**AI BASED ONLINE TRANSACTION FRAUD DETECTION SYSTEM**

**PROJECT REPORT**

Submitted to

## NEHRU ARTS AND SCIENCE COLLEGE, COIMBATORE (AUTONOMOUS)

In Partial Fulfillment of the Requirement of the Award for the degree of

**BACHELOR OF COMPUTER APPLICATIONS**

Submitted by

**S. DEEKSHITH (Reg NO.:19UGCA035)**

Under the guidance of

**Dr. A. KALAIVANI., MCA., M. Phil., Ph. D.,**

Assistant Professor & Head Department of Computer Science School of Computational Sciences



**NEHRU ARTS AND SCIENCE COLLEGE**

(An Autonomous Institution affiliated to Bharathiar University) (Reaccredited with “A” Grade by NAAC, ISO 9001:2015 & 14001:2004 Certified

Recognized by UGC with 2(f) &12(B), Under Star College Scheme by DBT, Govt. of India) Nehru Gardens, Thirumalayampalayam, Coimbatore - 641 105, Tamil Nadu.

**JUNE-2022**

**NEHRU ARTS AND SCIENCE COLLEGE**

(An Autonomous Institution affiliated to Bharathiar University) (Reaccredited with “A” Grade by NAAC, ISO 9001:2015 & 14001:2004 Certified

Recognized by UGC with 2(f) &12(B), Under Star College Scheme by DBT, Govt. of India) Nehru Gardens, Thirumalayampalayam, Coimbatore - 641 105, Tamil Nadu.

**School of Computational Sciences**

**DEPARMENT OF COMPUTER APPLICATIONS**



**CERTIFICATE**

This is to certify that project report entitled “**AI BASED ONLINE TRANSACTION FRAUD DETECTION SYSTEM”** is a Bonafide work done by **S. DEEKSHITH (Reg NO.:19UGCA035)** in partial fulfillment of the requirement of the award for the Degree of **BACHELOR OF COMPUTER APPLICATIONS** of NEHRU ARTS AND SCIENCE COLLEGE (AUTONOMOUS), COIMBATORE.

**INTERNAL GUIDE HEAD OF THE DEPARTMRNT**

Certify that we examined the Candidate in Project Work/Viva-Voice Examination held at **NEHRU ARTS AND SCIENCE COLLEGE** on

 .

**Internal Examiner External Examiner**

**DECLARATION**

I hereby declare that the project entitled **“AI BASED ONLINE TRANSACTION FRAUD DETECTION SYSTEM”**, submitted to Nehru Arts and Science College (Autonomous), Coimbatore, is a record of the original work done by **S. DEEKSHITH** under the supervision and guidance of **Dr. A. KALAIVANI., MCA., M. Phil., Ph. D.,** Department of Computer Applications, **Nehru Arts and Science College** during the Academic Year of 2021-2022.

PLACE: COIMBATORE Signature of the Candidate

#### DATE: S. DEEKSHITH

**ACKNOWLEDGEMENT**

I wish to record my indebtedness and thankfulness to all who helped me prepare this Report titled JOB POINT and present it in a satisfactory way. This Report is part of my work related to the Project and Viva Voce. I sincerely thank our Principal **DR. B. ANIRUDHAN, M. A, B.Ed., M.Phil., Ph.D.,** Nehru Arts and Science for permitting me to undertake this project as a part of curriculum and for giving me the best facilities and infrastructure for the course and project. My immense gratitude for my Head of the Department, **Dr. K. SELVAVINAYAKI, MCA, M.Phil., PhD.,** for her consultant help and creative ideas over the period of the project work.

I owe a great deal of gratitude to my faculty guide **Dr. A. KALAIVANI., MCA., M. Phil., Ph. D.,** Department of Computer Applications for her consultant help and creative ideas over the period of project work. I express my sincere words of gratitude to other faculty members and staff members of the Department of Computer Applications. I extend my sincere thanks to my parents and I would like to thank all my friends for their moral support rendered to me to complete this project in grand success.

#### S. DEEKSHITH

**CONTENTS**

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| Sl.No | **CONTENTS** | **Page. No** |
| I | **ACKNOWLEDGEMENT** |  |
| II | **DECLERATION** |  |
| III | **CERTIFICATE FROM THE COMPANY** |  |
| IV | **BONAFIDE CERTIFICATE** |  |
| V | **SYNOPSIS** |  |
| 1 | **INTRODUCTION** | 1 |
| 1.1 | About the project | 2 |
| 1.2 | Organization profile | 4 |
| 2 | **SYSTEM ANALYSIS** | 7 |
| 2.1 | Existing system | 8 |
| 2.2 | Proposed system | 9 |
| 2.2.1 | System study | 9 |
| 2.3 | System specification | 11 |
| 2.3.1 | Hardware specification | 11 |
| 2.3.2 | Software specification | 11 |
| 2.3.3 | About the software | 12 |
| 3 | **SYSTEM DESIGN** | 18 |
| 3.1 | Design Notations | 19 |
| 3.1.1 | Data flow diagram | 21 |
| 3.1.2 | ER diagram | 24 |
| 3.2 | Design Process | 25 |

|  |  |  |
| --- | --- | --- |
| 3.2.1 | Input design | 25 |
| 3.2.2 | Database design | 26 |
| 3.2.3 | Output design | 31 |
| 4 | **SYSTEM TESTING AND IMPLEMENTATION** | 32 |
| 4.1 | Testing methodologies | 33 |
| 4.2 | System implementation | 36 |
| 5 | **CONCLUSION & FUTURE ENHANCEMENTS** | 38 |
| 5.1 | Bibliography | 40 |
| 5.2 | Appendix | 41 |
| a | Sample Screens | 51 |
| b | Reports | 59 |

# SYNOPSIS

### ABSTRACT

The objective of this project AI BASED ONLINE TRANSACTION FRAUD DETECTION is to develop a website which has capability to restrict and block the transaction performing by attacker from genuine user’s credit card details. The system here is developed for the transactions higher than the customer’s current transaction limit. We tried to detect fraudulent transaction before transaction succeeds. During registration we take required information which is efficient to detect fraudulent user activity. An FDS runs at a credit card issuing bank. Each incoming transaction is submitted to the FDS for verification.

FDS receives the card details and transaction value to verify, whether the transaction is genuine or not. The types of goods that are bought in that transaction are not known tothe FDS. Bank declines the transaction if FDS confirms the transaction to be fraud. User spending patterns and geographical location is used to verify the identity. If any unusual pattern is detected, the system requires re-verification. Based upon previous data of that user the system recognizes unusual patterns in the payment procedure. System will block the user after 3 invalid attempts.

The frontend of this project **Django framework, python 3.8 , CSS, java script** and the backend is **MySQL server 5.6** which makes it easy for creating and generating code. Windows 10 is used as Operating System and hosting the web application in WSGI.

# INTRODUCTION

## 1. INTRODUCTION

### ABOUT THE PROJECT

Credit Cardholders have many beneficiary schemes to hold interest-free balances for

almost two months with “grace-period”. Suitable information on fraudulent activities is tactical to the banking industry. Banks have huge databases. Extraction of important business information can be done from these data stores. Data stores have some patterns into clusters thatare natural to the input data. Concept of fraud detection has been laid on data mining techniqueswhich include association rules, clustering and classification. The chief point of research on fraud detection has been focused on pattern matching in which abnormal patterns are identified from the normal ones. Popularity of online shopping is growing day by day at a high pace.

Nowadays, Credit card is the most popular mode of payment (59 percent), Germany and Great Britain have the largest number of online shoppers. Usually, Retailers like Wal-Mart handle a much larger number of credit card transactions online as well as regular purchases . There are numerous options for handing credit card payments on the Internet, as the processing of credit card transactions is generally independent of the type of e-commerce exchange. While a huge portion of e-commerce would consist of credit card purchases, like regular or often. It is more important for businesses and organizations that rely upon income from ecommerce to know the options available as well as costs linked with credit card transaction processing on the Internet. No one has any hint about transactions being processed are whether a fraudulent transaction or legitimate which has passed the prevention mechanisms. Therefore, the goal of thefraud detection systems is to pre-determine every transaction for the possibility of being fraudulent regardless of the prevention mechanisms and to categorize transactions as fraudulent ones as early as possible after the fraudster has begun to commit a fraudulent transaction. Credit card fraud detection is a tremendous task but also a trendy problem to solve. Many fraud detection systems estimate the transactions and generate a suspicion score (commonly a probability between 0 and 1) which shows the chances of that transaction to be fraudulent.

Computational procedures of these scores are applicable to the techniques used to build the model(s) in the fraud detection systems. These corresponding scores are used with a predefined

threshold value to differentiate between fraudulent transactions from the legitimate ones easily.

### Modules and their Description

This system comprises of 2 major modules with its sub-modules:

### Admin

* + - 1. **Login:** Admin need to login using valid login credentials in order to access the system.
			2. **Add / View Products:** Admin can add new product with its details into the system.
			3. **View Transactions:** System allows admin to view all the transactions done by the registered users.

### User

* + - 1. **Registration:** Here, user first need to registration themselves with details to access the system.
			2. **Login:** After a successful registration, user then need to login into the system by inputting their credentials into the system.
			3. **View Products:** User can view multiple products with its details. Interested users can purchase a product via online transaction.
			4. **Buy a Product:** User can select payment mode to perform transactions by providing the card details like card no., CVV code, Expiry Date and Holders name.
			5. **View Transaction:** List of all the transactions will be displayed to the user.

### ORGANIZATION PROFILE

NOYCE Cyber Solutions – NOYCE provides enterprise level web design, web development and maintenance services in and outside India with the development center in Palakkad and global delivery center all over India. Our expert web designers and developers accomplished various website projects and portals starting from simple cms to complex customized web portals for many business domains.

Every Individual website that we design is custom built according to your specific needs of our clients. Our website designers are good at delivering value for digital branding of our clients profile online. We always make sure our clients get an impressive and effective online presence as per global standards. Our website design and maintenance service will ensure client’s stability and maintain your critical Internet services. Noyce is a brand as website Development Company in Kerala. Noyce is good at developing and designing affordable and professional website designs for companies all over India. As a website designing company we help small, medium and large businesses to develop successful websites that will separate our clients from the rest and eventually help them get customers from all over the world through online presence. In the modern fast growing digital world, a website can represent a company globally**.** So it has to be designed with international standards, modern facility and should be better from competitor’s website. This is the service that we offer for the global companies. Website is a powerful digital tool that can achieve your marketability on web. We can promote your website as well as the business with impressive logo, web site design and more informative contents.

#### Our Team

#### Empowering Growth

Our people are our greatest asset and biggest differentiator. They are passionate about results, and also believe in having a lot of fun along the way. However, that does not take away from the focus on work. Our people are passionate about delivering results to clients. All Noyce are direct and

straightforward—even if that means telling the uncomfortable truth. We are ambitious and impatient for success, and yet down-to-earth and approachable.

In short, Noyce are not only the kind of people you would love to work with, but they are also the people you would want to socialize with outside work. We encourage you to take every opportunity to interact with our people and witness the vibrancy of the office in person

#### Quality Pledge

We are committed to being very aggressive in our attitude towards quality and customer service, primarily since we want to be ranked as the "best" in our business. Quality is not just another goal, it is our basic strategy for survival and future growth.

#### Priority

Our customers demand and warrant a high quality product---it is our responsibility to give them what they want. If we don’t they’ll find someone who can. If customer requirements are unclear, then it is our job to seek out a better understanding of their requirements/specifications. If we fail at any time, then we must determine what went wrong and assure that it doesn't happen again.

#### Objectives

Our quality objectives are to furnish high quality products, on time, and at the lowest cost. The attainment of such objectives will lead to, customer satisfaction, enhanced copper performance at the application level, and ongoing improvements in process efficiency. Once an objective is achieved, it should be recognized and reset to stimulate further quality improvement. To reach our objectives, we will have to maintain a constant focus on quality with full dedication, commitment, and teamwork.

#### Vision

Our journey is Total Quality Management--fully satisfying our customers’ requirements through a process of continuous improvement. It's critical to understand that Total Quality Management is not a short term program. It's a long term commitment aimed at continuously improving the way we work, providing a safe work environment, managing our business processes, and supplier selection/retention. It is our goal to posture our company for market expansion, thereby providing improved job security and quality of life for all.

#### Quality first

It must be clearly understood that we'll not allow quality to take second place behind cost or schedule. All employees have the right to question their supervisor's decisions or actions if they feel thatquality is being compromised.

#### Product

Noyce is a leading customized web application development company with expertise in application development using open source technology. We have combined experience of 25 + years in web application development including development of retail sites, matrimony website, trading website, classified website, content management website etc. We develop scalable and robust web applications from scratch on platforms like Drupal, WordPress, Joomla, Opencart & Magento.

We also develop custom themes, plugins for wordpress, joomla, opencart & web applications with php frameworks like Codeigniter/CackePHP. With our experience in web application developmentfor publishing houses and corporate businesses – we are capable of building performance driven, responsive scalable web application, for cloud and traditional servers.

#### Application development

At Noyce we develop unique web based applications which serves all the needs of our esteemed client from several business domain. We have a team of experienced, enthusiastic, result oriented, dedicated manpower who delivers of web application development. We develop platform independent web applications which can be run on all kinds of platforms and serves the exact need of clients.

# SYSTEM ANALYSIS

## 2. SYSTEM ANALYSIS

### EXISTING SYSTEM

In the Existing System, many transaction media such as ATM, bank cards or debit cards, require the use of pins, passwords, and in some cases “biometrics” to authenticate the legitimate owner. Credit cards create fascinating problems since generally no pin is required for their use; only the name, expiration date and account number is required. Popular means of criminally transacting with credit cards is by stealing someone’s identity & in some cases, creating a new fake identity. Therefore, fraudulent electronic transactions (E-transaction) by credit card are the key problem. Credit cards need not be necessarily physically obtainable to transact and over the internet they can be used to fraudulently transact on the web for better and heavier losses for banks and their customers if caught by fraudsters.

### Drawbacks

* + - Fraud often leads to chargebacks. A chargeback is a sum that must be returned by the merchant to the cardholder after a fraudulent transaction.
		- Processing a chargeback includes operational costs such as transaction fees, legal fees, currency conversions etc.
		- Another loss is the product sold to the fraudster - the merchant will not get the “sold” product back.
		- If the merchant incurs a large number of chargebacks the result can, at worst, be that he can’t find an acquirer to process his payments, as he is considered a high-risk customer.

### PROPOSED SYSTEM

The chief idea in fraud detection is that fraud may be detected by noticing significant deviation from the “normal behavior” of a customer’s account. That is why; behavior of an account can thus be used to protect that account. Currently banks have come to realize that a fused, global app.

#### Advantages

* The system stores previous transaction patterns for each user.
* Based upon the user spending ability and even country, it calculates user’s characteristics
* The system is more secure with OTP (One Time Password) implementation.
* IP address tracking at every transaction.
* Security questions for payment limit crossed
* More than 20-30 % deviation of the user's transaction (spending history and operating country) is considered as an invalid attempt and the system takes action.

### SYSTEM STUDY

#### FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and a business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of themajor requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

* + - * ECONOMICAL FEASIBILITY
			* TECHNICAL FEASIBILITY
			* SOCIAL FEASIBILITY

### ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of funds that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

### TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on theavailable technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system musthave a modest requirement, as only minimal or null changes are required for implementing this system.

### SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

### SYSTEM SPECIFICATION

* + 1. **Hardware Specification:**
			- Processor :Core i3
			- Hard Disk :160 GB
			- Memory : 1GB RAM
			- Monitor

### Software Specification:

* + - * Windows 7 or higher
			* Python
			* Django framework
			* MySQL database

### About The Software

**FRONT END**

Python is an interpreter, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding; make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

#### DJANGO FRAMEWORK

Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It’s free and open source.

**ADVANTAGES**

**Complete**

Django follows the "Batteries included" philosophy and provides almost everything developers might want to do "out of the box". Because everything you need is part of the one "product", it all works seamlessly together, follows consistent design principles, and has extensive and [up-to-date](https://docs.djangoproject.com/en/stable/) [documentation.](https://docs.djangoproject.com/en/stable/)

**Versatile**

Django can be (and has been) used to build almost any type of website — from content management systems and wikis, through to social networks and news sites. It can work with any client- side framework, and can deliver content in almost any format (including HTML, RSS feeds, JSON, XML, etc). The site you are currently reading is built with Django!

Internally, while it provides choices for almost any functionality you might want (e.g. several popular databases, templating engines, etc.), it can also be extended to use other components if needed.

**Secure**

Django helps developers avoid many common security mistakes by providing a framework that

has been engineered to "do the right things" to protect the website automatically. For example, Django provides a secure way to manage user accounts and passwords, avoiding common mistakes like putting session information in cookies where it is vulnerable (instead cookies just contain a key, and the actual data is stored in the database) or directly storing passwords rather than a password hash.

A password hash is a fixed-length value created by sending the password through a [cryptographic hash function.](https://en.wikipedia.org/wiki/Cryptographic_hash_function) Django can check if an entered password is correct by running it through the hash function and comparing the output to the stored hash value. However due to the "one-

way" nature of the function, even if a stored hash value is compromised it is hard for an attacker to work out the password.

Django enables protection against many vulnerabilities by default, including SQL injection, cross-site scripting, cross-site request forgery and clickjacking (see [Website security](https://developer.mozilla.org/en-US/docs/Learn/Server-side/First_steps/Website_security) for more details of such attacks).

**Scalable**

Django uses a component-based “[shared-nothing](https://en.wikipedia.org/wiki/Shared_nothing_architecture)” architecture (each part of the architecture is independent of the others, and can hence be replaced or changed if needed).

**Maintainable**

Django code is written using design principles and patterns that encourage the creation of maintainable and reusable code. In particular, it makes use of the Don't Repeat Yourself (DRY) principle so there is no unnecessary duplication, reducing the amount of code. Django also promotes the grouping of related functionality into reusable "applications" and, at a lower level, groups related code into modules (along the lines of the [Model View Controller (MVC)](https://developer.mozilla.org/en-US/docs/Glossary/MVC) pattern).

**Portable**

Django is written in Python, which runs on many platforms. That means that you are not tied to any particular server platform, and can run your applications on many flavours of Linux, Windows, and Mac OS X. Furthermore, Django is well-supported by many web hosting providers, who often provide specific infrastructure and documentation for hosting Django sites.

**BACK END**

**SQLyog** provides you with powerful means to manage your MySQL databases, combining the features of MySQL Query Browser, Administrator, phpMyAdmin and other MySQL Front Ends and MySQL GUI tools in a single intuitive interface. SQLyog is a fast, easy to use and compact graphical tool for managing your MySQL databases. SQLyog was developed for all who use MySQL as their preferred RDBMS. SQLyog provides you with powerful means to manage your MySQL databases. It features the simplicity of MySQL Front, with the power of EMS MySQL Admin. SQLyog provides detailed profile information for every SQL statement executed. SQLyog is a powerful MySQL manager that restores your previous session the way you left it. SQLyog allows easy access to frequently used SQL scripts. The script files can be stored as files from the SQLyog interface or link to an existing file anywhere where windows can access it - on a local drive or a shared network drive. The SQL scripts and the file links can be organized in folder and subfolders.

#### Sql architecture:



**FEATURES OF MYSQL:**

**Relational Database System:** Like almost all other database systems on the market, MySQL is a relational database system.

**Client/Server Architecture:** MySQL is a [client/server system](https://searchdatamanagement.techtarget.com/feature/Understanding-and-comparing-six-types-of-processing-systems). There is a database server (MySQL) and arbitrarily many clients, which communicate with the server. The clients can run on the same computer as the server or on another computer.

**SQL compatibility:** MySQL supports as its database language -- as its name suggests – SQL (Structured Query Language). SQL is a standardized language for querying and updating data and for the administration of a database. There are several SQL dialects. MySQL adheres to the current SQL standard, although with significant restrictions and a large number of extensions.

**Stored procedures:** Stored procedures (SPs for short) are generally used to simplify certain steps, such as inserting or deleting a data record. For client programmers this has the advantage that they do not have to process the tables directly, but can rely on SPs. Like views, SPs help in the administration of large database projects. SPs can also increase efficiency. MySQL has supported SPs since version 5.0.

**User interface:** There are a number of convenient user interfaces for administering a MySQL server.

# SYSTEM DESIGN

## 3. SYSTEM DESIGN

### DESIGN NOTATIONS

#### Data Flow Design

Data flow diagram is a way of representing system requirements in a graphic form. A DFD also known as a “Bubble Chart” has the purpose of clarifying system requirements and identifies major transformations that will become programs in system design. So it is the starting point of the design phase that functionally decomposes the requirements specifications down to the lowest level of details. A DFD consists of a series of bubbles joined by lines. The bubbles represent data transformation and the lines represent data flow in the system.

#### DFD Symbols

In a DFD, there are four symbols

A square defines a source or destination of system data.

An arrow identifies data flow in motion. It is a pipeline through which in format flows

A circle or a bubble represents a process that transforms incoming data flows into going data flows.

An open rectangle is a data source or data at rest or a temporary of data constructing a DFD.

#### Rules in drawing DFD’s

* Process should have name and numbered for easy reference.
* The direction of flow is from top to bottom and from left to right. Data traditionally flow from source to destination, although they may flow from source.
* When a process is exploded into lower levels, they are numbered.
* The names of data source, sources and destination are written in capital letters. process and data flow names have the first letter of each word capital

.

### DATA FLOW DIAGRAM

#### LEVEL 0 : Hire Appliance System



Fig 3.1.1.1 hire appliance system

#### LEVEL 1 Admin



Fig 3.1.1.2 admin

#### LEVEL1 User



Fig 3.1.1.3 user

### ER DIAGRAM



Fig 3.1.2 ER diagram

### DESIGN PROCESS

* + 1. **Input Design**

Input design is a part of overall system design, which requires careful attention. It is the process of converting user-originated inputs to a computer-based format. The major objective of the input design is to make data entry easy, logical and error free.

In Visual Basic input to the system is entered through forms. A form is “any surface on which information is to be entered, the nature of which is determined by what is already on the surface”. If the data going into the system is incorrect, then processing and output will magnify these errors. So designer should ensure that form is acceptable and understandable by the user.

This application has been developed in a user-friendly manner. The layout of the form is made in such a way that the user will not find any difficulty in going from one field to other by just pressing the tab. During the processing the cursor is placed in the position where the data must be entered.

The user is also provided with an option of selecting an appropriate input from a list of values. Necessary dropdown list boxes and combo boxes are included for necessary fields so that the user need not remember all the data and can just select from it.

Validation is made for each and every data entered. Help messages are also provided whenever the users enter a wrong data into a particular field. This makes the user to understand what is to be entered, moreover whenever an erroneous data is entered the error message is displayed and the user can move to the next field only after entering the correct data.

The clear label for the menus and fields are also provided. Consultations are provided so that a user can view the details of any process at any time.

### Database Design

The activity deals with the design of the database. A key is to determine how the access paths are to be implemented. A physical path is derived from a logical path. The general theme behind database is to handle information as a whole. A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and efficiently.

The general objective is to make information access easy, quick, inexpensive and flexible for the user. Database design is the most critical part of the design phase. An elegantly designed, well-defined database is a strong foundation for the whole system. Files in a relational database are called as tables. Columns of tables represent data and rows represent the records in conventional technology.

Bank Registration: This table contains the bank registration details Contact : This table contains the details of the contact

Feedback: This table contains the details of feedback Payment: This table contains the details of Payment. Product: This table contains the details of Product.

Shopping\_cart: This table contains the details of Shopping cart. User\_details: This table contains the details of the Users

**Table Design**

**FEEDBACK**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Slno | Attribute | type | Constraints | Description |
| 1 | Fid | INT(10) | PRIMARY KEY | Fid |
| 2 | Name | VARCHAR(30) | NOTNULL | Name |
| 3 | Email | VARCHAR(30) | NOTNULL | Email |
| 4 | Mnum | INT (12) | NOTNULL | Mnum |
| 5 | Msgg | VARCHAR(100) | NOTNULL | Msgg |

**PAYMENT**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Slno | Attribute | type | Constraints | Description |
| 1 | Transaction No | INT(30) | PRIMARY KEY | Transaction No |
| 2 | User Id | VARCHAR(30) | NOTNULL | User Id |
| 3 | User Name | VARCHAR(30) | NOTNULL | User Name |
| 4 | User Email | VARCHAR(40) | NOTNULL | User Email |
| 5 | Amount | INT(50) | NOTNULL | Amount |
| 6 | Date | VARCHAR(50) | NOTNULL | Date |

**PRODUCT**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Slno | Attribute | type | Constraints | Description |
| 1 | Ser No | INT(11) | PRIMARY KEY | Ser No |
| 2 | Product Code | VARCHAR(200) | NOTNULL | Product Code |
| 3 | Product Name | VARCHAR(200) | NOTNULL | Product Name |
| 4 | Product Category | VARCHAR(200) | NOTNULL | Product Category |
| 5 | Product Price | VARCHAR(200) | NOTNULL | Product Price |
| 6 | Description | VARCHAR(200) | NOTNULL | Description |
| 7 | Brand | VARCHAR(200) | NOTNULL | Brand |
| 8 | Category | VARCHAR(200) | NOTNULL | Category |
| 9 | Features | VARCHAR(200) | NOTNULL | Features |
| 10 | Platform | VARCHAR(200) | NOTNULL | Platform |
| 11 | Model | VARCHAR(200) | NOTNULL | Model |
| 12 | Type | VARCHAR(200) | NOTNULL | Type |
| 13 | Display | VARCHAR(200) | NOTNULL | Display |
| 14 | Waranty | VARCHAR(200) | NOTNULL | Waranty |
| 15 | Shipping Time | VARCHAR(300) | NOTNULL | Shipping Time |
| 16 | Price | VARCHAR(100) | NOTNULL | Price |
| 17 | Offer Price | VARCHAR(100) | NOTNULL | Offer Price |
| 18 | Save | VARCHAR(200) | NOTNULL | Save |
| 19 | Image 1 | VARCHAR(200) | NOTNULL | Image 1 |

**SHOPPINGCART**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Slno | Attribute | type | Constraints | Description |
| 1 | Ser No | INT(11) | PRIMARY KEY | Ser No |
| 2 | Email | VARCHAR(500) | NOTNULL | Email |
| 3 | Product Code | VARCHAR(200) | NOTNULL | Product Code |
| 4 | Product Name | VARCHAR(200) | NOTNULL | Product Name |
| 5 | Image 1 | VARCHAR(500) | NOTNULL | Image 1 |
| 6 | Quantity | VARCHAR(200) | NOTNULL | Quantity |
| 7 | Price | VARCHAR(200) | NOTNULL | Price |
| 8 | Txn Status | VARCHAR(255) | NOTNULL | Txn Status |
| 9 | Payment | VARCHAR(100) | NOTNULL | Payment |

**USERDETAILS**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Slno | Attribute | type | Constraints | Description |
| 1 | Ser No | INT(11) | PRIMARY KEY | Ser No |
| 2 | Name | VARCHAR(200) | NOTNULL | Name |
| 3 | Email | VARCHAR(200) | NOTNULL | Email |
| 4 | Contact | VARCHAR(200) | NOTNULL | Contact |
| 5 | Password | VARCHAR(200) | NOTNULL | Password |

### Output Design

One of the most important features of a system for users in the output it produces. Output design should improve the system’s relationship with the user and help in decision-making. Considering the future use of output required, and depending on the nature, it is displayed on the monitor for immediate need of obtaining the hard copy.

The objective of output design is to define the controls and format of all printed documents and reports and of screens that will be produced by the system. Computer output isthe most important and direct source of information to the user.

Objectives Of Output Design

1. Design output to serve the intended purpose.
2. Deliver the appropriate quality of output.
3. Choose the right output method.
4. Provide output on time.

Output, generally refers to the results that are generated by the system. The output of the system is designed so as to include a number of reports. Reports reflect the output design**.**

# SYSTEM TESTING AND IMPLEMENTATION

## 4. SYSTEM TESTING AND IMPLEMENTATION

### Testing Methodologies

#### System Testing

System testing includes code testing which examines the logic of the program. Each and every part of the program is checked or executed individually to find out the errors. Once the errors in the program are found out, they are debugged. If wrong data is entered, an error message is displayed on the screen so that the user can correct the data at that time itself.

System testing is actually a series of different tests whose primary purpose is to fully exercise the computer based system. A series of testing are performed for the proposed system before the system is ready for the user acceptance test. A candidate system is subject to variety of tests – volume, stress, recovery, security and usability tests.

The steps in the system testing can be categorized as follows:

* + Unit Testing
	+ Requirement Testing
	+ User Testing
	+ Validation Testing
	+ Integration Testing
	+ User Acceptance Testing

#### Unit Testing

Unit testing focuses on verification efforts on the smallest unit of software design i.e., the module. The unit testing is always white box oriented and the step can be conducted in parallel for modules. The module interface is tested to ensure that information properly flows in and out of the program unit under test. The ‘local data structures ‘ are examined to ensure that data stored temporarily maintains its integrity during all steps in an algorithm execution.” Boundary Conditions” are tested to ensure that the module operates properly at boundaries established to limit or restrict processing. All ‘independent paths ‘ through the control structures are exercised to ensure that all statement in a module have been executed at least once. Finally all “ Error-Handling “ are tested.

#### Requirement Testing

The main aim of this test plan is to see whether the outputs created and inputs were given according to the user requirements and specifications that have been established. This was done in the security department by having the developer as a secondary person and another employee who conducted the actual test. Some Suggestion was made while requirement testing was done that has been incorporated.

#### User Testing

During the testing the tester places the role of the individual who desires to penetrate the system. The tester may attempt to acquire passwords through external clerical means and may attack the system with the custom software design to break down any defenses that have been constructed. The tester may also overwhelm the system thereby denying service to other s and may purposely cause system errors to penetrate during recovery and may browse through insecure data, hoping to find key to system entry.

#### Validation Testing

At the end of user testing, software is completely assembled as a package, interfacing errors have now being uncovered and correcting test begins. Software testing and validation are achieved through a serious black box tests that demonstrate conformity with the requirement.

A plan outlines the classes of tests to be conducted and test procedure defines specific cases that will be used to demonstrate conformity with requirements. Both the plan and the procedure are designed to ensure that all functional requirements are achieved, documentation is correct and other requirements are met. After the validation test, one of the conditions exists.

They are,

The function or performance characteristics confirm to specification and are accepted. A deviation from the specification is uncovered and a deficiency list is created. The deviation or error discovered at this stage in a project can rarely be corrected prior to scheduled completion. It is necessary to negotiate with the customer to establish methods.

#### Integration testing

Integration testing is a systematic technique for constructing the program structure while conducting tests to uncover errors associated with interfacing. The objective is to take unit testing

using “big-bang” approach. All modules are combined in advance. The entire program is tested as a whole. When a set of errors is encountered, correction is difficult because isolation of causes is complicated by the vast expanse of the entire program. Once these errors are corrected new ones appear and the process continues in a seemingly endless loop.

Incremental integration is the antithesis of big-bang approach. The program is constructed and tested in small sequence, where errors are easier to isolate and correct; interfaces are more likely to be tested completely; and a systematic test approach may be applied.

#### User acceptance testing

Acceptance testing involves planning and execution of functional tests, performance test, and stress tests to verify that the implemented system satisfies its requirements. Acceptance tests are typically performed by quality assurance and/or customer organizations. Functional and performance tests are performed to determine the limitations of the system. Typically , acceptance typically, acceptance test will incorporate test cases developed during unit testing and integration testing. Additional test cases are added to achieve desired level of functional , performance, and stress testing of the entire system. Tools of Special importance during acceptance testing include a test Coverage analyzer, a timing analyzer, and a coding standards checker. A test coverage analyzer records the control paths followed for each test case. Timing analyzer reports the time spent in various regions of the source code and different test cases.

**White Box Testing**

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

**Black Box Testing**

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

### System Implementation

#### Implementation Planning:

This section describes about the Implementation of the This application and the details of how to access this control from any application.

Implementation is the process of assuring that the information system is operational and then allowing users take over its operation for use and evaluation. Implementation includes the following activities.

* Obtaining and installing the system hardware.
* Installing the system and making it run on its intended hardware.
* Providing user access to the system.
* Creating and updating the database.
* Documenting the system for its users and for those who will be responsible for maintaining it in the future.
* Making arrangements to support the users as the system is used.
* Transferring ongoing responsibility for the system from its developers to the operations or maintenance part.
* Evaluating the operation and use of the system.

#### Implementation Phase in this project:

This new system has been implemented. The database was put into the MySQL server.. The database is accessible through DriverManager. Documentation is provided well in such a way that it is useful for users and maintainers.

#### Maintenance:

Maintenance is any work done to change the system after it is in operational. The term maintenance is used to describe activities that occur following the delivery of the product to the customer. The maintenance phase of the software life cycle is the time period in which a software product performs useful work.

Maintenance activities involve making enhancements to products, adapting products to new environments, correcting problems.

In this be retrieve the data from the database design by searching the database. So, for maintaining data our project has a backup facility so that there is an additional copy of data, which needs to be maintained.

More over this project would update the annual data on to a CD, which could be used for later reference

#### Methods of implementation:

The four basic methods of implementation are:

* + 1. Parallel system method.
		2. Direct cut over method.
		3. Pilot system method.
		4. Phase in method.

# CONCLUSION AND FUTURE

**ENHANCEMENTS**

## 5. CONCLUSION & FUTURE ENHANCEMENTS

### CONCLUSION

In this **Online Transaction Fraud Detection-Backlogging on E-Commerce Website** management report in python paper we studied the problem of credit card fraud in E-commerce applications. We explored various approaches to solve the problem. The knowledge of various approaches can improve the scope of protecting E-commerce applications. Finally we implemented genetic algorithm for credit card fraud detection. As the adversaries change their means of attack every time, it is important to have constant vigil on the methods they use and update the techniques accordingly.

In this paper build a prototype application in Java platform in order to demonstrate the proof of concept. The management report in python application uses Behavior and Location Analysis (BLA) . The experimental results reveal that the proposed application is useful and can be used in real world systems.

### FUTURE ENHANCEMENTS.

The further work is to find a way to make full use of different resource optimization ability to enhance the web service’s performance and efficiency. Metadata is increasingly used to define the ability to increase discovery reuse and to facilitate interoperability. This analyses a number of possible approaches using Semantic Web standards and outlines how the chosen approach will extend my research in implementing an automated system for the integration of Web-based content.

1. Adding more level of protection.
2. Extends the futures to biomatric

### BIBLIOGRAPHY

**References**

* + 1. A. Shen, R. Tong, and Y. Deng, "Application of classification models on credit card fraud detection," June 2007.
		2. Abhinav Srivastava, Amlan Kundu, Shamik Sural, Arun K. Majumdar, “Credit Card Fraud Detection using Hidden Markov Model,” IEEE Transactions On Dependable And Secure Computing, vol. 5, Issue no. 1, pp.37- 48, January-March 2008.
		3. Aihua Shen, Rencheng Tong, Yaochen Deng, Application of Classification Models on Credit Card Fraud Detection, 2007 IEEE.
		4. Amlan Kundu, Suvasini Panigrahi, Shamik Sural and Arun K. Majumdar, “Credit card fraud detection: A fusion approach using Dempster–Shafer theory and Bayesian learning,” Special Issue on Information Fusion in Computer Security, Vol. 10, Issue no 4, pp.354- 363, October 2009.
		5. CLIFTON PHUA1\*, VINCENT LEE1, KATE SMITH1 & ROSS GAYLER2A Comprehensive Survey of Data Mining-based Fraud Detection Research
		6. Jon T. S. Quah and M. Sriganesh, Real Time Credit Card Fraud Detection using

Computational Intelligence, Proceedings of International Joint Conference on Neural Networks, Orlando, Florida, USA, August 2007.Kou, Y., Lu, C.-T., Sirwongwattana, S., Huang, Y.-P.: Survey of fraud detection techniques. In: Proceedings of the 2004 IEEE International Conference on Networking, Sensing and Control, Taipei, Taiwan (2004).

* + 1. Pengyue J. Lin, Behrokh Samadi, Alan Cipolone, Daniel R. Jeske, Development of a Synthetic Data Set Generator for Building and Testing Information Discovery Systems, Proceedings of the Third International Conference on Information Technology: New Generations (ITNG'06)2006 IEEE.
		2. Tej Paul Bhatla, Vikram Prabhu & Amit Dua “Understanding Credit Card Frauds,” 2003.
		3. Y. Sahin, E. Duman “Detecting Credit Card Fraud by ANN and Logistic Regression” 2011. 10.Raghavendra Patidar, Lokesh Sharma, “Credit Card Fraud Detection Using Neural Network”, International Journal of Soft Computing and Engineering (IJSCE) June 2011.

11. Y. Sahin, E. Duman, Detecting Credit Card Fraud by ANN and Logistic Regression, ©2011I

### APPENDIX

**Source Code**

from django.shortcuts import render import mysql.connector

from django.contrib import messages

from django.shortcuts import render\_to\_response from django.conf import settings

import os

from django.conf import settings

from django.http import HttpResponse, Http404 import mimetypes

from django.core.files.storage import FileSystemStorage

from django.core.mail import send\_mail from django.conf import settings

defhome(request): mydb=mysql.connector.connect(host="localhost",user="root",password="1111",database="home") mycursor=mydb.cursor()

bc="select \* from product where pro\_cat='tv' limit 0,4"print(bc)

cd=mycursor.execute(bc) rows1=mycursor.fetchall() print(rows1)

bc="select \* from product where pro\_cat='music' limit 0,4"print(bc) cd=mycursor.execute(bc)

rows2=mycursor.fetchall() print(rows2)

bc="select \* from product where pro\_cat='camera' limit 0,4" print(bc)

cd=mycursor.execute(bc) rows3=mycursor.fetchall() print(rows3)

bc="select \* from product where pro\_cat='mobile' limit 0,4" print(bc)

cd=mycursor.execute(bc) rows4=mycursor.fetchall() print(rows4)

return render(request,'index.html',{'row1':rows1,'row2':rows2,'row3':rows3,'row4':rows4}) def viewproductbl(request):

rproductcode=request.GET.get("rproductcode") print(rproductcode)

mydb=mysql.connector.connect(host="localhost",user="root",password="1111",database="home") mycursor=mydb.cursor()

bc="select \* from product where product\_code='"+rproductcode+"'"

print(bc) cd=mycursor.execute(bc) rows1=mycursor.fetchall() print(rows1)

return render(request,'product.html',{'row1':rows1}) def signup(request):

return render(request,'signup.html') def signupvalidator(request):

name=request.POST.get("name") email=request.POST.get("email") contact=request.POST.get("contact") password=request.POST.get("pwd")

mydb=mysql.connector.connect(host="localhost",user="root",password="1111",database="home") mycursor=mydb.cursor()

bc="select \* from user\_details where email='"+email+"'" print(bc)

cd=mycursor.execute(bc) rows1=mycursor.fetchone() if rows1!=None:

return render(request,'signup.html',{'message': 'User is Already Exist.'}) else:

try:

mn="INSERT INTO

user\_details(name,email,contact,password)VALUES('"+name+"','"+email+"','"+contact+"','"+password+"') "

print(mn) mycursor.execute(mn) mydb.commit()

return render(request,'signup.html',{'message': 'Successfully Added.'}) except:

return render(request,'signup.html',{'message': 'Some issue. Try after some time.'}) def login(request):

return render(request,'login.html') def loginvalidator(request):

email=request.POST.get("email") password=request.POST.get("pwd") print("email"+email) print("password"+password)

mydb=mysql.connector.connect(host="localhost",user="root",password="1111",database="home") mycursor=mydb.cursor()

bc="select \* from user\_details where email='"+email+"' and password='"+password+"'" print(bc)

cd=mycursor.execute(bc) rows1=mycursor.fetchone() if rows1!=None:

request.session['id']=rows1[0] request.session['name']=rows1[1] request.session['email']=rows1[2] request.session['contact']=rows1

return render(request,'userho.html',{'id':request.session['id'],'name':request.session['name'],'email':request.se ssion['email'],'contact':request.session['contact']})

else:

return render(request,'login.html',{'message':'Invalid credentials'})def userhome(request):

return render(request,'userho.html',{'id':request.session['id'],'name':request.session['name'],'email':request.se ssion['email'],'contact':request.session['contact']})

def myaccount(request):

return render(request,'myaccount.html',{'id':request.session['id'],'name':request.session['name'],'email':reques t.session['email'],'contact':request.session['contact']})

def edit\_profile(request):

return render(request,'editprofile.html',{'id':request.session['id'],'name':request.session['name'],'email':request

.session['email'],'contact':request.session['contact']}) def editprofilevalidator(request):

userno=str(request.session['id']) name=request.GET.get("name") email=request.GET.get("email") contact=request.GET.get("contact")

mydb=mysql.connector.connect(host="localhost",user="root",password="1111",database="home") mycursor=mydb.cursor()

try:

mn="update user\_details set name='"+name+"', email='"+email+"', contact='"+contact+"' where ser\_no="+userno

print(mn) mycursor.execute(mn)

mydb.commit()

return render(request,'editprofile.html',{'id':request.session['id'],'name':name,'email':email,'contact':contact})

except:

return render(request,'editprofile.html',{'id':request.session['id'],'name':name,'email':email,'contact':contact,'m essage':'There is some issue. Try after some time.'})

def changepaswd(request):

return render(request,'changepaswd.html',{'id':request.session['id'],'name':request.session['name'],'email':req uest.session['email'],'contact':request.session['contact']})

rows1=mycursor.fetchall() print(rows1)

return render(request,'adm/update1.html',{'rows1':rows1}) def update2(request):

pcode=request.POST.get("productcode") mydb=mysql.connector.connect(host="localhost",user="root",password="1111",database="home") mycursor=mydb.cursor()

bc="select \* from product where product\_code='"+pcode+"'" print(bc)

cd=mycursor.execute(bc) rows1=mycursor.fetchall() print(rows1)

return render(request,'adm/update2.html',{'rows1':rows1}) def update2handler(request):

pcode=request.POST.get("productcode") pname=request.POST.get("pname")

print("pname"+pname) price=request.POST.get("price") print("price"+price) desc=request.POST.get("desc") print("desc"+desc) brand=request.POST.get("brand") print("brand"+brand) category=request.POST.get("category") print("category"+category) features=request.POST.get("features") print("features"+features) plateform=request.POST.get("plateform") print("plateform"+plateform) model=request.POST.get("model") print("model"+model) type=request.POST.get("type") print("type"+type) display=request.POST.get("display") print("display"+display) waranty=request.POST.get("waranty") print("waranty"+waranty) shipping\_time=request.POST.get("shipping\_time") print("shipping\_time"+shipping\_time) price=request.POST.get("price") print("price"+price)

save=request.POST.get("save") print("save"+save) offer\_price=request.POST.get("offer\_price") print("offer\_price"+offer\_price)

mydb=mysql.connector.connect(host="localhost",user="root",password="1111",database="home") mycursor=mydb.cursor()

try:

mn="Update product set product\_name='"+pname+"',product\_price='"+price+"',description='"+desc+"', brand='"+brand+"',category='"+category+"',features='"+features+"',plateform='"+plateform+"',model='"+ model+"',type='"+type+"',display='"+display+"',waranty='"+waranty+"',shipping\_time='"+shipping\_time+ "',price='"+price+"',offer\_price='"+offer\_price+"',save='"+save+"' where product\_code='"+pcode+"'"

print(mn) mycursor.execute(mn) mydb.commit()

bc="select \* from product where product\_code='"+pcode+"'" print(bc)

cd=mycursor.execute(bc) rows1=mycursor.fetchall() print(rows1)

return render(request,'adm/update2.html',{'rows1':rows1,'message':'Succesfully Updated'})except: bc="select \* from product where product\_code='"+pcode+"'"

print(bc) cd=mycursor.execute(bc) rows1=mycursor.fetchall() print(rows1)

return render(request,'adm/update2.html',{'rows1':rows1,'message':'There is some issue. Try after some time.'})

def delete(request):

return render(request,'adm/delete.html') def delete1(request):

cate=request.POST.get("pcat") mydb=mysql.connector.connect(host="localhost",user="root",password="1111",database="home") mycursor=mydb.cursor()

bc="select \* from product where pro\_cat='"+cate+"'"print(bc) cd=mycursor.execute(bc)

rows1=mycursor.fetchall() print(rows1)

return render(request,'adm/delete1.html',{'rows1':rows1}) def delp(request):

productno=request.POST.get("productno") cate=request.POST.get("pcat")

mydb=mysql.connector.connect(host="localhost",user="root",password="1111",database="home") mycursor=mydb.cursor()

try:

bc="delete from product where ser\_no='"+productno+"'" print(bc)

cd=mycursor.execute(bc) mydb.commit()

bc="select \* from product where pro\_cat='"+cate+"'" print(bc)

cd=mycursor.execute(bc) rows1=mycursor.fetchall() print(rows1)

return render(request,'adm/delete1.html',{'rows1':rows1,'message':'Deleted'}) except:

bc="select \* from product where pro\_cat='"+cate+"'" print(bc)

cd=mycursor.execute(bc) rows1=mycursor.fetchall() print(rows1)

return render(request,'adm/delete1.html',{'rows1':rows1,'message':'There is some issue. Try after some time.'})

### Sample Screens



**Fig. Home**



**Fig. Product Details**



**Fig. Signin**



**Fig. Register**



**Fig. User home**



**Fig. Edit Profile**



**Fig. Change password**



**Fig. View Cart**



**Fig. Feedback**



**Fig. Admin Login**



**Fig. Admin Home**



**Fig. View Customer Details**



**Fig. View Payment Details**



**Fig. View Feedback**



**Fig. Handling Products**

### Reports



**Fig. b**



**Fig. b**



**Fig. b**



**Fig. b**