**GNIP LOGO**

**GURU NANAK INSTITUTIONS TECHNICAL CAMPUS**

School of Engineering and Technology

## Department of Computer Science & Engineering

**CERTIFICATE**

This is to certify that this project report entitled **“APPLICATION FOR GST INVOICE USING CLOUD****”** being submitted by **BUKKA NIHAL(15WJ1A0545), AIMREDDY PREETHI(16WJ5A0501), UPPULA BHARATH(16WJ8A0501)** in partial fulfillment for the award of the Degree of **Bachelor of Technology** in **Department of** **Computer Science and Engineering** of the **Jawaharlal Nehru Technological University, Hyderabad** during the academic year 2018-2019, is a record of bonOfide work carried out under my guidance and supervision.

**Mr. A. Ravi Mr. A. Ravi Dr. J. Rajeshwar**

**INTERNAL GUIDE PROJECT CO-ORDINATOR HOD-CSE**

**EXTERNAL EXAMINER**

**ACKNOWLEDGEMENT**

We wish to express our sincere thanks to **Dr. Rishi Sayal,** Associate Director, GNITC for providing us the conductive environment for carrying through our academic schedules and Project with ease.

We have been truly blessed to have a wonderful adviser **Dr. J.Rajeshwar,** HOD of CSE, GNITCfor guiding us to explore the ramification of our work and we express my sincere gratitude towards him for leading me thought the completion of Project.

We would like to say our sincere thanks to **MR. A.Ravi**, Associate Professor, Department of CSE, Major Project Coordinator, for providing seamless support and right suggestions are given in the development of the project.

We specially thank our internal guide **Mr. A.Ravi,** Associate Professor, Department of CSE, for her suggestions and constant guidance in every stage of the project.

Finally, i would like to thank my family members and faculty members for their moral support and encouragement to achieve goals.

**BUKKA NIHAL (15WJ1A0545)**

**AIMREDDY PREETHI (16WJ5A0501)**

**UPPULA BHARATH (16WJ8A0501)**

**ABSTRACT**

The GST accounting software is an interface or an application which makes the calculation of taxes easy for the owners, warehouse operators, and transporters. GST is to be paid monthly, quarterly or yearly depending on business turnover. A GST billing software aids in an accurate calculation and helps in generation of GST transactions and invoices according to the GST format. GST software also helps in tracking and generation of an e-way bill which is to be paid when goods cross state borders. It is a tax return software that helps in calculating and filing GST returns. It ensures an easy, quick and error-free GST calculation. we are using open stack for the host purpose in cloud platform.

Many countries across the world have single unified GST system but due to non-consensus between central and state government of India, India shall adopt a Dual GST model, meaning that the GST would be administered both by the Central and the State Governments. A Dual GST will be levied on the taxable value of every transaction of supply of goods and services

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURE NO** | **NAME OF FIGURE** | **PAGE NO** |
| 1.5.1 | OPEN STACK LANDSCAPE | 07 |
| 4.2.1 | DFD DIAGRAM | 25 |
| 4.2.2 | CLASS DIAGRAM | 26 |
| 4.2.3 | ACTIVITY DIAGRAM | 27 |
| 4.2.4 | USECASE DIAGRAM | 28 |
| 4.2.5 | COMPONENT DIAGRAM | 29 |
| 4.2.6 | DEPLOYMENT DIAGRAM | 30 |
| 5.4.1 | LOGIN PAGE | 57 |
| 5.4.2 | ADMIN LOGIN PAGE | 58 |
| 5.4.3 | PRODUCT PAGE | 58 |
| 5.4.4 | ADDING PRODUCT PAGE | 59 |
| 5.4.5 | DELETING PRODUCT PAGE | 59 |
| 5.4.6 | ADMIN USER PAGE | 60 |
| 5.4.7 | ADDING USER PAGE | 61 |
| 5.4.8 | DELETING USER PAGE | 61 |
| 5.4.9 | INVOICE PAGE | 62 |
| 5.4.10 | ABOUT GST | 62 |
| 5.4.11 | CONTACT PAGE | 63 |
| 6.4 | BLOCK BOX TESTING | 66 |
| 6.5 | WHITE BOX TESTING | 67 |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **S.no** | **Table Name** | **Page no** |
| 1 | Test Case1 | 70 |
| 2 | Test Case2 | 71 |
| 3 | Test Case3 | 72 |

**LIST OF ABBREVIATIONS**

|  |  |
| --- | --- |
| GST | Goods service tax |
| SWOT | Strengths Weakness Opportunities Threats |
| VAT | Values Added Tax |
| CST | Central sales tax |
| Paas | Platform as a service |
| RHV | Red Hat virtualization |
| API | Application Program Interface |
| GDP | Gross Domestic Product |
| ENU | N-ethyl-N-nitrosourea |
| S3 | Simple storage service |
| URL | Uniform Resource Locator |
| PHP | HyperText PreProcessor |
| SQL | Structured Query Language |
| CLI | Command Line Interface |
| HTML | Hyper Text Markup Language |
| CGI | Common Gateway Interface |
| IIS | Internet Information Service |
| ICU | International Components for Unicod |
| JIT | just-in-time |
| SAPIs | server application programming interfaces |
| DFD | Data Flow Diagram |
| UML | Unified Modeling Language |
| SDLC | Software Development Life Cycle |

**CHAPTER 1**

**INTRODUCTION**

The GST software is used to download invoice containing information about the tax related vat that are imposed by the government. Hosting this on open stack will provide the distributed access to the user and reduces the CapEx expenditures. The following will describe the purpose and scope of our system.

# PURPOSE

The Goods and Service Tax (GST) is a comprehensive value added Tax on the supply of Goods and services. GST will replace all the indirect taxes (like Excise duty, VAT, CST etc.) levied on goods and services by Government once it is implemented. The main motive of GST is to reduce the cascading effect of tax on the cost of goods and services and create a common, cooperative and undivided Indian market to make economy stronger and powerful. So the GST system will combine Central excise duty, additional excise duty, service tax, State VAT entertainment tax etc. under one banner. It will impact tax structure, tax incidence, tax computation, credit utilization and reporting, leading to a complete overhaul of the current indirect tax system. This is one of the biggest taxation reform that will take place in India once it is officially passed by Government.

Many countries across the world have single unified GST system but due to non-consensus between central and state government of India, India shall adopt a Dual GST model, meaning that the GST would be administered both by the Central and the State Governments. A Dual GST will be levied on the taxable value of every transaction of supply of goods and services.

**Key Features of the GST Accounting Software**

Invoice Generation as per the GST standards

E-Filing for the Returns and Return Reconciliation

E-Payment Facility

Tax Computation and Validation

Client Management

Data import

ITC Match/Mismatch Report Generation

Credit Management

Budget and Controls

# 1.2 SCOPE

GST is one indirect tax for the whole nation, which will make India one unified common market. It is a single tax on the supply of goods and services, right from the manufacturer to the consumer. Credits of input taxes paid at each stage will be available in the subsequent stage of value addition, which makes GST essentially a tax only on value addition at each stage. The final consumer will thus bear only the GST charged by the last dealer in the supply chain, with setoff benefits at all the previous stages.

The benefits of GST can be summarized as under:

Which taxes at the Centre and State level are being subsumed into GST?

## For business and industry

Easy compliance: A robust and comprehensive IT system would be the foundation of the GST regime in India. Therefore, all tax payer services such as registrations, returns, payments, etc. would be available to the taxpayers online, which would make compliance easy and transparent.

**Uniformity of tax rates and structures:** GST will ensure that indirect tax rates and structures are common across the country, thereby increasing certainty and ease of doing business. In other words, GST would make doing business in the country tax neutral, irrespective of the choice of place of doing business.

**Removal of cascading:** A system of seamless tax credits throughout the value chain, and across boundaries of States, would ensure that there is minimal cascading of taxes. This would reduce hidden costs of doing business.

**Improved competitiveness:** Reduction in transaction costs of doing business would eventually lead to an improved competitiveness for the trade and industry.

**Gain to manufacturers and exporters**: The subsuming of major Central and State taxes in GST, complete and comprehensive setoff of input goods and services and phasing out of Central Sales Tax (CST) would reduce the cost of locally manufactured goods and services. This will increase the competitiveness of Indian goods and services in the international market and give boost to Indian exports. The uniformity in tax rates and procedures across the country will also go a long way in reducing the compliance cost.

## For Central and State Governments

**Simple and easy to administer:** Multiple indirect taxes at the Central and State levels are being replaced by GST. Backed with a robust endtoend IT system, GST would be simpler and easier to administer than all other indirect taxes of the Centre and State levied so far.

**Better controls on leakage:** GST will result in better tax compliance due to a robust IT infrastructure. Due to the seamless transfer of input tax credit from one stage to another in the chain of value addition, there is an inbuilt mechanism in the design of GST that would incentivize tax compliance by traders.

**Higher revenue efficiency:** GST is expected to decrease the cost of collection of tax revenues of the Government, and will therefore, lead to higher revenue efficiency.

## For the consumer

**Single and transparent tax proportionate to the value of goods and services:** Due to multiple indirect taxes being levied by the Centre and State, with incomplete or no input tax credits available at progressive stages of value addition, the cost of most goods and services in the country today are laden with many hidden taxes.

# CHAPTER 2

# LITERATURE SURVEY

A literature Survey is a text of a scholarly paper, which includes the current knowledge including substantive findings as well as theoretical and methodological contributions to a particular topic. Literature reviews are secondary sources and do not report new or original experimental work. Most often associated with academic-oriented literature, such reviews are found in academic journals, and are not to be confused with books reviews C.A narrow-scope literature review may be included as a part of a peer-reviewed journal article presenting new research, serving to situate the current study within the body of the relevant literature and to provide context for the reader .In such a case ,the review usually precedes the methodology and results sections of the work. Producing a literature review may also be part of graduate and postgraduate student work, including in the preparation of a thesis ,dissertation ,or a journal article.

## 2.1 REVIEW OF RELATED LITERATURE FROM RELATED

### RESEARCH PAPERS

**Title:** GST in India Concept and SWOT Analysis

**Author:** Bharti Sharma,Manmohan Singh,Manjeet Kharub.

#### Year: 2018

**Description:** The introduction of Goods and Services Tax (GST) is a very significant step in the field of indirect tax reforms in India. By amalgamating a large number of Central and State taxes into a single tax, it mitigates the cascading or double taxation in a major way and pave the way for a common national market. From the consumer point of view, the biggest advantage is in terms of a reduction in the overall tax burden on goods, which was previously estimated to be around 25%-30%. This paper presents the basic concept of GST and a SWOT (Strengths, Weaknesses, Opportunity, and Threats) analysis. History of GST, benefits, shortcomings, Government policies on GST along with SWOT analysis have been provided in this paper, which may be helpful for the businessmen and the layman to understand GST in broader way.

**Title:** A Comprehensive Analysis of Goods And Services Tax(GST)

**Author:** Anand Nayyar, Inderpal Singh.

**Year:** 2018

**Description:** The Goods and Services Tax (GST), implemented on July 1, 2017, is regarded as a major taxation reform till date implemented in India since independence in 1947. GST was planned to be implemented in April 2010, but was postponed due to political issues and conflicting interest of stakeholders. The primary objective behind development of GST is to subsume all sorts of indirect taxes in India like Central Excise Tax, VAT/Sales Tax, Service tax, etc. and implement one taxation system in India. The GST based taxation system brings more transparency in taxation system and increases GDP rate from 1% to 2% and reduces tax theft and corruption in country. The paper highlighted the background of the taxation system, the GST concept along with significant working, comparison of Indian GST taxation system rates with other world economies, and also presented in-depth coverage regarding advantages to various sectors of the Indian economy after levising GST and outlined some challenges of GST.

**Title:** GST billing system

**Author:** Apurva Rewatkar,Akshata birelliwar,Hemalata R.kosare.

#### Year: 2018

**Description:** GST is characterized as the bigger backhanded expense structure intended to help and upgrade the financial development of a country. It would enthusiasm to comprehend why this proposed GST administration may hamper the development and advancement of the nation.

**Title:** IMPACT OF GST ON WAREHOUSING INDUSTRY

**Author:** Deepak Mittal

#### Year: 2018

**Description:** India has now implemented most important tax reform GST. More than 160 nations have already adopted a unified indirect tax structure in Asia which creating an effective tax system with a comparatively lower cost of administration and collection due to multiple tax rates at the state level makes the system inefficient now remove all state tax barriers and creating a efficient tax system. Storage and warehousing of agricultural produce has been exempted from GST, clearly with the intent to reduce the tax burden on the farming sector. This is a continuation of the earlier exemption of service tax for these services.

**Title:** Reduced Anthocyanins in Petioles codes for a GST anthocyanin transporter that is essential for the foliage and fruit coloration in strawberry.

**Author:** HuifengLuo, ChengDai, Yongping Li,Jia Feng, Zhongchi Liu and

Chunying Kang

**Year:** 2018

**Description:** The red color of the foliage and fruit in strawberry comes from anthocyanins stored in the vacuole; however, how this anthocyanin accumulation is regulated remains unclear. A reduced anthocyanin in petioles (rap) mutant was identified in an N-ethyl-N-nitrosourea (ENU) mutagenized population of YW5AF7, a white-fruited variety of the wild strawberry Fragaria vesca. The causative mutation was identified to be a premature stop codon in a glutathione S-transferase (GST) gene. In addition to the foliage coloration, RAP also mediates fruit pigmentation and acts downstream of the fruit-specific transcription factor FvMYB10. Among all eight GST genes in the same subfamily, RAP is most abun- dantly expressed in the ripening fruit. Expression analysis and transient expression assays demonstrated that RAP is the principal transporter of anthocyanins among the paralogs. Moreover, domain-swap experiments showed that both the N- and C-terminals of RAP are essential for the binding capability of anthocyanins. In addition, transient knock-down of RAP resulted in reduced fruit coloration in cultivated strawberry. Collectively, our results demonstrate that RAP encodes the principal GST transporter of anthocyanins in the strawberry foliage and fruit,and itt could be modified to alter the fruit color in strawberry.

# CHAPTER 3

**SYSTEM REQUIREMENTS AND SPECIFICATIONS**

A software requirements specification (SRS) is a description of a [software system](https://en.wikipedia.org/wiki/Software_system) to be [developed](https://en.wikipedia.org/wiki/Developed). It is modeled after [business requirements specification](https://en.wikipedia.org/wiki/Business_requirements)  also known as a [stakeholder requirements specification (StRS)](https://en.wikipedia.org/w/index.php?title=Stakeholder_requirements_specification_(StRS)&action=edit&redlink=1). The software requirements specification lays out [functional](https://en.wikipedia.org/wiki/Functional_requirement) and [non-functional requirements](https://en.wikipedia.org/wiki/Non-functional_requirements), and it may include a set of [use cases](https://en.wikipedia.org/wiki/Use_case) that describe user interactions that the software must provide to the user for perfect interaction.

## 3.1 PROJECT REVIEW

## 3.1.1 EXISTING SYSTEM

The existing system does not support the distributed access and does not have any secured. This system does not have recovery methodologies. There should be a manual update for any website

**Drawbacks of Existing Algorithm are:**

Scalability

No high availability

Capex expenditures

## 3.1.2 PROPOSED SYSTEM

The proposed system will be the system which is used to provide the Paas in which the news are dynamically accesssed by the users and the host managers need not manage the site every time.Distributed access to the GST will be provided on the cloud open stack for hosting purpose.

**FEATURES**

**Admin login**: This allows only the admin to login.

**User login**: This allows only the registered users to login.

**Product details**: This module helps the admin and user to insert all the product details which are available near the user and admin.The products can be deleted by user and admin.User and admin can also edit product details.

**User details**: This module helps to maintain data of users and aslo admin can check whether the users are in active mode or inactive mode.Admin can delete users when ever they are not required. Admin can also edit user details if required.

**Invoice:** Admin and User can access this page to view or download the previous invoices.

**About GST:** Admin and User can access this page to learn about GST.

**Contact page:** If their is any queries then user can contact admin by viewing details in contact page.

**Front End:** PHP

**ADVANTAGES**

Distributed access to the GST will be provided on the cloud open stack for hosting purpose.

The system which is used to provide the Paas in which the news are dynamically accesssed by the users and the host managers need not manage the site every time.

**ADVANTAGES OF HOSTING YOUR STATIC WEBSITE ON OPEN STACK**

There is an increasing trend nowadays where many organization use Cloud based storage services, like Amazon S3 for hosting a complete static website. Using OPENSTACK this can be done by following some simple configurations steps. In S3 console, you create a bucket, enable web hosting feature and provide public re ad access. You can upload your static file s directly or using any S3 client. Once you have done all configurations in S3, you are provided with a link that you can directly launch in browser to visit the website . This URL can also be mapped to a domain name using Route53. In this blog, we are going to discuss some of the advantages and disadvantages to this approach.

* No CAPEX expenditures

S3 is fully responsible for serving your content to the visitors and completely eliminates the need of purchasing hosting servers or launching virtual machines and managing web server.

* Scale up automatically

Without making any change s to your initial setup OPENSTACK will automatically scale up the infrastructure to meet the growing demand.

* High Availability

OPENSTACK guarantees 99.99% availability of S3 which means there is almost no chance of losing your data. S3 achieves this by replication across multiple data centers

* Fast Content Serving

Decreases the latency and resulting in optimum response time.

## DISADVANTAGES

**•** The system is not fully automated, it needs to be checked for services else unexpected cost may get incurred.

## APPLICATION

• This project can be used by all the retailers and wholesalers to update their GST invoice and will provide the user with tax details.

### 3.2 SOFTWARE AND HARDWARE REQUIREMENTS

#### 3.2.1 SOFTWARE REQUIREMENTS

The software requirements document is the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide a `basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the teams and tracking the team’s progress throughout the development activity.

* Platform : Openstack Dashboard
* Coding interface : PHP
* Backend : SQL

#### 3.2.2 HARDWARE COMPONENTS

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system.

Hard disk : 40 GB

RAM : 2GB(minimum)

## 3.3 FEASIBILITY ANALYSIS

Feasibility of the project is analyzed in this phase and 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 to be carried out. This is to ensure that the purpose of the system is not burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

### 3.3.1 TECHNICAL FEASIBILITY

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

### TECHNICAL REQUIREMENTS

This chapter is about the software language and the tools used in the development of the project. The Scripting language used here is PHP.

**3.3.2 FEATURES OF PHP.**

Hypertext Preprocessor (or simply PHP) is a general-purpose programming language originally designed for web development. It was originally created by Rasmus Lerdorf in 1994;the PHP reference implementation is now produced by The PHP Group.PHP originally stood for Personal Home Page, but it now stands for the recursive initialism PHP: Hypertext Preprocessor.

PHP code may be executed with a command line interface (CLI), embedded into HTML code, or it can be used in combination with various web template systems, web content management systems, and web frameworks. PHP code is usually processed by a PHP interpreter implemented as a module in a web server or as a Common Gateway Interface (CGI) executable. The web server combines the results of the interpreted and executed PHP code, which may be any type of data, including images, with the generated web page. PHP can be used for many programming tasks outside of the web context, such as standalone graphical applications and robotic drone control.

The standard PHP interpreter, powered by the Zend Engine, is free software released under the PHP License. PHP has been widely ported and can be deployed on most web servers on almost every operating system and platform, free of charge.

The PHP language evolved without a written formal specification or standard until 2014, with the original implementation acting as the de facto standard which other implementations aimed to follow. Since 2014, work has gone on to create a formal PHP specification.

#### 3.3.3 History of PHP

PHP development began in 1994 when Rasmus Lerdorf wrote several Common Gateway Interface (CGI) programs in C,which he used to maintain his personal homepage. He extended them to work with web forms and to communicate with databases, and called this implementation "Personal Home Page/Forms Interpreter" or PHP/FI.

PHP/FI could be used to build simple, dynamic web applications. To accelerate bug reporting and improve the code, Lerdorf initially announced the release of PHP/FI as "Personal Home Page Tools (PHP Tools) version 1.0" on the Usenet discussion group comp.infosystems.www.authoring.cgi on June 8, 1995.This release already had the basic functionality that PHP has today. This included Perl-like variables, form handling, and the ability to embed HTML. The syntax resembled that of Perl, but was simpler, more limited and less consistent.

Early PHP was not intended to be a new programming language, and grew organically, with Lerdorf noting in retrospect: "I don't know how to stop it, there was never any intent to write a programming language [...] I have absolutely no idea how to write a programming language, I just kept adding the next logical step on the way."A development team began to form and, after months of work and beta testing, officially released PHP/FI 2 in November 1997.

The fact that PHP was not originally designed, but instead was developed organically has led to inconsistent naming of functions and inconsistent ordering of their parameters. In some cases, the function names were chosen to match the lower-level libraries which PHP was "wrapping",while in some very early versions of PHP the length of the function names was used internally as a hash function, so names were chosen to improve the distribution of hash values.

##### PHP 3 and 4

Zeev Suraski and Andi Gutmans rewrote the parser in 1997 and formed the base of

PHP 3, changing the language's name to the recursive acronym PHP: Hypertext Preprocessor.Afterwards, public testing of PHP 3 began, and the official launch came in June 1998. Suraski and Gutmans then started a new rewrite of PHP's core, producing the Zend Engine in 1999.They also founded Zend Technologies in Ramat Gan, Israel.

On May 22, 2000, PHP 4, powered by the Zend Engine 1.0, was released.As of August 2008 this branch reached version 4.4.9. PHP 4 is no longer under development nor will any security updates be released.

##### PHP 5

On July 14, 2004, PHP 5 was released, powered by the new Zend Engine II.PHP 5

included new features such as improved support for object-oriented programming, the PHP Data Objects (PDO) extension (which defines a lightweight and consistent interface for accessing databases), and numerous performance enhancements.In 2008, PHP 5 became the only stable version under development. Late static binding had been missing from PHP and was added in version 5.3.

Many high-profile open-source projects ceased to support PHP 4 in new code as of February 5, 2008, because of the GoPHP5 initiative, provided by a consortium of PHP developers promoting the transition from PHP 4 to PHP 5.

Over time, PHP interpreters became available on most existing 32-bit and 64-bit operating systems, either by building them from the PHP source code, or by using pre-built binaries.For PHP versions 5.3 and 5.4, the only available Microsoft Windows binary distributions were 32-bit x86 builds,requiring Windows 32-bit compatibility mode while using Internet Information Services (IIS) on a 64-bit Windows platform. PHP version 5.5 made the 64-bit x86-64 builds available for Microsoft Windows.

Official security support for PHP 5.6 ended on 31 December 2018,but Debian 8.0 Jessie will extend support until June 2020.

###### PHP 6 and Unicode

PHP received mixed reviews due to lacking native Unicode support at the core language level.In 2005, a project headed by Andrei Zmievski was initiated to bring native Unicode support throughout PHP, by embedding the International Components for Unicode (ICU) library, and representing text strings as UTF-16internally.Since this would cause major changes both to the internals of the language and to user code, it was planned to release this as version 6.0 of the language, along with other major features then in development.

However, a shortage of developers who understood the necessary changes, and performance problems arising from conversion to and from UTF-16, which is rarely used in a web context, led to delays in the project.As a result, a PHP 5.3 release was created in 2009, with many non-Unicode features back-ported from PHP 6, notably namespaces. In March 2010, the project in its current form was officially abandoned, and a PHP 5.4 release was prepared containing most remaining non-Unicode features from PHP 6, such as traits and closure re-binding.Initial hopes were that a new plan would be formed for Unicode integration, but as of 2014none had been adopted.

##### PHP 7

During 2014 and 2015, a new major PHP version was developed, which was numbered PHP 7. The numbering of this version involved some debate.While the PHP 6 Unicode experiment had never been released, several articles and book titles referenced the PHP 6 name, which might have caused confusion if a new release were to reuse the name.After a vote, the name PHP 7 was chosen.

The foundation of PHP is a PHP branch that was originally dubbed PHP next generation (phpng). It was authored by Dmitry Stogov, Xinchen Hui and Nikita Popov,and aimed to optimize PHP performance by refactoring the Zend Engine while retaining near-complete language compatibility.As of 14 July 2014, WordPress-based benchmarks, which served as the main benchmark suite for the phpng project, showed an almost 100% increase in performance. Changes from phpng are also expected to make it easier to improve performance in the future, as more compact data structures and other changes are seen as better suited for a successful migration to a just-in-time (JIT) compiler.Because of the significant changes, the reworked Zend Engine is called Zend Engine 3, succeeding Zend Engine 2 used in PHP 5.

Because of major internal changes in phpng it must receive a new major version number of PHP, rather than a minor PHP 5 release, according to PHP's release process.Major versions of PHP are allowed to break backward-compatibility of code and therefore PHP 7 presented an opportunity for other improvements beyond phpng that require backward-compatibility breaks. In particular, it involved the following changes:

* Many fatal- or recoverable-level legacy PHP error mechanisms were replaced with modern object-oriented exceptions.
* The syntax for variable dereferencing was reworked to be internally more consistent and complete, allowing the use of the operators ->, [], (),{}, and ::, with arbitrary meaningful left-side expressions.
* Support for legacy PHP 4-style constructor methods was deprecated.
* The behavior of the foreach statement was changed to be more predictable.
* Constructors for the few classes built-in to PHP which returned null upon failure were changed to throw an exception instead, for consistency.
* Several unmaintained or deprecated server application programming interfaces (SAPIs) and extensions were removed from the PHP core, most notably the legacy mysql extension.
* The behavior of the list() operator was changed to remove support for strings
* Support was removed for legacy ASP-style delimiters <% and %> and <script language="php"> ... </script>
* An oversight allowing a switch statement to have multiple default clauses was fixed.
* Support for hexadecimal number support in some implicit conversions from strings to number types was removed.
* The left-shift and right-shift operators were changed to behave more consistently across platforms.
* Conversions between integers and floating point numbers were tightened and implemented more consistently across platforms.

**PHP 7** also included new language features. Most notably, it introduces return type declarations for functions which complement the existing parameter type declarations, and support for the scalar types (integer, float, string, and boolean) in parameter and return type declarations.

#### 3.3.4 WHY SOFTWARE DEVELOPERS CHOOSE PHP

PHP is an excellent choice for huge and powerful websites and most of the popular and leading websites make use of the PHP scripting language. PHP is a server-side scripting language, like ASP and JSP. It is an open source and free to download and use. PHP is a powerful scripting language for designing the dynamic websites. PHP files will consist of text, markup language tags, and scripts. PHP files also have a file extension like .php, .php3, and .phtml.

PHP will be able to run on different platforms like Windows, Linux, etc. PHP is simple and fast to learn and also runs well on the server side. PHP is a popular and widely used free and good alternative to the competitors like Java and JSP. PHP will be well suitable for web development and can be included directly into the HTML code. PHP is commonly used along with Apache web server on multiple OS. And moreover, PHP is compatible with all the servers used nowadays.

**3.3.5 OPERATION FEASIBILITY**

The operation staff in the organization feasibility. The employees of the concerned organization is supportive enough to implement the proposed system. Hence, it is operationally feasible. Therefore, the proposed system is feasible in all aspects. Hence, it is encouraging to undertaking a detailed system analysis.

**3.3.6 ECONOMIC FEASIBILITY**

The study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is omitted. The expenditure must be justified. Thus, the development system as well as within the budget and this was achieved because most of the technologies used are freely available. Only customized product bad to be purchased.

**3.4 FUNCTIONAL REQUIREMENTS**

In [software engineering](https://en.wikipedia.org/wiki/Software_engineering) and [systems engineering](https://en.wikipedia.org/wiki/Systems_engineering), a functional requirement defines a function of a [system](https://en.wikipedia.org/wiki/System) or its component, where a function is described as a specification of behaviour between outputs and inputs. Functional requirements may involve calculations, technical details, data manipulation and processing, and other specific functionality that define what a system is supposed to accomplish. Behavioral requirements describe all the cases where the system uses the functional requirements, these are captured in [use cases](https://en.wikipedia.org/wiki/Use_case). Functional requirements are supported by [non-functional requirements](https://en.wikipedia.org/wiki/Non-functional_requirement) (also known as "quality requirements"), which impose constraints on the design or implementation (such as performance requirements, security, or reliability).

## 3.5 NON-FUNCTIONAL REQUIREMENTS

In [systems engineering](https://en.wikipedia.org/wiki/Systems_engineering) and [requirements engineering](https://en.wikipedia.org/wiki/Requirements_engineering), a non-functional requirement (NFR) is a [requirement](https://en.wikipedia.org/wiki/Requirement) that specifies criteria that can be used to judge the operation of a system, rather than specific behaviour’s. They are contrasted with [functional requirements](https://en.wikipedia.org/wiki/Functional_requirement) that define specific behaviour or functions. The plan for implementing functional requirements is detailed in the [system design](https://en.wikipedia.org/wiki/Systems_design). The plan for implementing non-functional requirements is detailed in the [system architecture](https://en.wikipedia.org/wiki/Systems_architecture), because they are usually [architecturally significant requirements](https://en.wikipedia.org/wiki/Architecturally_significant_requirements).

# CHAPTER 4

# SYSTEM DESIGN

Systems design is the process of defining the [architecture](https://en.wikipedia.org/wiki/Systems_architecture), modules, interfaces, and [data](https://en.wikipedia.org/wiki/Data) for a [system](https://en.wikipedia.org/wiki/System) to satisfy specified [requirements](https://en.wikipedia.org/wiki/Requirement). Systems design could be seen as the application of [systems theory](https://en.wikipedia.org/wiki/Systems_theory) to [product development](https://en.wikipedia.org/wiki/Product_development). There is some overlap with the disciplines of [systems analysis](https://en.wikipedia.org/wiki/Systems_analysis), [systems architecture](https://en.wikipedia.org/wiki/Systems_architecture) and [systems engineering](https://en.wikipedia.org/wiki/Systems_engineering)

### 4.1 SYSTEM ARCHITECHTURE

A system architecture or systems architecture is the [conceptual model](https://en.wikipedia.org/wiki/Conceptual_model) that defines the [structure](https://en.wikipedia.org/wiki/Structure), [behavior](https://en.wikipedia.org/wiki/Behavior), and more [views](https://en.wikipedia.org/wiki/View_model) of a [system](https://en.wikipedia.org/wiki/System).[[1]](https://en.wikipedia.org/wiki/Systems_architecture#cite_note-1) An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the [structures](https://en.wikipedia.org/wiki/Structure) and [behaviors](https://en.wikipedia.org/wiki/Behavior) of the system.

A system architecture can consist of system [components](https://en.wikipedia.org/wiki/System) and the sub-systems developed, that will work together to implement the overall system. There have been efforts to formalize languages to describe system architecture, collectively these are called [architecture description languages](https://en.wikipedia.org/wiki/Architecture_description_languages) (ADLs).

Diagram

Description automatically generated

**Fig: 4.1 System Architecture**

### 4.2.1 DATA FLOW DIAGRAM

A data flow diagram (DFD) is a graphical representation of the “flow” of data through an information system. It differs from the flowchart as it shows the data flow instead of the control flow of the program. A data flow diagram can also be used for the visualization of data processing. The DFD is designed to show how a system is divided into smaller portions and to highlight the flow of data between those parts.

SETTINGS

ENTRY DETAILS

LOGIN DETAILS

ADD CLIENT DETAILS

MAKE IT AVAILABLE ON OPENSTACK

CALCULATE

GENERATE INVOICES

**Fig: 4.2.1 Data Flow diagram**

#### 4.2.2 Class Diagram

The class diagram is the main building block of object-oriented modeling. It is used for general conceptual modeling of the structure of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling. The classes in a class diagram represent both the main elements, interactions in the application, and the classes to be programmed.

Diagram

Description automatically generated

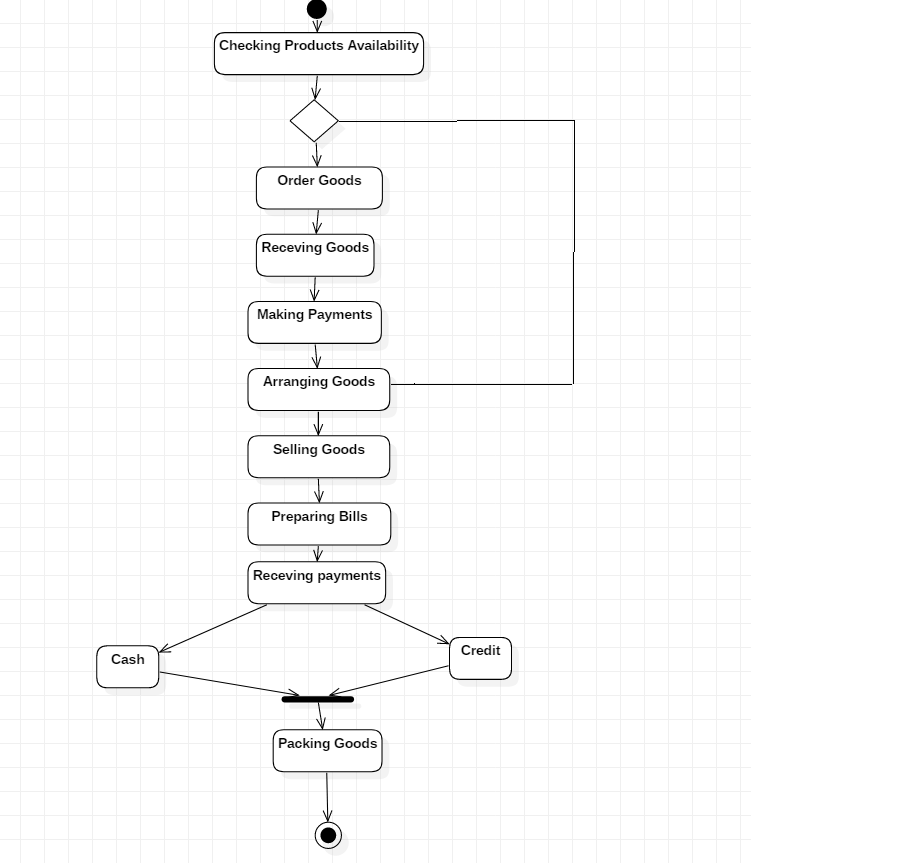
**Fig: 4.2.2 class diagram**

**DESCRIPTION**

The above diagram represents the class diagram of a GST invoice application.

### 4.2.3 Activity diagram

Activity diagram are a loosely defined diagram to show workflows of stepwise activities and actions, with support for choice, iteration and concurrency. UML, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. UML activity diagrams could potentially model the internal logic of a complex operation. In many ways UML activity diagrams are the object-oriented equivalent of flow charts and data flow diagrams (DFDs) from structural development.



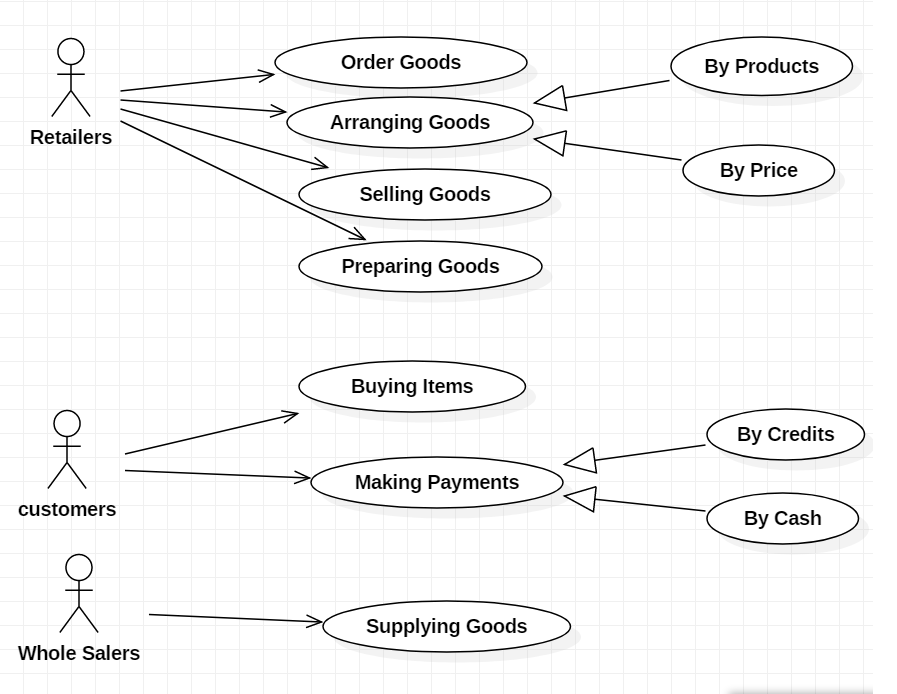
**Fig: 4.2.3 Activity diagram**

#### DESCRIPTION

The above activity diagram shows how the the flow of data takes place between different nodes and subnodes in the system.

### 4.2.4 Usecase diagram

A use case diagram is a type of behavioral diagram created from a Use-case analysis. The purpose of use case is to present overview of the functionality provided by the system in terms of actors, their goals and any dependencies between those use cases.



**Fig: 4.2.4 usecase diagram**

#### DESCRIPTION

The above usecase diagram shows how the admin and user relates their action with different modules of the system.

### 4.2.5 Component diagram

Component diagrams are different in terms of nature and behavior. Component diagrams are used to model the physical aspects of a system. Now the question is, what are these physical aspects? Physical aspects are the elements such as executables, libraries, files, documents, etc. which reside in a node.

Diagram

Description automatically generated

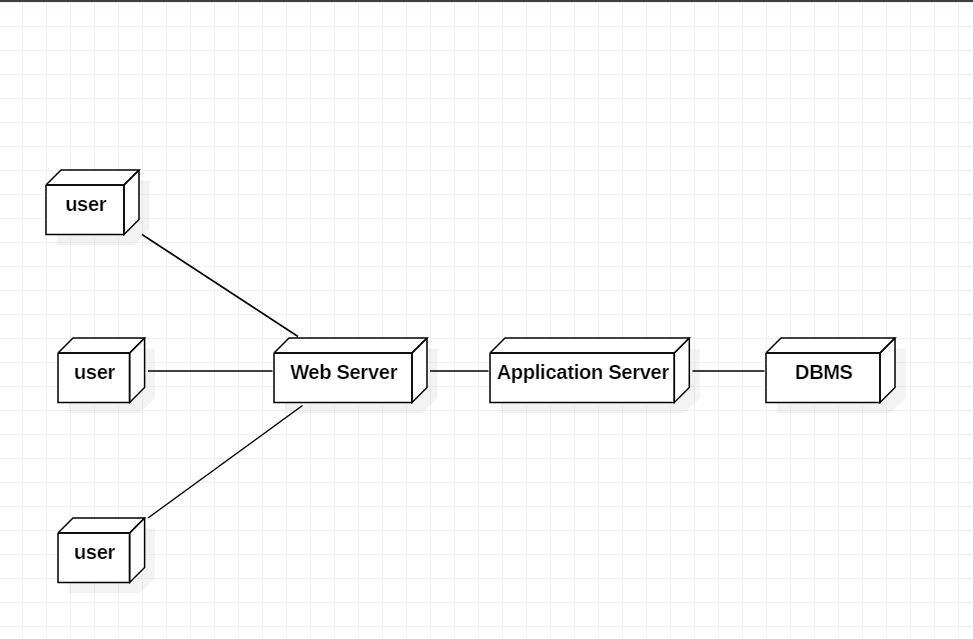
**4.2.5 Component diagram**

**DESCRIPTION**

The above diagram represents the component diagram of GST invoive application.

### 4.2.6 Deployment diagram

A deployment diagram is a UML diagram type that shows the execution architecture of a system, including nodes such as hardware or software execution environments, and the middleware connecting them. Deployment diagramsare typically used to visualize the physical hardware and software of a system.



**Fig : 4.2.6 Deployment diagram**

#### DESCRIPTION

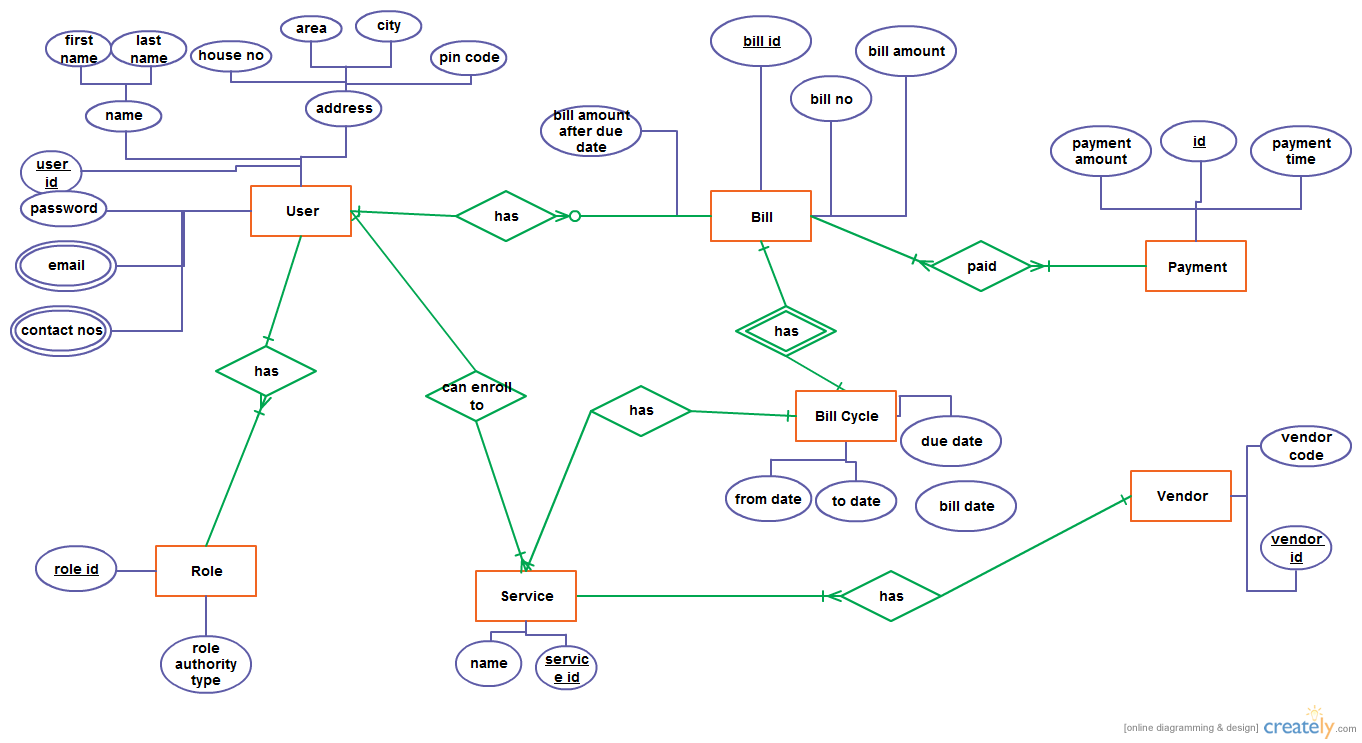
The above diagram represents the deployment diagram of GST invoice application.

**4.2.7 ER DIAGRAM**

An entity relationship diagram (ERD) shows the relationships of entity sets stored in a database. An entity in this context is an object, a component of data. An entity set is a collection of similar entities. These entities can have attributes that define its properties.

By defining the entities, their attributes, and showing the relationships between them, an ER diagram illustrates the logical structure of databases.

ER diagrams are used to sketch out the design of a database



**Description:**

This diagram shows the relationship between different entities and their attributes

**CHAPTER 5**

**IMPLEMENTION**

This specifies how the system application is implemented the usage of the modules specifications ,interactions involved in the system for the access purpose

## 5.1 MODULE ANALUSIS

## There are three modules in our GST invoice downloader to provide the access to the system and to make the functionalities work accordingly

## Module 1: Customer access

## The Module 1 is a lower level module that indicates the customer services to whom we are providing. There are B2B and B2C which indicates two different types of customers. One is the business oriented firm that contracts between the business to business and the second one is Business to Customer where and individual person or indicates a small shops.

**Module 2: Gst Ecosystem**

This module provides the other information related on which the Gstt server services are provided to the user this indicates both the services and the platforms throw which it is communication to the user.

* Web portal
* Mobile App
* Accounting
* ERP
* Custom options

**Module 3: GST server**

This is the main modules where all our services being offered like Adding client interaction to the client invoice maker and all others takes place.Here the whole web application is hosted on open stack increasing the availability to the user.

## 5.2 CLOUD COMPUTING

Computing as a service over the internet. Cloud computing, often referred to as simply “the cloud,” is the delivery of on-demand computing resources everything from applications to data centers over the internet on a pay-for-use basis.

**Features**

Elastic resources — Scale up or down quickly and easily to meet demand.

Metered service so you only pay for what you use.

Self service — All the IT resources you need with self-service access.

## Services

## The cloud computing model offers mainly three services:

## Software as a service (SaaS)

Cloud-based applications — or software as a service — run on distant computers “in the cloud” that are owned and operated by others and that connect to users’ computers via the internet and, usually, a web browser.

### Following are the benefits of saas

You can sign up and rapidly start using innovative business apps

Apps and data are accessible from any connected computer

No data is lost if your computer breaks, as data is in the cloud

The service is able to dynamically scale to usage needs

#### Platform as a service (PaaS)

Platform as a service provides a cloud-based environment with everything required to support the complete lifecycle of building and delivering web-based (cloud) applications — without the cost and complexity of buying and managing the underlying hardware, software, provisioning, and hosting.Following are the benefits of paas

Develop applications and get to market faster

Deploy new web applications to the cloud in minutes

Reduce complexity with middleware as a service

#### Infrastructure as a service (IaaS)

Infrastructure as a service provides companies with computing resources including servers, networking, storage, and data center space on a pay-per-use basis. Following are the benefits of iaas

No need to invest in your own hardware

Infrastructure scales on demand to support dynamic workloads

Flexible, innovative services available on demand

## 5.2 SAMPLE SOURCE CODING

### Creating a Login page

<!DOCTYPE html>

<html lang="en">

<head>

<title>GST Invoice</title>

<?php include('includes/head.php'); ?>

</head>

<body>

<h1>username: admin<br>password: xyz123</h1><br>

<div class="container">

<h1 class="centerheading"><img src="images/logo.png" /></h1>

<div class="loginbox">

<p style="margin:-10px -10px 10px -10px; padding:3px; background:#666; color:#fff; borderradius:5px 5px 0px 0px;">Login</p>

<form name="loginform" method="post" action="login.php">

<table border="0" cellpadding="5" cellspacing="0"> <tr><td>Username</td><td><input type="text"

name="username" id="username" class="required" label="Username" /></td></tr>

<tr><td>Password</td><td><input type="password"

name="password" id="password" class="required" label="Password" /></td></tr>

<tr><td>&nbsp;</td><td style="text-align:right">

<input type="hidden" name="redir"

value="<?php echo !empty($\_REQUEST['redir']) ?

urlencode( $\_REQUEST['redir'] ) : ""; ?>">

<input type="hidden" name="action" value="login"> <input type="button" name="loginsubmit" id="loginsubmit" value="Login" /></td></tr>

</table>

</form>

</div>

</div>

<script type="text/javascript"> var doLogin = function() { if(!APP.validateForm($('form[name=loginform]'))) {

return false; }

$.ajax({

url: "ajax.php", data:$('form[name=loginform]').serialize(), method:'post', success: function(resp) { console.log(resp); if(resp && resp.success && resp.success===true) { APP.redirecting();

if(resp.data && resp.data.redir &&

resp.data.redir!=null && resp.data.redir!='')

window.location =decodeURIComponent(resp.data.redir);

else window.location = 'index.php';

} else {

APP.showError("Login failed." +

(resp.message && resp.message ? '<br>' +

resp.message : ''));

}

}

});

};

$(document).ready(function(){

$('#loginsubmit').click(function(e) { doLogin();

});

$('#username, #password').keyup(function(e) { if(e.which===13) { doLogin();

}

});

$('#username').focus();

});

</script>

<?php include('includes/uielements.php'); ?>

</body> </html>

### Code for MYSQL

-- phpMyAdmin SQL Dump

-- version 4.7.7

-- https://www.phpmyadmin.net/

--

-- Host: localhost:3306

-- Generation Time: Mar 17, 2019 at 05:22 AM

-- Server version: 5.6.41-84.1-log

-- PHP Version: 5.6.30

SET SQL\_MODE = "NO\_AUTO\_VALUE\_ON\_ZERO";

SET AUTOCOMMIT = 0;

START TRANSACTION;

SET time\_zone = "+00:00";

/\*!40101 SET

@OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET

@OLD\_CHARACTER\_SET\_RESULTS=@@CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET

@OLD\_COLLATION\_CONNECTION=@@COLLATION\_CONNECTION \*/;

/\*!40101 SET NAMES utf8mb4 \*/;

--

-- Database: `valid64s\_GST`

--

-- --------------------------------------------------------

--

-- Table structure for table `hsn`

--

CREATE TABLE `hsn` (

`hsn\_code\_id` int(11) NOT NULL,

`hsn\_code` varchar(20) COLLATE utf8\_unicode\_ci NOT NULL,

`hsn\_code\_name` text COLLATE utf8\_unicode\_ci NOT NULL,

`GST\_rate` double NOT NULL DEFAULT '0'

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci;

--

-- Dumping data for table `hsn`

--

INSERT INTO `hsn` (`hsn\_code\_id`, `hsn\_code`, `hsn\_code\_name`, `GST\_rate`)

VALUES

(1, '1011010', 'LIVE HORSES, ASSES, MULES AND HINNIES PURE-BRED

BREEDING ANIMALS HORSES', 28),

(2, '96083021', 'HIGH VALUE PENS', 18);

-- --------------------------------------------------------

--

-- Table structure for table `invoice`

--

CREATE TABLE `invoice` (

`invoice\_id` int(11) NOT NULL,

`invoice\_ref\_no` varchar(30) COLLATE utf8\_unicode\_ci DEFAULT NULL,

`update\_by` int(11) NOT NULL,

`update\_ts` timestamp NOT NULL DEFAULT CURRENT\_TIMESTAMP ON

UPDATE CURRENT\_TIMESTAMP,

`sub\_total` double NOT NULL,

`tax\_total` double NOT NULL DEFAULT '0',

`gross\_total` double NOT NULL,

`archive\_in` tinyint(4) NOT NULL DEFAULT '0'

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci;

-- --------------------------------------------------------

--

-- Table structure for table `invoice\_item`

--

CREATE TABLE `invoice\_item` (

`item\_id` mediumint(9) NOT NULL,

`item\_name` varchar(100) COLLATE utf8\_unicode\_ci NOT NULL,

`item\_unit\_price` double NOT NULL,

`item\_unit\_name` varchar(30) COLLATE utf8\_unicode\_ci NOT NULL,

`item\_qty` double NOT NULL,

`item\_tax\_rate` double NOT NULL,

`item\_price` double NOT NULL,

`item\_tax\_total` double NOT NULL DEFAULT '0',

`item\_total` double NOT NULL,

`invoice\_id` mediumint(9) NOT NULL,

`ref\_product\_id` mediumint(9) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci;

-- --------------------------------------------------------

--

-- Table structure for table `product`

--

CREATE TABLE `product` (

`prod\_id` mediumint(9) NOT NULL,

`prod\_name` varchar(100) COLLATE utf8\_unicode\_ci NOT NULL,

`prod\_details` text COLLATE utf8\_unicode\_ci,

`unit\_id` int(11) NOT NULL,

`unit\_price` double NOT NULL,

`hsn\_code` varchar(20) COLLATE utf8\_unicode\_ci DEFAULT NULL,

`tax\_rate` double NOT NULL DEFAULT '0',

`stock\_qty` double NOT NULL DEFAULT '0',

`active\_in` tinyint(4) NOT NULL DEFAULT '1',

`update\_by` int(11) NOT NULL,

`update\_ts` timestamp NOT NULL DEFAULT CURRENT\_TIMESTAMP,

`upc` varchar(40) COLLATE utf8\_unicode\_ci DEFAULT NULL,

`gtin` varchar(40) COLLATE utf8\_unicode\_ci DEFAULT NULL,

`mpn` varchar(40) COLLATE utf8\_unicode\_ci DEFAULT NULL,

`barcode` int(40) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci;

--

-- Dumping data for table `product`

--

INSERT INTO `product` (`prod\_id`, `prod\_name`, `prod\_details`, `unit\_id`,

`unit\_price`, `hsn\_code`, `tax\_rate`, `stock\_qty`, `active\_in`, `update\_by`, `update\_ts`,

`upc`, `gtin`, `mpn`, `barcode`) VALUES

(1, 'Test Product 1', 'Test details 1', 1, 10, '', 10, 100, 1, 1, '2017-08-29 19:31:01', '', '', '', 0),

(2, 'Test Product 2', 'Test details 2', 2, 11, '96083021', 0, 100, 1, 1, '2017-08-29 19:31:01',

'', '', '', 0),

(3, 'Test Product 3', 'Test details 3', 1, 12, '', 12, 100, 1, 1, '2017-08-29 19:31:01', '', '', '', 0),

(4, 'Test Product 4', 'Test details 4', 2, 13, '96083021', 0, 100, 1, 1, '2017-08-29 19:31:01',

'', '', '', 0),

(5, 'Test Product 5', 'Test details 5', 1, 14, '', 14, 0, 1, 1, '2017-08-29 19:31:01', '', '', '', 0),

(6, 'Test Product 6', 'Test details 6', 1, 15, '1011010', 0, 100, 1, 1, '2017-08-29 19:31:01', '',

'', '', 0),

(7, 'Test Product 7', 'Test details 7', 1, 16, '', 16, 100, 1, 1, '2017-08-29 19:31:01', '', '', '',

0),

(8, 'Test Product 8', 'Test details 8', 1, 17, '', 17, 100, 1, 1, '2017-08-29 19:31:01', '', '', '',

0),

(9, 'Test Product 9', 'Test details 9', 1, 18, '', 18, 100, 1, 1, '2017-08-29 19:31:01', '', '', '', 0),

(10, 'Test Product 10', 'Test details 10', 1, 19, '', 19, 0, 1, 1, '2017-08-29 19:31:01', '', '', '', 0),

(11, 'Test Product 11', 'Test details 11', 1, 20, '', 20, 100, 1, 1, '2017-08-29 19:31:01', '', '',

'', 0),

(12, 'Test Product 12', 'Test details 12', 1, 21, '', 21, 100, 1, 1, '2017-08-29 19:31:01', '', '',

'', 0),

(13, 'Test Product 13', 'Test details 13', 2, 22, '1011010', 0, 100, 1, 1, '2017-08-29

19:31:01', '', '', '', 0),

(14, 'Test Product 14', 'Test details 14', 1, 23, '', 23, 100, 1, 1, '2017-08-29 19:31:01', '', '',

'', 0),

(15, 'Test Product 15', 'Test details 15', 1, 24, '', 24, 100, 1, 1, '2017-08-29 19:31:01', '', '',

'', 0),

(16, 'Test Product 16', 'Test details 16', 1, 25, '', 25, 100, 1, 1, '2017-08-29 19:31:01', '', '',

'', 0),

(17, 'Test Product 17', 'Test details 17', 1, 26, '', 26, 100, 1, 1, '2017-08-29 19:31:01', '', '',

'', 0),

(18, 'Test Product 18', 'Test details 18', 1, 27, '1011010', 0, 100, 1, 1, '2017-08-29

19:31:01', '', '', '', 0),

(19, 'Test Product 19', 'Test details 19', 1, 28, '', 28, 0, 1, 1, '2017-08-29 19:31:01', '', '', '', 0),

(20, 'Test Product 20', 'Test details 20', 1, 29, '', 29, 100, 1, 1, '2017-08-29 19:31:01', '', '',

'', 0),

(21, 'Test Product 21', 'Test details 21', 1, 30, '', 30, 100, 1, 1, '2017-08-29 19:31:01', '', '',

'', 0),

(22, 'Test Product 22', 'Test details 22', 1, 31, '', 31, 100, 1, 1, '2017-08-29 19:31:01', '', '',

'', 0),

(23, 'Test Product 23', 'Test details 23', 1, 32, '', 32, 100, 1, 1, '2017-08-29 19:31:01', '', '',

'', 0);

-- --------------------------------------------------------

--

-- Table structure for table `product\_unit`

--

CREATE TABLE `product\_unit` (

`unit\_id` int(11) NOT NULL,

`unit\_name` varchar(30) COLLATE utf8\_unicode\_ci NOT NULL,

`fraction\_allowed\_in` tinyint(4) NOT NULL DEFAULT '0'

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci;

--

-- Dumping data for table `product\_unit`

--

INSERT INTO `product\_unit` (`unit\_id`, `unit\_name`, `fraction\_allowed\_in`)

VALUES

(1, 'Piece', 0),

(2, 'KG', 1);

-- --------------------------------------------------------

--

-- Table structure for table `user`

--

CREATE TABLE `user` (

`user\_id` int(11) NOT NULL,

`full\_name` varchar(30) COLLATE utf8\_unicode\_ci NOT NULL,

`username` varchar(15) COLLATE utf8\_unicode\_ci NOT NULL,

`password` varchar(20) COLLATE utf8\_unicode\_ci NOT NULL,

`email\_id` varchar(50) COLLATE utf8\_unicode\_ci DEFAULT NULL,

`active\_in` tinyint(1) NOT NULL DEFAULT '1',

`access\_level` tinyint(4) NOT NULL DEFAULT '2'

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci;

--

-- Dumping data for table `user`

--

INSERT INTO `user` (`user\_id`, `full\_name`, `username`, `password`, `email\_id`,

`active\_in`, `access\_level`) VALUES

(1, 'Prithwiraj Bose', 'admin', 'xyz123', 'sribasu@gmail.com', 1, 1),

(2, 'Pom Bose', 'sampleuser1', 'xyz124', 'exampleid10000@gmail.com', 1, 2),

(3, 'Parijat Bose', 'sampleuser2', 'xyz125', 'exampleid10001@gmail.com', 1, 2),

(4, 'Ishita Bose', 'sampleuser3', 'xyz126', 'exampleid10000@gmail.com', 1, 2),

(5, 'Tilottama Bose', 'sampleuser4', 'xyz127', 'exampleid10001@gmail.com', 1, 2),

(6, 'Partha Sarathi Bose', 'sampleuser5', 'xyz128', 'exampleid10000@gmail.com', 1, 2),

(7, 'Mukti Bose', 'sampleuser6', 'xyz129', 'exampleid10001@gmail.com', 1, 2),

(8, 'Prithwiraj Bose', 'sampleuser7', 'xyz130', 'exampleid10000@gmail.com', 1, 2),

(9, 'Prithwiraj Bose', 'sampleuser8', 'xyz131', 'exampleid10001@gmail.com', 1, 2),

(10, 'Prithwiraj Bose', 'sampleuser9', 'xyz132', 'exampleid10000@gmail.com', 1, 2),

(11, 'Prithwiraj Bose', 'admin23', 'xyz123', 'sribasu@gmail.com', 0, 1),

(12, 'Pom Bose', 'hostuser1', 'xyz124', 'exampleid10000@gmail.com', 1, 2),

(13, 'Parijat Bose', 'hostuser2', 'xyz125', 'exampleid10001@gmail.com', 1, 2),

(14, 'Ishita Bose', 'hostuser3', 'xyz126', 'exampleid10000@gmail.com', 1, 2),

(15, 'Tilottama Bose', 'hostuser4', 'xyz127', 'exampleid10001@gmail.com', 1, 2),

(16, 'Partha Sarathi Bose', 'hostuser5', 'xyz128', 'exampleid10000@gmail.com', 1, 2),

(17, 'Mukti Bose', 'hostuser6', 'xyz129', 'exampleid10001@gmail.com', 1, 2),

(18, 'Prithwiraj Bose', 'hostuser7', 'xyz130', 'exampleid10000@gmail.com', 1, 2),

(19, 'Prithwiraj Bose', 'hostuser8', 'xyz131', 'exampleid10001@gmail.com', 1, 2);

--

-- Indexes for dumped tables

--

--

-- Indexes for table `hsn`

--

ALTER TABLE `hsn`

ADD PRIMARY KEY (`hsn\_code\_id`),

ADD UNIQUE KEY `hsn\_code` (`hsn\_code`);

--

-- Indexes for table `invoice`

--

ALTER TABLE `invoice`

ADD PRIMARY KEY (`invoice\_id`);

--

-- Indexes for table `invoice\_item`

--

ALTER TABLE `invoice\_item`

ADD PRIMARY KEY (`item\_id`),

ADD KEY `invoice\_items\_invoice\_id` (`invoice\_id`);

--

-- Indexes for table `product`

--

ALTER TABLE `product`

ADD PRIMARY KEY (`prod\_id`),

ADD KEY `prod\_name` (`prod\_name`,`upc`,`gtin`,`mpn`,`barcode`),

ADD KEY `unit\_id` (`unit\_id`),

ADD KEY `hsn\_code` (`hsn\_code`);

--

-- Indexes for table `product\_unit`

--

ALTER TABLE `product\_unit`

ADD PRIMARY KEY (`unit\_id`);

--

-- Indexes for table `user`

--

ALTER TABLE `user`

ADD PRIMARY KEY (`user\_id`),

ADD UNIQUE KEY `username` (`username`);

--

-- AUTO\_INCREMENT for dumped tables

--

--

-- AUTO\_INCREMENT for table `hsn`

--

ALTER TABLE `hsn`

MODIFY `hsn\_code\_id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=3;

--

-- AUTO\_INCREMENT for table `invoice`

--

ALTER TABLE `invoice`

MODIFY `invoice\_id` int(11) NOT NULL AUTO\_INCREMENT;

--

-- AUTO\_INCREMENT for table `invoice\_item`

--

ALTER TABLE `invoice\_item`

MODIFY `item\_id` mediumint(9) NOT NULL AUTO\_INCREMENT;

--

-- AUTO\_INCREMENT for table `product`

--

ALTER TABLE `product`

MODIFY `prod\_id` mediumint(9) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=24;

--

-- AUTO\_INCREMENT for table `product\_unit`

--

ALTER TABLE `product\_unit`

MODIFY `unit\_id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=3;

--

-- AUTO\_INCREMENT for table `user`

--

ALTER TABLE `user`

MODIFY `user\_id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=21;

--

-- Constraints for dumped tables

--

--

-- Constraints for table `product`

--

ALTER TABLE `product`

ADD CONSTRAINT `product\_ibfk\_1` FOREIGN KEY (`unit\_id`)

REFERENCES `product\_unit` (`unit\_id`) ON UPDATE NO ACTION;

COMMIT;

/\*!40101 SET

CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET

CHARACTER\_SET\_RESULTS=@OLD\_CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET

COLLATION\_CONNECTION=@OLD\_COLLATION\_CONNECTION \*/;

### Desiging code for style

body {

font-family: 'Roboto', sans-serif !important; font-size: 1em; background: #efefef; margin: 0px;

padding: 0px; box-sizing: border-box; }

a, a:link,

a:visited { color: #0078ff; text-decoration: none; transition: .5s text-shadow;

} a:hover { text-decoration: none; color: #0060cb;

text-shadow: 1px 1px 1px #ddd;

}

input[type=text], input[type=password], input[type=email], input[type=button], input[type=submit], button, textarea, select { font-family: 'Roboto', sans-serif; font-size: 1em; line-height: 20px;

border: 1px solid #999; border-radius: 2px; padding: 2px 5px;

background: #fff;

} input[type=checkbox],

input[type=radio] { line-height: 20px; border: 1px solid #999; border-radius: 2px; height: 18px; width: 18px;

background: #fff;

}

input[type=submit], input[type=button],

button { border-radius: 3px; background: #fefefe url(images/button-bg.png) bottom repeat-x; color: #0055ff;

padding: 4px 20px; font-size: 1em; cursor: pointer; font-weight: normal; letter-spacing: 1px;

margin: 2px 4px;

}

input[type=submit].smallbtn, input[type=button].smallbtn,

button.smallbtn { border-radius: 3px; background: #fefefe url(images/button-bg.png) bottom repeat-x; color: #0055ff;

padding: 2px 10px; font-size: 1em; cursor: pointer; font-weight: normal; letter-spacing: 0px; line-height: 18px;

margin: 0px 4px;

}

input[type=submit]:hover, input[type=button]:hover,

button:hover {

color: #fff; font-weight: normal;

background: #0055ff url(images/button-hover.png) bottom repeat-x;

}

.container { width: 90%;

margin: 0px auto;

}

.main { width: 100%; margin: 0px auto; background: #fff; padding: 10px;

}

.content { width: 100%; padding: 15px 10px; margin: 0; min-height: 400px;

}

.footer { width: 100%; margin: 0px; padding: 20px 10px;

text-align: center;

background: #999;

color: #fff; font-size: 0.8em; min-height: 40px;

}

.loginbox { width: 290px; margin: 100px auto;

text-align: center; background: #fff; border-radius: 5px; border: 1px solid #ccc;

padding: 10px;

}

.loginbox input[type=text],

.loginbox input[type=password] {

width: 182px;

}

.centerheading { text-align: center;

margin: 30px;

}

.ajaxloader { border: none !important; background: transparent;

background-color: transparent !important;

}

.prompt .bd {

text-align: center;

line-height: 24px;

}

.ui-widget-overlay { opacity: .80 !important;

/\* Make sure to change both of these, as IE only sees the second one \*/

filter: Alpha(Opacity=80) !important;

background-color: rgb(50, 50, 50) !important;

/\* This will make it darker \*/

}

.error {

color: red; }

input.error, select.error,

textarea.error { color: red; border: 1px solid red; background: lightyellow;

} nav {

background: #333; padding: 2px 20px;

} nav ul {

margin: 0; padding: 0; } nav ul li {

display: inline-block;

list-style-type: none;

} nav>ul>li>a { color: #bfddff !important; display: block; line-height: 2em; padding: 2px 20px; margin: 0px; text-decoration: none; transition: .5s color;

text-decoration: none;

}

nav>ul>li>a:hover { background: url(images/menu-item-bg.png) bottom repeat-x; border-radius: 5px; color: #fff !important;

text-decoration: none;

}

.logo { margin: 0px 5px 5px 5px;

padding: 0px;

float: left;

width: 50%;

}

.logo img {

border: none; width: 250px;

height: auto;

}

.toolbox {

float: right;

width: 30%;

text-align: right;

margin: 5px 10px; padding: 0;

}

.clear { clear: both; padding: 0; margin: 0; line-height: 1px; font-size: 1px; height: 1px;

}

.threecol {

float: left;

width: 32%; padding: 5px; margin: 6px;

}

.twocol {

float: left;

width: 47%; padding: 5px; margin: 6px; }

.onecol {

float: left;

width: 96%; padding: 5px; margin: 6px;

}

.widgetbox { border: 1px solid #ddd; border-radius: 5px; font-size: 0.8em !important;

}

.sectionheading { font-size: 1.3em; font-weight: normal; color: #0060ab;

margin: 2px 2px 10px 2px; padding: 5px; line-height: 1.4em; font-weight: bold; background: #eee;

}

.datagrid { table-layout: fixed !important; font-size: 0.9em !important;

}

.datagrid td { word-wrap: break-word !important;

cursor: pointer; padding: 5px !important;

}

.DataTables\_sort\_wrapper { vertical-align: bottom !important;

}

.fullwidth { width: 98%; }

input.fullwidth, button.fullwidth,

textarea.fullwidth {

width: 99%; }

select.fullwidth {

width: 100%;

}

.gridCheck { height: 14px !important; width: 14px !important; line-height: 16px !important;

}

.ui-dialog { font-size: 0.9em;

}

.CodeMirror { min-height: 100px; height: 100px;

}

.requiredStar { color: red; font-weight: bold; font-size: 1.2em; line-height: 1em; margin-left: 5px; font-family: monospace;

}

## 5.3 SCREENSHOTS

### 5.3.1 Login page

Graphical user interface, application

Description automatically generated

**Fig: 5.3.1 Login page**

#### DESCRIPTION

Login page:The login page can bev used by both admin and retailer to view their

details.

### 5.3.2 Admin home page

A computer screen capture

Description automatically generated with medium confidence

**Fig:5.3.2 Admin home page**

**Description**

Admin home page:The admin is the wholesaler who has the product list and the users list.

### 5.3.3 Product page

Graphical user interface, text, application

Description automatically generated

**Fig: 5.3.3 Product page**

#### Description

Product page: In this page where admin can check the current availabilty of product stock.

### 5.3.4 Adding products

Graphical user interface, text, table, email

Description automatically generated

**Fig: 5.3.4 Adding products**

#### Description

Adding products:The new products with their product details and stock details can be showen.

### 5.3.5 Deleting product

Graphical user interface

Description automatically generated

**Fig: 5.3.5 Deleting product**

**Description**

Deleting products:The products which are not in the list can be deleted.

### 5.3.6 Admin users page

Graphical user interface, application, email

Description automatically generated

**Fig: 5.3.6 Admin Users page**

#### Description

Admin Users page:This page can be accessed by only admin to check whether all the users in active or not.

**5.3.7 Adding user:**

Graphical user interface, application

Description automatically generated

**Fig: 5.3.7 Adding user**

**Description**

Adding user:A new can be added each time as a customer get added.

### 5.3.8 Deleting user

Graphical user interface, text, application

Description automatically generated

**Fig: 5.3.8 Deleting user**

#### Description

Deleting user:when the user goes to an inactive state then the details of the user can be deleted.

### 5.3.9 Invocie page

A picture containing chart

Description automatically generated

**Fig: 5.3.9 Invoice page**

#### Description

Invoice page:In this page where user and admin can check previous invoice and can also download the invoices.

### 5.3.10 About GST page

Graphical user interface, text

Description automatically generated

**Fig: 5.3.10 About GST page**

#### Description

About GST page: In this page where the users and admin can go through this and can learn about GST.

### 5.3.11 Contact page

Graphical user interface, application

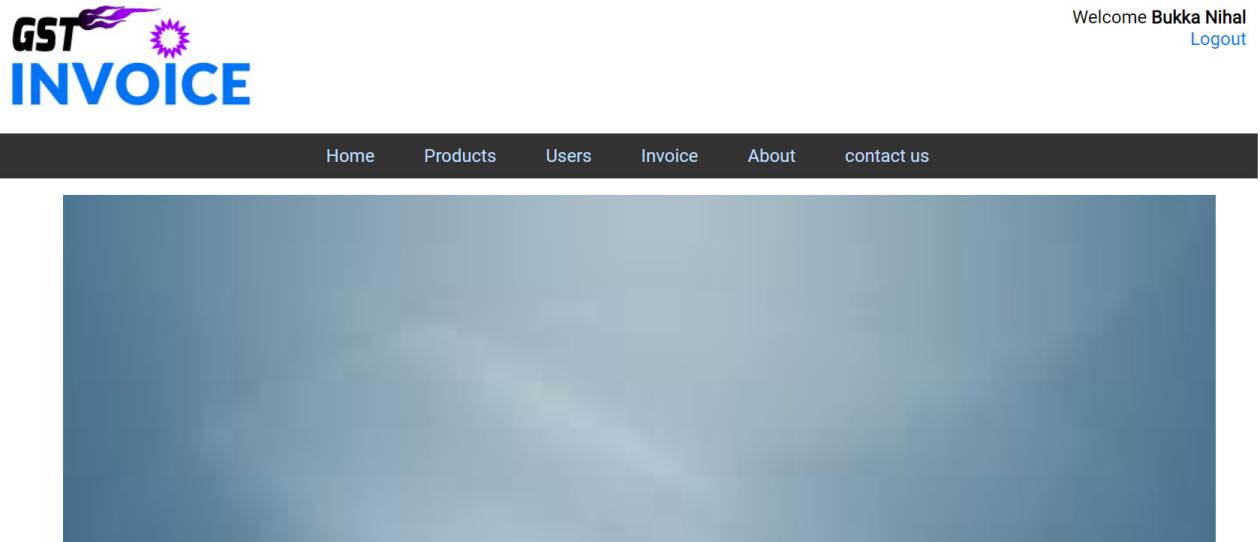
Description automatically generated

**Fig: 5.3.11 Contact page**

**Description**

Contact page: If users having any queires they can contact admin using this page.

### 5.3.12 User home page



**Fig: 5.3.12 User home page**

#### Description

User home page:The user is the retailer who will have the list of products.

**CHAPTER 6**

**SOFTWARE TESTING**

Software Testing is evaluation of the software against requirements gathered from users and system specifications. Testing is conducted at the phase level in software development life cycle or at module level in program code. Software testing comprises of Validation and Verification.

#### 6.1 SOFTWARE VALIDATION

Validation is process of examining whether or not the software satisfies the user requirements. It is carried out at the end of the SDLC. If the software matches requirements for which it was made, it is validated.

Validation ensures the product under development is as per the user requirements.

Validation answers the question – "Are we developing the product which attempts all that user needs from this software?".

Validation emphasizes on user requirements.

### 6.2 SOFTWARE VERIFICATION

Verification is the process of confirming if the software is meeting the business requirements, and is developed adhering to the proper specifications and methodologies.

Verification ensures the product being developed is according to design specifications.

Verification answers the question– "Are we developing this product by firmly following all design specifications?”

Verifications concentrate on the design and system specifications.

#### 6.3 TARGET OF THE TEST AREA

**Errors** -These are actual coding mistakes made by developers. In addition, there is a difference in output of software and desired output, considered as an error.

**Fault** - When error exists fault occurs. A fault, also known as a bug, is a result of an error which can cause system to fail.

**Failure** - failure is said to be the inability of the system to perform the desired task. Failure occurs when fault exists in the system.

#### 6.4 BLACK-BOX TESTING

It is carried out to test functionality of the program. It is also called ‘Behavioral’ testing. The tester in this case, has a set of input values and respective desired results. On providing input, if the output matches with the desired results, the program is tested ‘ok’, and problematic otherwise.

Calendar

Description automatically generated

**Fig 6.4:Black box testing**

In this testing method, the design and structure of the code are not known to the tester, and testing engineers and end users conduct this test on the software.

##### **6.4.1 BLACK-BOX TESTING TECHNIQUES**

**Equivalence class** - T he input is divided into similar classes. If one element of a class passes the test, it is assumed that all the class is passed.

**Boundary values** - T he input is divided into higher and lower end values. If these values pass the test, it is assumed that all values in between may pass too.

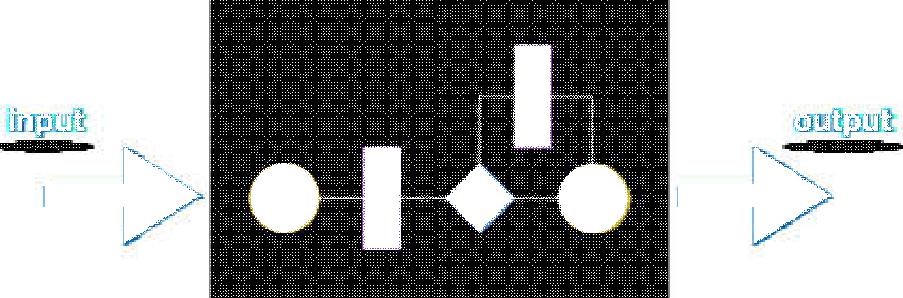
**Cause-effect graphing** - In both previous methods, only one input value at a time is tested. Cause (input) – Effect (output) is a testing technique where combinations of input values are tested in a systematic way.

**Pair-wise Testing** - The behavior of software depends on multiple parameters. In pair wise testing,the multiple parametrs are tested pair-wise for their different values.

**State-based testing** - The system changes state on provision of input.These systems are tested based on their states and input.

##### **6.5 WHITE-BOX TESTING**

It is conducted to test program and its implementation, in order to improve code efficiency or structure. It is also known as ‘Structural’ testing.



**Fig 6.5:White box testing**

In this testing method, the de sign and structure of the code are known to the tester. Programmers of the code conduct this test on the code

###### The following are some White-box testing techniques

**Control-flow testing** - The purpose of the control-flow testing to set up a test case which covers all statements and branch conditions. The branch conditions are tested for both being true and false, so that all statements can be covered.

**Data-flow testing** - This testing technique emphasis to cover all the data variables included in the program. It tests where the variables were declared and defined and where they were used or changed.

##### **6.6 TESTING LEVELS**

Testing itself may be defined at various levels of SDLC. The testing process runs parallel to software development. Before jumping on the next stage, a stage is tested, validated and verified. Testing separately is done just to make sure that there are no hidden bugs or issues left in the software. Software is tested on various levels -

###### **6.6.1 UNIT TESTING**

###### While coding, the programmer performs some tests on that unit of program to know if it is error free. Testing is performed under white-box testing approach. Unit testing helps developers decide that individual units of the program are working as per requirement and are error free.

###### **6.6.2 INTEGRATION TESTING**

Even if the units of software are working fine individually, there is a need to find out if the units if integrated together would also work without errors. For example, argument passing and data updation etc.

**6.6.3 SYSTEM TESTING**

The software is compiled as product and then it is tested as a whole. This can be accomplished using one or more of the following tests:

**Functionality testing** - Tests all functionalities of the software against requirement.

**Performance testing** - This test proves how efficient the software is. It tests the effectiveness and average time taken by the software to do desired task. Performance testing is done by means of load testing and stress testing where the software is put under high user and data load under various environment conditions.

**Security & Portability** - These tests are done when the software is meant to work on various platforms and accessed by number of persons.

**6.6.4 ACCEPTANCE TESTING**

When the software is ready to hand over to the customer it has to go through last phase of testing where it is tested for user-interaction and response. This is important because even if the software matches all user requirements and if user does not like the way it appears or works, it may be rejected.

**Alpha testing** - The team of developer themselves perform alpha testing by using the system as if it is being used in work environment. They try to find out how user would react to some action in software and how the system should respond to inputs.

**Beta testing** - After the software is tested internally, it is handed over to the users to use it under their production environment only for testing purpose. This is not as yet the delivered product. Developers expect that users at this stage will bring minute problems, which were skipped to attend.

###### **6.6.5 REGRESSION TESTING**

Whenever a software product is updated with new code, feature or functionality, it is tested thoroughly to detect any negative impact of any added code

**6.7 TESTING REPORTS**

**Case 1:**

|  |  |  |
| --- | --- | --- |
| **Test case1:** Admin Login | **Priority (H, L):** High | |
| **Test Objective:** To update the data into the server and to view user details | | |
| **Test Objective:** To update the data into the server and to view user details | | |
| **Requirements verified:** Yes | | |
| **Test Environment:** PC | | |
| **Actions** | | **Expected** |
| The admin logs in | | Admin can access the details of the users and the database |
| Admin can access the details of the users and the database | | |
| **Problems/Issues: NIL** | | |

**Notes: Successfully executed**

**Case 2:**

|  |  |  |
| --- | --- | --- |
| **Test case1:** User Login | **Priority (H, L):** High | |
| **Test Objective:** To access the data into the server | | |
| **Test Description:** The user can login using credentials. Display user home page and can view available data | | |
| **Requirements verified:** Yes | | |
| **Test Environment:** PC | | |
| **Actions** | | **Expected** |
| The user logs in | | Display user home page and can view  available data |
| user can access the details of the invoice and the database | | |
| **Problems/Issues: NIL** | | |

**Notes: Successfully executed**

**CHAPTER 7**

**CONCLUSION AND FUTURE ENHANCEMENT**

**7.1 CONCLUSION**

The proposed system will be the system which is used to provide the Paas in which the news are dynamically accesssed by the users and the host managers need not manage the site every time.Distributed access to the GST will be provided on the cloud open stack for hosting purpose.

**7.2 FUTURE ENHANCEMENT**

This system will furture be refered for the security purpose the present system only consist of the invoice option and in future enhancement we will include additional calculated formats

**LIST OF SYMBOLS**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.NO** | **NOTATION NAMe** | **NOTATION** | **DESCRIPTION** |
| 1. | Class | *+ public*  *-private*  *# protected*  *Class Name*  *-attribute*  *-attribute*  *+operation*  *+operation*  *+operation* | Represents a collection of similar entities grouped together. |
| 2. | Association | Class B  Class A  Class A  Class B | Associations represents static relationships between classes. Roles represents the way the two classes see each other. |
| 3. | Actor |  | It represent the user. |
| 4. | Aggregation | Class B  Class B  Class A  Class A | It aggregates several classes into a single classes. |
| 6. | Communication |  | Communication between various use cases. |
| 7. | Usecase |  | Interaction between the system and external environment. |

**REFERENCES**

**JOURNALS**

* GST in India Concept and SWOT Analysis.
* A Comprehensive Analysis of Goods and Services Tax (GST) in India.
* GST BILLING SYSTEM.
* IMPACT OF GST ON WAREHOUSING INDUSTRY.
* Reduced Anthocyanins in Petioles codes for a GST anthocyanin transporter that is essential for the foliage and fruit coloration in strawberry.

**WEBSITES**

* SQL DataBase
* XAMP
* <https://services.gst.gov.in/services/login>