed. These data flows are the inputs and outputs of the DFD. Since they are external to the system being analysed, these entities are typically placed at the boundaries of the diagram. They can represent another system or indicate a subsystem.

**Process** – An activity that changes or transforms data flows. Since they transform incoming data to outgoing data, all processes must have inputs and outputs on a DFD. This symbol is given a simple name based on its function, such as “Ship Order,” rather than being labelled “process” on a diagram. In Gane-Sarson notation, a rectangular box is used and may be labelled with a reference number, location of where in the system the process occurs and a short title that describes its function. Processes are typically oriented from top to bottom and left to right on a data flow diagram.

**Data Store** – A data store does not generate any operations but simply holds data for later access. Data stores could consist of files held long term or a batch of documents stored briefly while they wait to be processed. Input flows to a data store include information or operations that change the stored data. Output flows would be data retrieved from the store.

**Data Flow** – Movement of data between external entities, processes and data stores is represented with an arrow symbol, which indicates the direction of flow. This data could be electronic, written or verbal. Input and output data flows are labelled based on the type ofdata or its associated process or data store, and this name is written alongside the arrow.



**8.1.2 Zero level (context level) data flow diagram**

**8.2 ENTITY RELATIONSHIP DIAGRAM**

ER Model is used to model the logical view of the system from data perspective which consists of these components:

**Entity, Entity Type, Entity Set –**

An Entity may be an object with a physical existence – a particular person, car, house, or employee – or it may be an object with a conceptual existence – a company, a job, or a university course.

An Entity is an object of Entity Type and set of all entities is called as entity set. e.g.; E1 is an entity having Entity Type Student and set of all students is called Entity Set. In ER diagram, Entity Type is represented as:

Student

Entity Type

Entity Set

**8.2.1 Entity-**

Entities are objects or concepts that represent important data. Entities are typically nouns such as product, customer, location, or promotion. There are three types of entities commonly used in entity relationship diagrams.

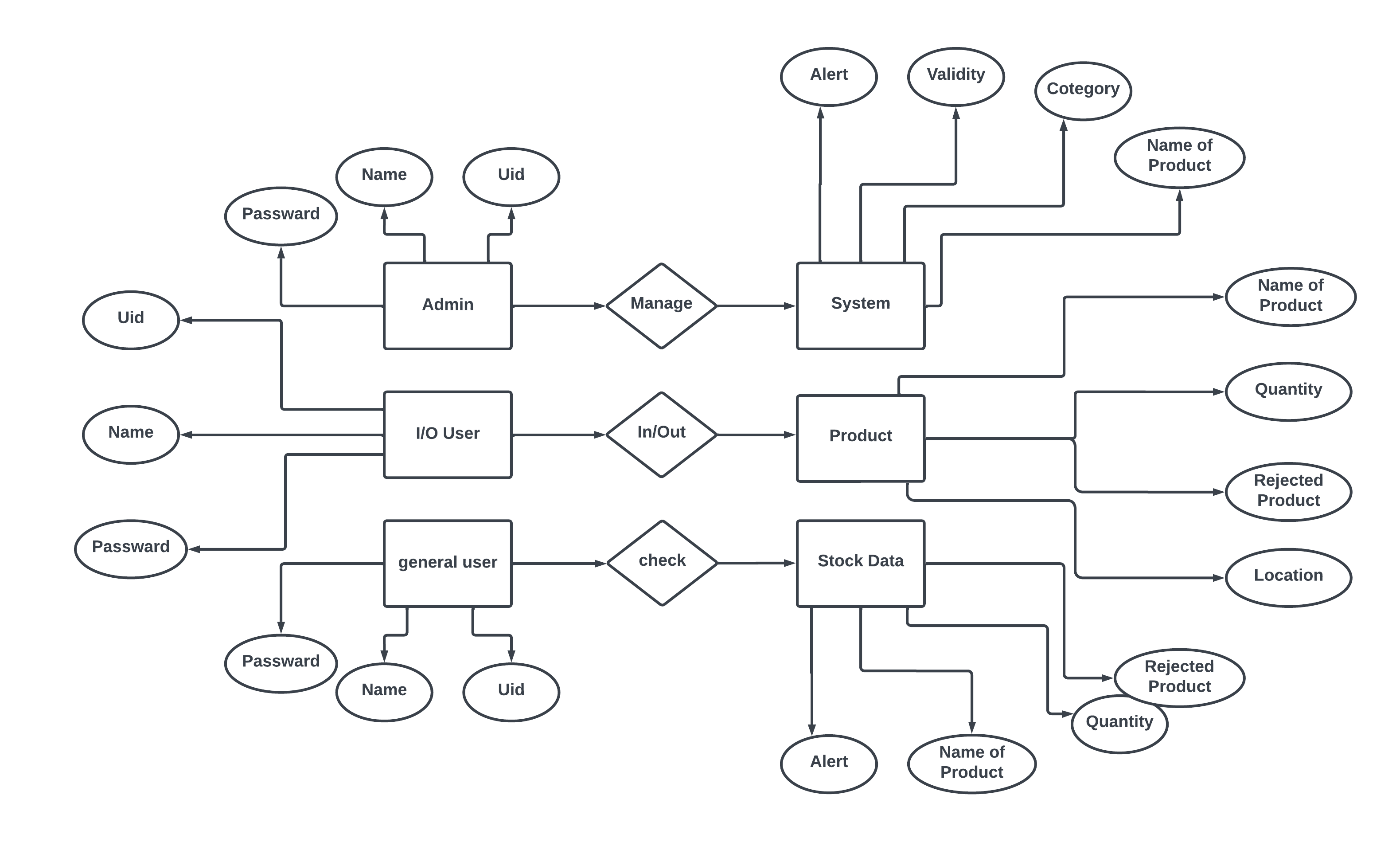
|  |  |  |
| --- | --- | --- |
| **Entity Symbol** | **Name** | **Description** |
| ERD Symbols and Meaning - Entity | Entity | An entity is represented by a rectangle which contains the entity’s name. |
| ERD Symbols and Meaning - Weak Entity | Weak Entity | An entity that cannot be uniquely identified by its attributes alone. The existence of a weak entity is dependent upon another entity called the owner entity. The weak entity’s identifier is a combination of the identifier of the owner entity and the partial key of the weak entity. |
| Associative  Entity | Associative Entity | An entity used in a many-to-many relationship. All relationship for the associative entity should be many. |

**8.2.2Attributes-**

ERD attributes are characteristics of the entity that help users to better understand the database. Attributes are included to include details of the various entities that are highlighted in a conceptual ER diagram.

|  |  |  |
| --- | --- | --- |
| **Attribute Symbol** | **Name** | **Description** |
| ERD Symbols and Meaning - Attribute | Attribute | Attributes are characteristics of an entity |
| ERD Symbols and Meaning - Key attribute | Key attribute | An attribute that uniquely identifies a particular entity. The name of a key attribute is underscored. |
| ERD Symbols and Meaning - Multivalue attribute | Multivalued attribute | Multivalued attributes are those that are can take on more than one value. |
| ERD Symbols and Meaning - Derived attribute | Derived attribute | Derived attributes are attributes whose value can be calculated from related attribute values. |

**ER DIAGRAM**



**modules**

**WORKING MODULES IN THIS PROJECT**

Our project includes the following modules.

* **Sign up module**
* **Sign in module**
* **System Design module**
* **Inventory handling module**
* **Status of Inventory module**

**9.1 SIGN UP MODULE-**

**9.2 SIGN IN MODULE-**

**9.3SYSTEM DESIGN MODULE-**

**9.4 INVENTORY HANDLING MODULE**-

**9.5STATUS OF INVENTORY MODULE-**

**CODE AND SNAP SHOTS**

**TESTING**

**SYSTEM TESTING**

Testing should be done throughout the implementation process. Even before an application is installed, it makes sense to verify that the basic platform is capable of achieving its design capabilities. System testing is a critical process. Testing is a process of executing a program with the explicit intention of finding errors that is, making the program fail. This help in finding the bottlenecks in the system. Executing a program in a simulated environment performs testing. The feedback from testing phase generally produces changes in the software to deal with errors and failures that are uncovered.

**BLACK BOX TESTING -**

In Black Box testing or functional testing test cases are decided. Test cases are decided on the basis of the requirements or the specifications of the program or module.

Black Box testing is done in the project to remove the errors:

* Incorrect or missing function.
* Interface errors.
* Errors in data structures or external database access.
* Behavioural or performance errors
* Errors in initiation & termination.

**WHITE BOX TESTING–**

The White Box testing or Structural testing performs close operation of procedural details. They test the software logical path by having test cases exercising specific sets of conditions and loops.

White Box testing is done in the project to remove the errors.

• All modules path have been exercised at least once.

• Exercised on logical decisions.

• Executed all loops at their boundaries and within their operational bounds.

• Exercised internal data structure to ensure their validity.

**UNIT TESTING-**

Unit testing focuses on verification efforts of the smallest grid of software designing i.e. a software component or module is tested.

Following tests were performed during unit test:

1. Module Interface Testing:

Module interface was tested to ensure information flow in and out of the program unit.

2.Local Data Structure Testing:

Local Data Structure were tested to ensure that data store temporally maintain their integrity during all steps in algorithm execution.

3.Boundary Condition Test:

Boundary Conditions were tested to make sure that the modules operate properly at boundaries.

4.Independent Path Test:

All independent paths through control structure were checked to make sure that all statements in a module have been executed.

5.Error Handling Path Test:

Error handling path test was performed to handle exceptions.

**INTEGRATION TESTING -**

Integration testing is done to tackle the problem of interface i.e. putting all the interfaces together. When the separate modules are put together and work in an integrated manner, this testing is performed. This testing is a Systematic technique. This testing is performed to check that data should not be lost across an interface. The objective is to take unit tested modules and build a program structure that has been dictated by design.

Regression: Regression was done to ensure proper working of each module with the whole system. Each module was embedded in the system and the whole tested for integrity.

**SYSTEM TESTING–**

System testing is done when the entire system has been fully integrated. The purpose of the system testing is to test how the different modules interact with each other and whether the system provides the functionality that was expected.

**FUTURE SCOPE**

**Conclusion**

In this project “ Bot for Stock” we have design a inventory management system which provide the AI based functionality to manage the stock system .so that the system will be convenient to use and help the company to work properly.

**BIBLIOGRAPHY**

**REFERENCE**

**Web References -**

<https://www.w3schools.com/>

<https://www.softwareadvice.com/>

Books References -

Software Engineering Books.

HTML5 AND CSS3 by Andy Harris.

JavaScript and jQuery by JON DUCKETT (Author)

Python Programming By JOHN ZELLE.

Let Us Python (5th Edition) BY Yashavant Kanetkar and Aditya Kanetkar.