**Chapter 1**

**INTRODUCTION**

**1.1 Company Profile**

**1.2 Introduction to Project**

**1.1 Company Profile**

Tech-Nebulas established in 2014.It is software Development Company, working in various domains such as CRM, ERP, Web Development, Domain Name Registration , Providing Bulk SMS. Tech-Nebulas is founded to address the need of cost effective yet powerful and user friendly systems, the software systems that donate only help to grow the business but also establish a level of trust between our clients and their end customer.

Tech-Nebulas offer end to end IT consulting services using the best practice in the industry. Our solution include IT Training, Corporate Training, Software development service, IT project service. Tech-Nebulas works with software development and services companies to leverage resources and maximize value to their customers. Tech-Nebulas extensive experience and standard software development process helps to create innovative software solutions with high reliability.

Tech-Nebulas were established with a mission to be client oriented, technology, quality oriented, adaptive and innovation in its approach.

Company Name : Tech-Nebulas

Company Address : 202 ‘Pitakunj’, Above Janaseva Bank,NDA Road, Warje, Pune-41105

E-mail : info@technebulas.com

* 1. **Introduction To Project**

**Chain management system**:

Supply chain are "linked" together through physical flow and information flow. Physical flow involve the transformation ,movement, storage of goods and materials. They are the most visible piece of the supply chain. But just as important are information flow. Information flows allow the various supply chain partners to co-ordinate their long-term plans , and to control the day-to-day flow of goods and material up and down the supply chain. This system allow all authority to some main admin or authorized person who maintain the all data related to that system. Other associates are consider to do the work done remains on the system.

Chin management system, is the active management of chain activities to minimize customer value and achieve sustainable competitive advantage.

**Features:**

**Portal Admin:**

Portal admin is the main authorized person of system. Who create more than one admin and give them whole authority to operate the system.

**Admin:**

Chain management system process is mainly handled by admin. all processes like creating associate and giving each associate particular key. As per associate creates particular key is considered for that associate.

After creating admin he will becomes authorized person to handle all over process related to system like creating associate one or more than one. The level will always increments whenever admin creates an associate. As per the process the level increments step by step and associates will also incremented. Admin having main activity to add product and select particular value of amount to each product. admin can directly sell the product to the customer or he can give the product to associate. Amount of product is set by admin only. Paying commission and view the all details by only admin.

**Associate:**

When associate is created he will be handle the next procedure of system like selling the products to customers. as per selling any product, associate can get proper commission.

**There are following levels which will be incremented step by step:**

1. Business officer
2. Field manager
3. Area manager
4. Divisional manager
5. Regional manager
6. Silver achiever
7. Gold achiever
8. Platinum achiever
9. Diamond achiever

**Purpose:**

It is always difficult to maintain some tedious work manually which may lead to missing information, duplicate data and many such un-avoidable situations.

To avoid such conditions it has become a necessity to implement a powerful system. This system will help seller , customer in their day-day to activities. It reduces paper work, save time and provides complete management over the college.

**Information of used Technology:**

1. **Introduction to PHP**

PHP is a powerful server-side scripting language for creating dynamic and interactive websites.

PHP is the widely-used, free, and efficient alternative to competitors such as Microsoft's ASP. PHP is perfectly suited for Web development and can be embedded directly into the HTML code.

The PHP syntax is very similar to Perl and C. PHP is often used together with Apache (web server) on various operating systems. It also supports ISAPI and can be used with Microsoft's IIS on Windows.

**What is PHP?**

* PHP stands for **P**HP: **H**ypertext **P**reprocessor.
* PHP is a server-side scripting language, like ASP.
* PHP scripts are executed on the server.
* PHP supports many databases (MySQL, Informix, Oracle, Sybase, Solid, PostgreSQL, Generic ODBC, etc.),
* PHP is open source software.
* PHP is free to download and use.

**B. JavaScript:**

|  |  |
| --- | --- |
|  |  |

* JavaScript is the scripting language of the web.
* JavaScript used in million of the web pages to add functionality, improve design, create cookies, validate forms, detect browsers, and much more.

**What is JavaScript?**

* JavaScript was designed to add interactivity to HTML pages
* JavaScript is a scripting language
* A scripting language is a lightweight programming language
* JavaScript is usually embedded directly into HTML pages
* JavaScript is an interpreted language (means that scripts execute without preliminary compilation).

**C: MySQL:**

* SQL is stand for Structured Query Language.
* MySQL is a standard language for accessing and manipulating databases.

**What Can MySQL do?**

* MySQL can execute queries against a database.
* MySQL can retrieve data from a database.
* MySQL can insert records in a database.
* MySQL can update records in a database.
* MySQL can delete records from a database.
* MySQL can create new databases.
* MySQL can create new tables in a database.
* MySQL can create stored procedures in a database.
* MySQL can create views in a database.
* MySQL can set permissions on tables, procedures, and views.

**Chapter 2**

**SYSTEM REQUIREMENT ANALYSIS**

**2.1 Need and Motivation**

**2.2 System Requirement Analysis**

**2.3 Software Process and Development model**

**2.4 Hardware/ Software Specification**

**2.1 Need and Motivation**

* It provides a GUI environment, which is easy to use and understand.
* You can handle your business from anywhere in world.
* Tracking of all the information related to the system becomes easy and fast.
* Fast access of data.
* It becomes a Customized system.
* As it will be shared data through network so will be accessible to the entire department.
* Easy updation of data throughout the database.
* The system is Cost effective and gives high productivity (i.e. High Productivity with low cost).
* The system provides all required information in hand and which helps to its Increase labor productivity.
* Less Manual Work and Quick information retrieval increases Administration efficiency.
* It can also help in establishing own fleet benchmarks.
* It also helps in operating budget projection.
* It any mistakes done can be found easily.
* Graphical reports are generated, so the overall status of the organization can be traced easily which help in tracing the organization growth can also help in the optimization of available resources, for the progress of the organization.

**2.2 System Requirement Analysis**

Software Requirements Specification plays an important role in creating quality software solutions. Specification is basically a representation process. Requirements are represented in a manner that ultimately leads to successful software implementation.

The Software requirement specification is developed as a consequence of analysis. Review is essential to ensure that the developer and the customer have the same perception of the system. The introduction of the software requirements specification states the goals and objectives of the software, describing it in the context of the computer-based system. The information description provides a detailed description of the problem that the software must solve. Information content, flow, and structures are documented. Hardware, software, and interfaces are described for external system elements and internal software functions.

A description of each function required to solve the problem is presented in the functional description. A processing narrative is provided for each function, design constraints are stated and justified, performances characteristics are stared, and one or more diagrams are included to graphically represent the overall structure of the software as a consequence of external events and internally generated control characteristics.

A validation Criteria is probably the most important section of the software requirement Specification. How do we recognize a successful implementation? What classes of tests must be conducted to validate function, performance, and constraints? Specification of validation criteria acts as an implicit review of all other requirements.

Finally, the specification includes a Bibliography and Appendix. The bibliography contains references to all documents that relate to the software. These include other software engineering documentation, technical references; vendor literature and standards .The appendix contains information that supplements the specification. Tabular data, detailed description of algorithms, charts, graphs, and other material are presented as appendixes.

The preliminary Users Manual presents the software as black box. That is heavy emphasis is placed on user input and the resultant output. The manual can serve as a valuable tool for recovering problems the human/machine interface. Both the Software developer and the customer conduct a review of the software requirements Specification. Requirement may be specified in the variety of ways. However there are some guidelines worth following.

* Representation format and content should be relevant to the problem
* Information contained within the specification should be nested
* Diagrams and other notational forms should be restricted in number and consistent in use.
* Representations should be revisable.

**2.3 Software Process and Development Model**

**2.3.1 The Software Process:**

The software process takes the problem definition as input and produces the software product. The process of software development is divided into various phases and each phase a set of activities is carried out to arrive at end product.

**2.3.2 Software Development Life Cycle:**

Each phase in development process will have a defined input and a defined output. The marketing department of an organization will carry out initial dialogue with the prospective client and get a project. The development can categorize into the following phases.

* Software Engineering and Analysis.
* Software Requirement Analysis.
* Design.
* Coding.
* Testing.

System Engineer

Analysis

Design

Coding

Testing

Maintenance

Fig. Waterfall Model

**Software Engineering and Analysis:**

The software is part of a largest system, works begins by establishing requirements for all system elements and then allocating same subset of these requirements to software, this system view is essential when software must interface with other elements such as hardware, people and database. Software engineering and analysis go the requirements gathering at system levels with a small amount of top-level design and analysis.

**Software Requirements Analysis:**

The requirements of the client are obtained, analyzed, documented and validated. Requirement engineering process is problem definition and output is validated as per the software requirements specification(SRS).To understand the nature of program to be built, the software engineer must understand the information of the software, as well as the required function and performance.

**Design:**

Software design involves architecture of application, design of the algorithms and data structures. The design process translates requirements into representation of the software. The output of the phase is to design various modules.

**Coding:**

The design must be translated into machine-readable language. The code generation step performs this task. The design document prepared by the software architecture should be given to the programmer and programmer should be able to do the coding.

**Testing:**

Once code has been generated, Software testing begins. The testing process focuses on the logical internals of software, ensuring that all statements have been tested, i.e. conducting tests to uncover errors and ensure that define input will procedure actual result that agree with required result

.

**2.4 Hardware/ Software Specification**

**Client Side Hardware Requirement**

Processor : PENTIUM4.

RAM : 1 GB or more.

HARD DISK : 40 GB or more.

**Client Side Software Requirement**

Operating system : Windows Operating System.

Browser : internet explorer, Google chrome, Mozilla Firefox.

**Server Side Hardware Requirement**

Processor : PENTIUM4.

RAM : 1 GB or more.

HARD DISK : 40 GB or more.

**Server Side Software Requirement**

Front End : JavaScript, css3, html 5, PHP, CodeIgniter.

Back end : My sql.

Operating system : Windows Operating System.

Browser : Internet explorer, Google chrome, Mozilla Firefox.

Server : XAMPP Server

**Chapter 3**

**FEASIBILITY STUDY**

**3.1 Introduction**

**3.2 Technical Feasibility**

**3.3 Economical Feasibility**

**3.4 Operational Feasibility**

**3.5 Behavioural Feasibility**

**3.1 Introduction**

A Feasibility study is a process to check possibility of system development. It is a method and different requirements and availability of financial and technical resources.

A project feasibility study is an exercise that involves documenting each of the potential solutions to a particular business problem or opportunity.

After initial investigation of the system some study is carried out to decide whether the proposed computer can solve candidate system based system by eliminating all the drawbacks and limitations of existing system. This study is known as ‘Feasibility study’. Performance of the proposed system is evaluated in terms of the tangible benefits (which Are physically miserable) and intangible benefits (which are in terms of the quality, user Satisfaction).

**3.2 Technical Feasibility:**

The proposed system can be implemented with the existing technology the company already has a set up for it. The user of the system, who needs very little training .The candidate system, is tested for Technical Feasibility.

**Is the system portable? Yes**

**Does necessary technology exist to do the required task? Yes**

**Is the system User friendly? Yes**

**Does it provide a help file? Yes**

**3.3 Economical Feasibility:**

Cost -Benefit Analysis is used for this economical feasibility. The proposed System is proved to be economical feasible.

* Cost of operation of existing system: The present system is time consuming.
* Cost of operation of proposed system: No additional cost will be incurred for buying new system.
* Man Power Cost: The Organization will not require any additional manpower, because user of the system is already computer literate and feasible environment of computerization in their organization.

**3.4 Operational Feasibility:**

Proposed projects are beneficial only if they can be turned out into information system. That will meet the organization’s operating requirements. Operational feasibility aspects of the project are to be taken as an important part of the project implementation. Some of the important issues raised are to test the operational feasibility of a project includes the following: -

Is there sufficient support for the management from the users?

* Will the system be used and work properly if it is being developed and implemented?
* Will there be any resistance from the user that will undermine the possible application benefits?

This system is targeted to be in accordance with the above-mentioned issues. Beforehand, the management issues and user requirements have been taken into consideration. So there is no question of resistance from the users that can undermine the possible application benefits.The well-planned design would ensure the optimal utilization of the computer resources and would help in the improvement of performance status.

**3.5 Behavioral Feasibility:**

The current working atmosphere of the organization is not disturbed by introduction of the proposed system. The time saver and immediate results produced by just one action of the finger greatly reduced the manual effort. As a whole, this system will help the organization to have a higher level of performance. Hence, the system is feasible

**Chapter 4**

**System Design**

**4.1 Project Architecture**

**4.2 Use Cases**

**4.3 Entity Relationship Diagram**

**4.4 Sequence Diagram**

**4.5 Activity Diagram**

**4.6 Deployment Diagram**

**4.7 Data Dictionary**

**4.1 Project Architecture**

**4.2 Use Cases**

**Use case for portal admin**

**Use case for admin and associate**

**4.3 Entity Relationship Diagram**

**4.4 Sequence Diagram**

**Sequence Diagram for Portal Admin:**

**Sequence Diagram for Admin:**

**.**

**Sequence Diagram for Associate:**

**4.5 Activity Diagram**

**Activity diagram for login:**

**Activity diagram for add product:**

**Activity diagram for key generate :**

**Activity diagram for sale product:**

**Activity diagram for payment:**

**4.6 Deployment Diagram**

**4.7 Data Dictionary**

|  |
| --- |
|  |
| |  | | --- | |  | |

**Chapter 5**

**TESTING**

**5.1 Testing Procedure**

**5.2 Unit Testing**

**5.3 Integral Testing**

**5.4 Validation Testing**

**5.5 User Acceptance Testing**

**5.1 Introduction**

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and code generation. It is a process of executing a program with a primary objective of finding errors. Testing gives the guarantee that the software does not fail and runs according to its specification and in the way the end user expects. This can be done by various software testing techniques which provide a systematic guidance for designing tests that exercise the internal logic of software components and exercise the input and output domains of the program to uncover errors in program function, behavior and performance.

The following software testing techniques were used in order to uncover errors in the system:

* + Unit testing.
  + Integral Testing
  + Validation Testing
  + User Acceptance Testing

**5.2 Unit Testing:**

Unit testing is normally considered as an adjunct to the coding step. It focuses verification effort on the smallest unit of software design. It has been conducted within the programming area. The individual unit have been tested in isolation and written to meet the specification of requirements, including any requirement to interact with other units in the system.

Unit testing is the test for the small units of the code, e.g. programs, modules or procedures, in order to ensure that they perform their intended functions. All possible paths through the control structure were exercised to ensure that all statements in a program were executed at least once .Unit testing was also done to test the data flow across a module interface.

The following errors were uncovered during unit testing.

* Comparison of different data types.
* Incorrect logical operators or precedence.
* Incorrect comparison of variables.
* Improper or nonexistent loop termination.
* Improperly modified loop variables.

**5.3 Integral testing**:

Testing of integrated modules to verify combined functionality after integration. Modules are typically code modules, individual applications, client and server applications on a network, etc. This type of testing is especially relevant to client/server and distributed systems.

**5.4 Validation testing:**

Validation testing occurs when we have to declare some validation regarding our input screens. For example, suppose we have an input screen in which we want to enter date between 1 to 31, so in case if the user enters date above 31 I. e. 32, then it will show an error. This is called validation testing.

**5.5 User acceptance testing:**

User acceptance of the system is key factor for the success of any system. The system under consideration is tested for the user acceptance by constantly keeping in touch with prospective system and user at the time of developing and making changes whenever required. This is done in regarding to the following points.

* Input Screen design
* Output Screen design
* Menu driven system

**Chapter 6**

**CONCLUDING REMARKS**

**6.1 Key Features**

**6.2 Possible Enhancements**

**6.1**  **Key Features**

* Easily sale the product to the customer.
* Easy to handle to each user.
* Easy to maintain the commission and levels of associates .
* The system personalized all the modules on dashboard which helps to understand each module.
* Easy to understand the workflow of system.
* The system is very customizable, it provides particular details in right way.
* Working on each module is efficient and convenient.
* The system is user friendly.

**6.2 Possible Enhancements**

Current system is designed in short amount of time so all functionality are not included in the system. More functionality are not included in the system in feature to help user of the system.

* System can be used for any number of company.
* Used by any number of employees.
* More attractive look of the forms and system.
* Modules are dependent to each other.
* No interaction of customer.
* Without key generate admin don’t sale the product.

**Chapter 7**

**REFERENCES**

**Books**

* + Head First HTML5 by Eric Freeman and Elisabeth Robson
  + Introducing HTML5 by Bruce Lawson and  Remy Sharp
  + Sams Teach Your Self JavaScript by Phil Ballard and  Michael Moncur
  + Complete Reference JavaScript by Thomas A. Powell  and Fritz Schneider
  + The Complete Reference PHP by Steven Holzner.

**Reference Website:**

* + www.w3schools.com
  + www.htmlfreecodes.com
  + www.html5rocks.com
  + www.jquery.com

**8.1 Appendix A – User Manual (if any)**

**Chapter 8**

**APPENDIX**

**8.2 Appendix B – Screen Layout**

**8.3 Appendix C – List of Tables**

**Appendix A: User Manual**

**Purpose**

For any system to be successful it is important that the intended user find the system easy to operate. The purpose of the user manual is to make user acquainted with the system and help user to understand the system and operate it conveniently. The manual contain several screenshots that describes how to use the entire system.

There are various basic requirements which are necessary to use the proposed system.

**Requirements:**

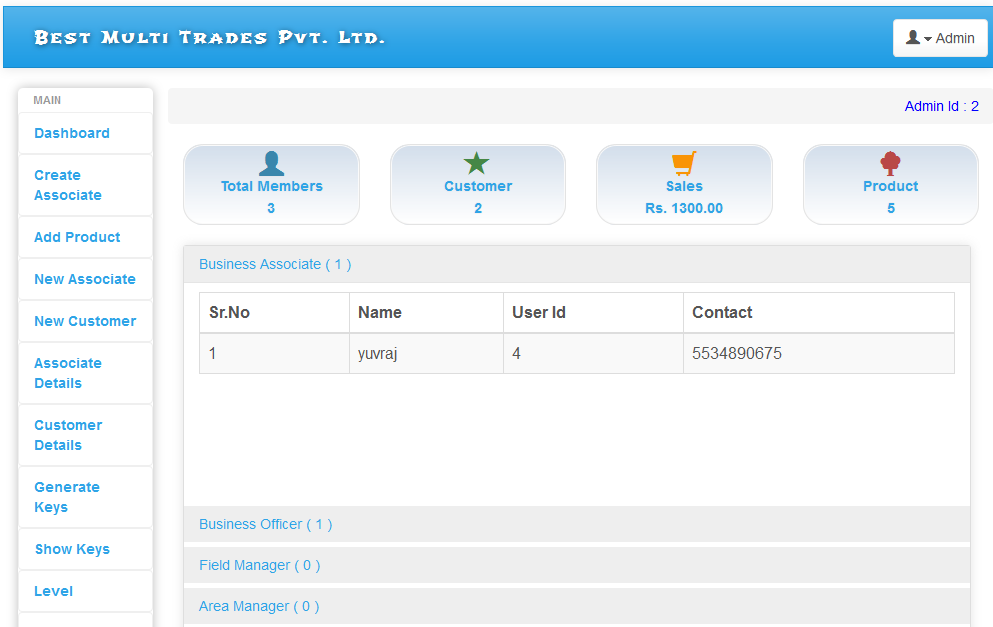
* Computer system
* Internet connection
* Client must have some knowledge of computer

**Process to Open Web Application:**

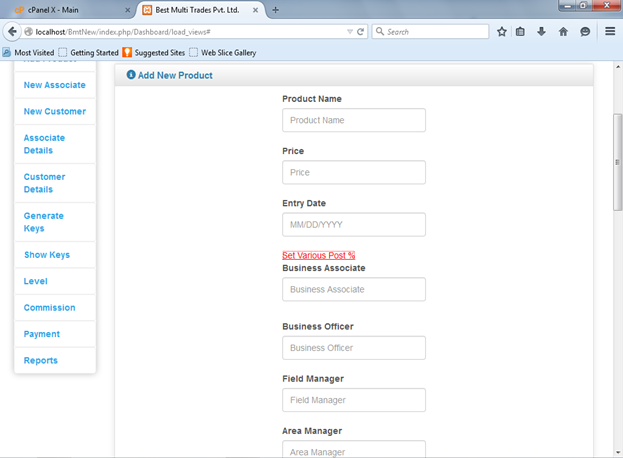
* Open Web browser
* Enter URL
* "localhost/BmtNew/index.php"
* You will get admin login screen

**Appendix B: Screen Layout**

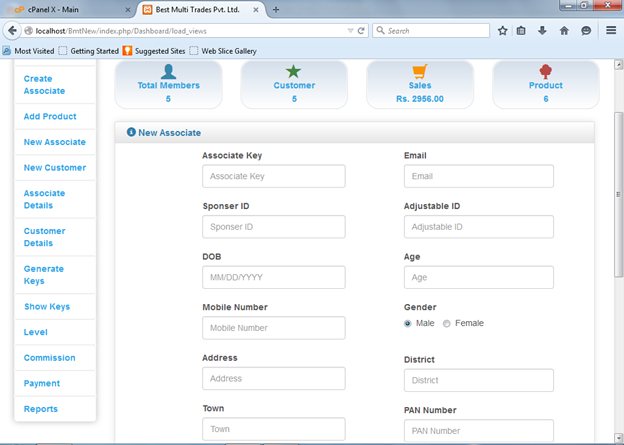
**Admin Dashboard**

****

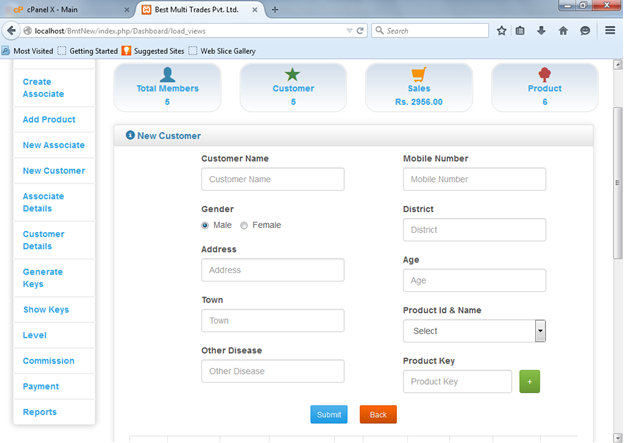
**Add Product**

****

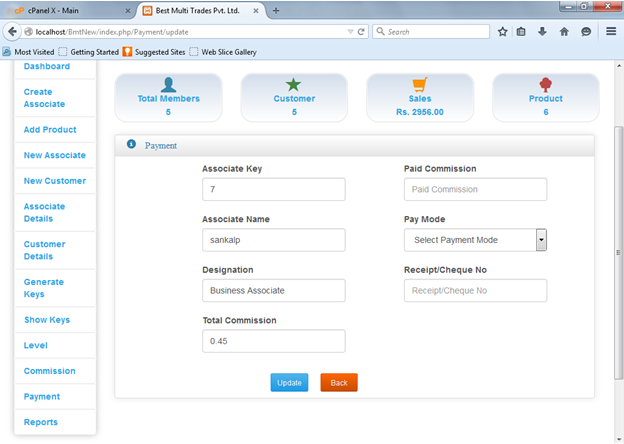
**New Associate**

****

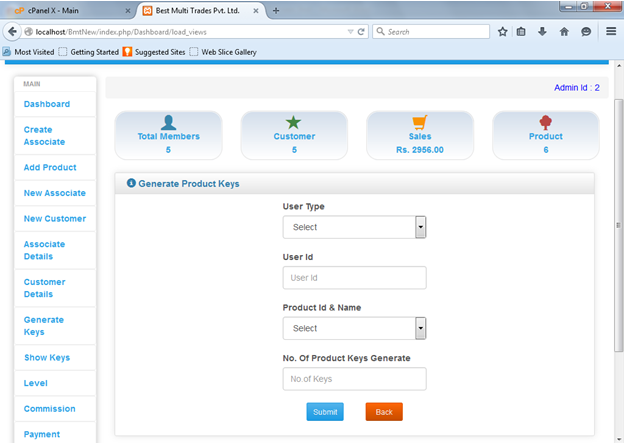
**New Customer**

****

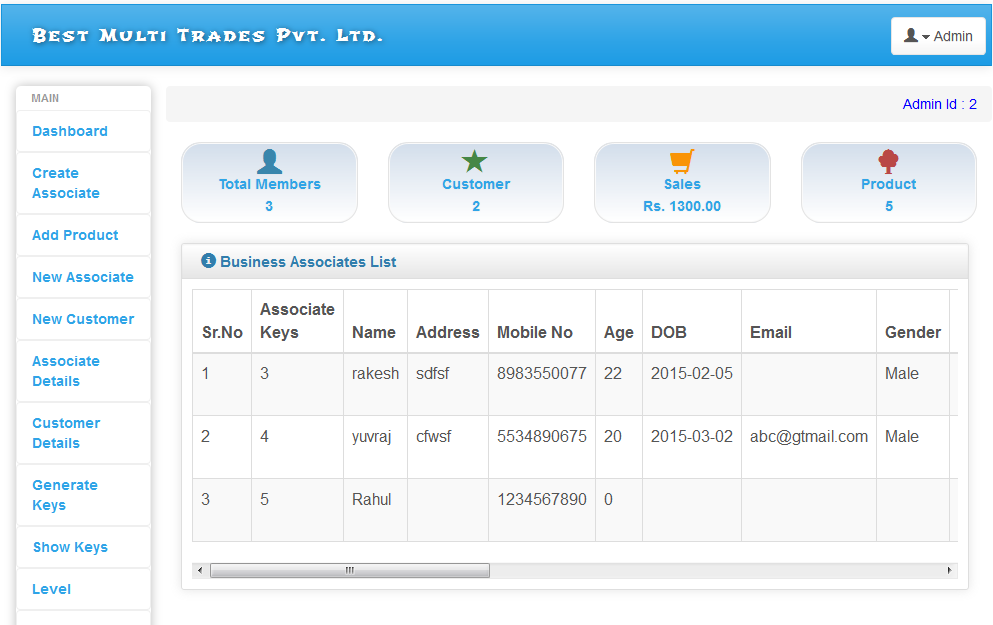
**Associate Payment**

****

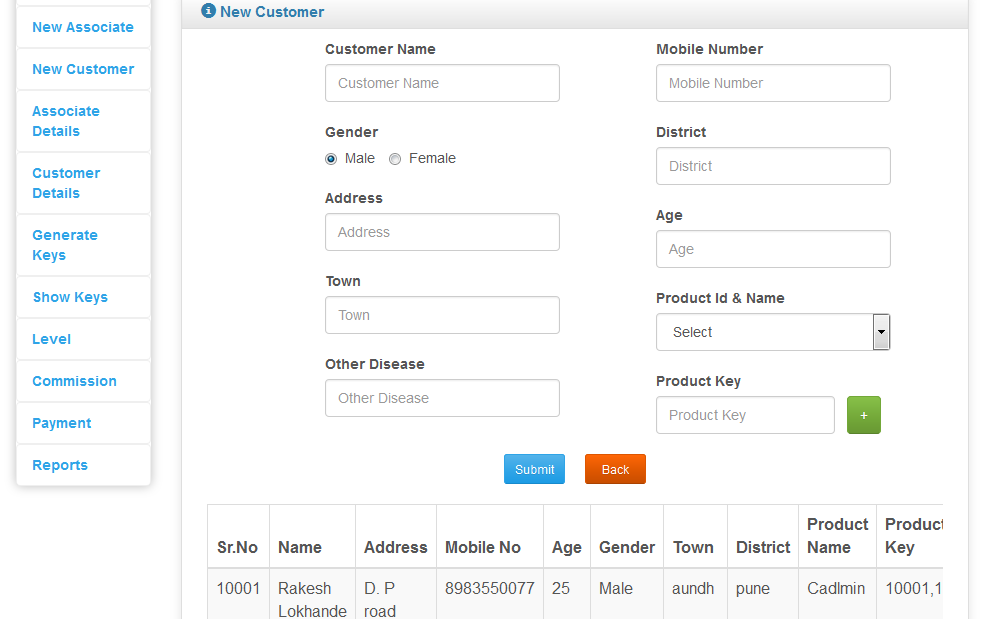
**Generate Product Keys**

****

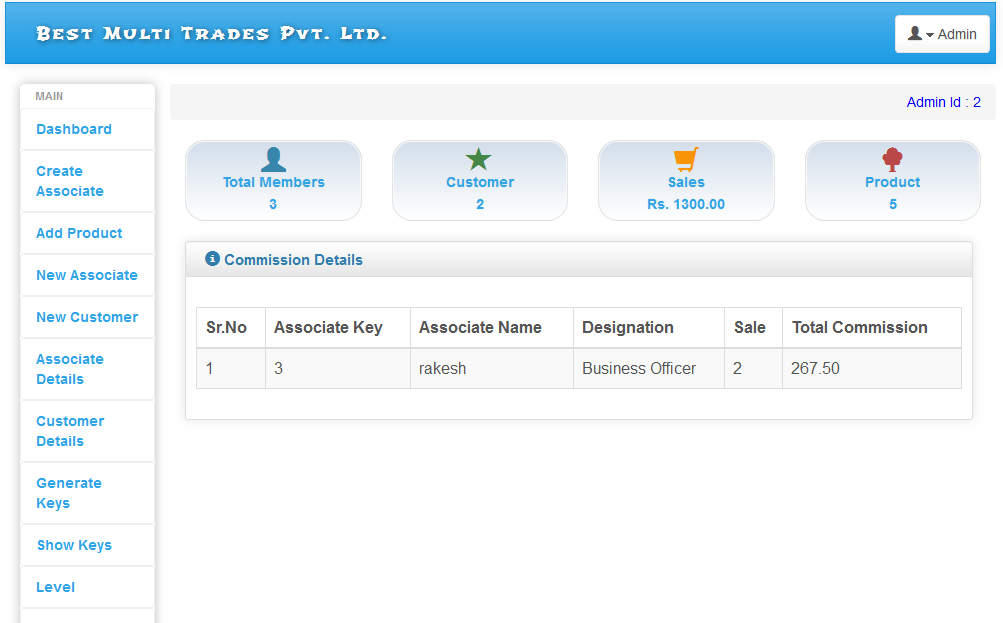
**Associate Details**

****

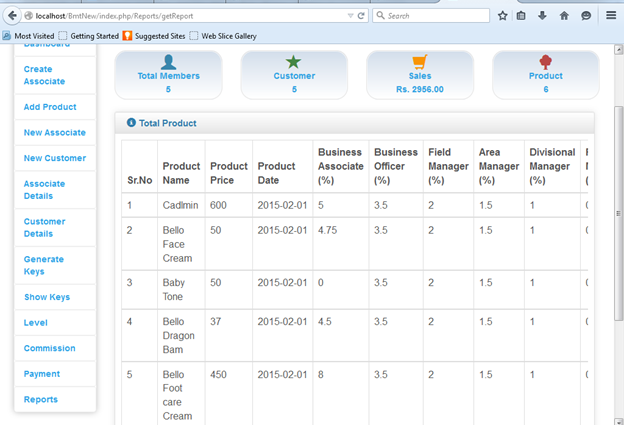
**Sale Product**

****

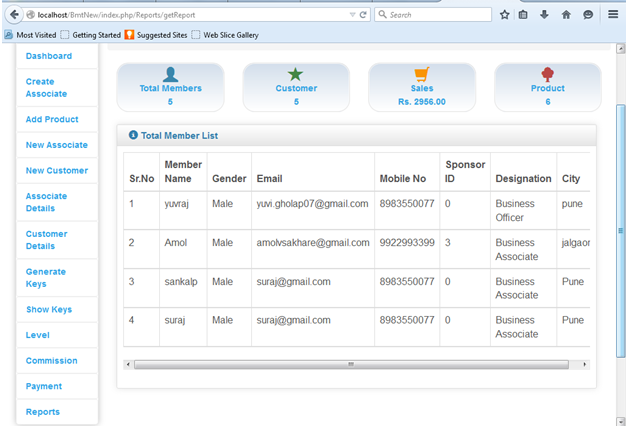
**Commission Details**

****

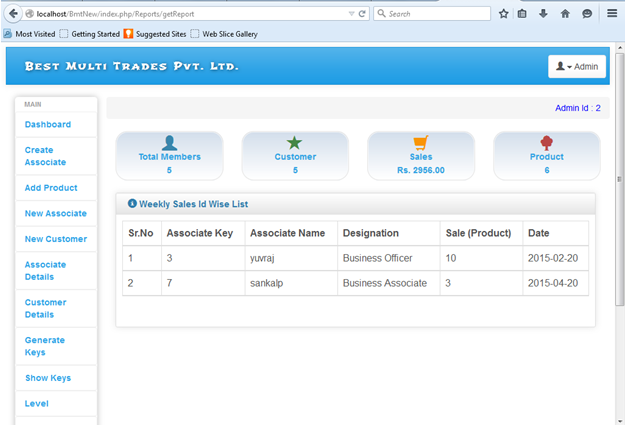
**Total Product Reports**

****

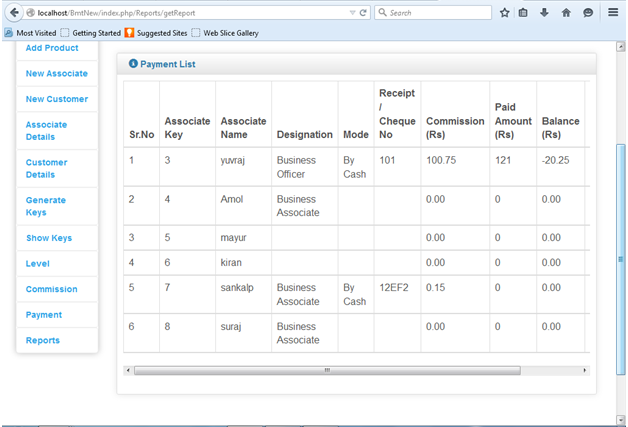
**Total Member List**

****

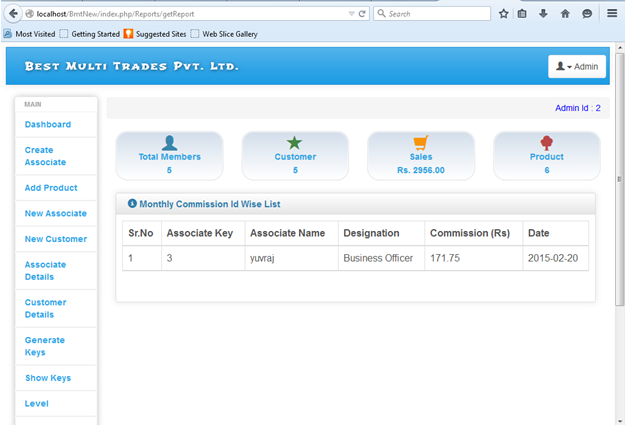
**Weekly sales Id wise**

****

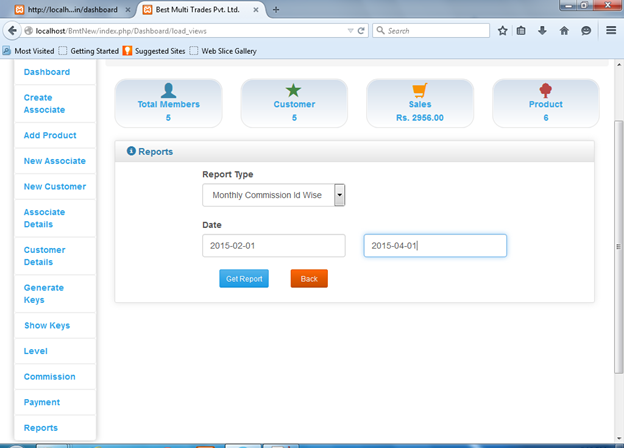
**Payment details**

****

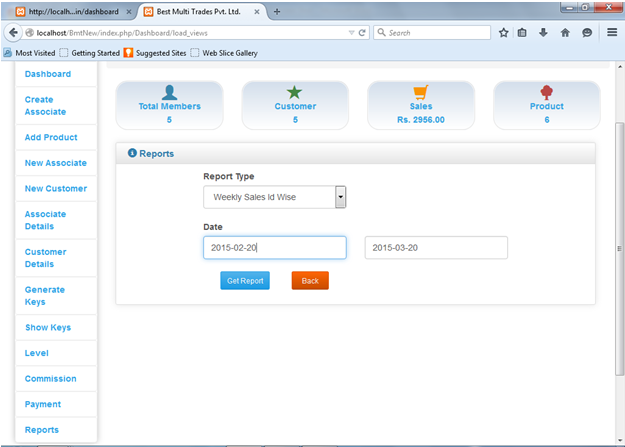
**Commission Id**

****

**Report Generate Date Wise**

****

**Sales Wise Report**

****

**Appendix C : List of tables**

**access\_function\_mngt**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr.No | Field Name | Data Type | Size | Constraint | Description |
| 1 | access\_id | Int | 5 | NOT NULL | Store access id. |
| 2 | function\_name | varchar | 20 | NOT NULL | Store function name. |
| 3 | Views | Varchar | 40 | DEFAULT NULL | Store views. |
| 4 | Images | Varchar | 30 | DEFAULT NULL | Store images. |
| 5 | No | Int | 5 | Primary key AUTO\_INCREMENT | Store no. |

**access\_type**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr.No | Field Name | Data Type | Size | Constraint | Description |
| 1 | access\_id | Int | 5 | Primary key | Store access id. |
| 2 | Type | varchar | 20 | DEFAULT NULL | Store type. |

**all\_commission**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr.No | Field Name | Data Type | Size | Constraint | Description |
| 1 | user\_id | Int | 10 | NOT NULL | Store user id. |
| 2 | Name | varchar | 50 | NOT NULL | Store name |
| 3 | Rank | int | 5 | NOT NULL | Store rank |
| 4 | commi\_amount | Float |  | NOT NULL | Store commi. amt. |
| 5 | created\_date | Date | date | NOT NULL | Store date. |

**customer**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr.No | Field Name | Data Type | Size | Constraint | Description |
| 1 | cust\_id | Int | 10 | Primary key AUTO\_INCREMENT | Store cust id. |
| 2 | customer\_name | varchar | 50 | NOT NULL | Store cusomer name. |
| 3 | customer\_addr | varchar | 70 | NOT NULL | Store cusomer address. |
| 4 | customer\_mob | varchar | 15 | NOT NULL | Store cusomer mob no. |
| 5 | Age | Int | 5 | NOT NULL | Store cusomer age. |
| 6 | Gender | varchar | 15 | NOT NULL | Store gender. |
| 7 | Town | varchar | 20 | NOT NULL | Store town. |
| 8 | District | varchar | 20 | NOT NULL | Store district |
| 9 | product\_key | varchar | 50 | DEFAULT NULL | Store product key. |
| 10 | product\_name | varchar | 50 | DEFAULT NULL | Store product name. |
| 11 | other\_disease | varchar | 70 | NOT NULL | Store other disease. |
| 12 | refrence\_id | Int | 10 | NOT NULL | Store refrence id. |
| 13 | status | Int | 5 | NOT NULL | Store staus. |
| 14 | created\_date | Date |  | NOT NULL | Store date. |

**payment**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr.No | Field Name | Data Type | Size | Constraint | Description |
| 1 | p\_id | Int | 5 | Primary key AUTO\_INCREMENT | Store payment id. |
| 2 | user\_id | Int | 5 | NOT NULL | Sore user id. |
| 3 | Name | Varchar | 50 | NOT NULL | Store name. |
| 4 | Rank | Varchar | 50 | NOT NULL | Store rank. |
| 5 | total\_pay | Varchar | 10 | NOT NULL | Store total pay. |
| 6 | Paid | Varchar | 10 | NOT NULL | Store paid amt. |
| 7 | Balance | Varchar | 10 | NOT NULL | Store balance amt. |
| 8 | Mode | Varchar | 10 | NOT NULL | Store mode. |
| 9 | No | Varchar | 50 | NOT NULL | Store no. |
| 10 | created\_date | Date |  | NOT NULL | Store created date. |

**product**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr.No | Field Name | Data Type | Size | Constraint | Description |
| 1 | Pid | Int | 11 | Primary key AUTO\_INCREMENT | Store product id. |
| 2 | product\_name | varchar | 50 | NOT NULL | Store prodt name. |
| 3 | product\_price | Float |  | NOT NULL | Store prodt price |
| 4 | entry\_date | Date |  | NOT NULL | Store entry date. |
| 5 | business\_associate | Float |  | NOT NULL | Store business asso. |
| 6 | business\_officer | float |  | NOT NULL | Store business offic. |
| 7 | field\_manager | float |  | NOT NULL | Store field mgr. |
| 8 | area\_manager | float |  | NOT NULL | Store area mgr. |
| 9 | divisional\_manager | float |  | NOT NULL | Store divisional mgr |
| 10 | regional\_manager | float |  | NOT NULL | Store reginal mgr. |
| 11 | silver\_achiever | Float |  | NOT NULL | Store silver achiever. |
| 12 | gold\_achiever | Float |  | NOT NULL | Store gold achiever. |
| 13 | platinum\_achiever | Float |  | NOT NULL | Store plati. achiever. |
| 14 | diamond\_achiever | Float |  | NOT NULL | Store dimnd achiever. |
| 15 | created\_by | Varchar | 50 | NOT NULL | Store created by. |
| 16 | Status | int | 5 | NOT NULL | Store status. |

**product\_keys**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr.No | Field Name | Data Type | Size | Constraint | Description |
| 1 | product\_id | int | 10 | NOT NULL | Store product id. |
| 2 | product\_keys | int | 10 | Primary key AUTO\_INCREMENT | Store product key. |
| 3 | Type | varchar | 15 | NOT NULL | Store product type. |
| 4 | user\_id | int | 10 | NOT NULL | Store user id. |
| 5 | Status | int | 2 | NOT NULL | Store status. |
| 6 | created\_by | int | 5 | NOT NULL | Store created by. |
| 7 | cur\_rank | int | 5 | NOT NULL | Store cur. rank. |
| 8 | post\_id\_1 | int | 10 | NOT NULL | Store post\_id\_1 |
| 9 | post\_1 | float |  | NOT NULL | Store  post\_1 |
| 10 | post\_id\_2 | int | 10 | NOT NULL | Store post\_id\_2 |
| 11 | post\_2 | float |  | NOT NULL | Store  post\_2 |
| 12 | post\_id\_3 | int | 10 | NOT NULL | Store post\_id\_3 |
| 13 | post\_3 | float |  | NOT NULL | Store  post\_3 |
| 14 | post\_id\_4 | int | 10 | NOT NULL | Store post\_id\_4 |
| 15 | post\_4 | float |  | NOT NULL | Store  post\_4 |
| 16 | post\_id\_5 | int | 10 | NOT NULL | Store post\_id\_5 |
| 17 | post\_5 | float |  | NOT NULL | Store  post\_5 |
| 18 | post\_id\_6 | int | 10 | NOT NULL | Store post\_id\_6 |
| 19 | post\_6 | float |  | NOT NULL | Store  post\_6 |
| 20 | post\_id\_7 | int | 10 | NOT NULL | Store post\_id\_7 |
| 21 | post\_7 | float |  | NOT NULL | Store  post\_7 |
| 22 | post\_id\_8 | int | 10 | NOT NULL | Store post\_id\_8 |
| 23 | post\_8 | float |  | NOT NULL | Store  post\_8 |
| 24 | post\_id\_9 | int | 10 | NOT NULL | Store post\_id\_9 |
| 25 | post\_9 | float |  | NOT NULL | Store  post\_9 |
| 26 | post\_id\_10 | int | 10 | NOT NULL | Store post\_id\_10 |
| 27 | post\_10 | float |  | NOT NULL | Store  post\_10 |
| 28 | created\_date | date |  | NOT NULL | Store created date. |

**user**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr.No | Field Name | Data Type | Size | Constraint | Description |
| 1 | user\_id | Int | 10 | Primary key AUTO\_INCREMENT | Store user id. |
| 2 | user\_name | varchar | 50 | NOT NULL | Store user name. |
| 3 | Password | varchar | 50 | NOT NULL | Store password. |
| 4 | access\_id | Int | 5 | NOT NULL | Store access id. |
| 5 | Status | Int | 5 | NOT NULL | Store status. |
| 6 | Gender | varchar | 15 | DEFAULT NULL | Store gender. |
| 7 | adjustable\_id | varchar | 5 | DEFAULT 0 | Store adjustable id. |
| 8 | sponser\_id | varchar | 10 | DEFAULT 0 | Store sponser id. |
| 9 | Dob | date |  | DEFAULT NULL | Store dob. |
| 10 | Age | Int | 5 | DEFAULT 0 | Store age. |
| 11 | Contact | varchar | 15 | NOT NULL | Store contact. |
| 12 | pan\_no | varchar | 30 | DEFAULT NULL | Store pan no. |
| 13 | Address | varchar | 70 | DEFAULT NULL | Store address. |
| 14 | District | varchar | 50 | DEFAULT NULL | Store district. |
| 15 | Town | varchar | 50 | DEFAULT NULL | Store town. |
| 16 | Email | varchar | 50 | DEFAULT NULL | Store email. |
| 17 | nominee\_name | varchar | 50 | DEFAULT NULL | Store nominee\_name |
| 18 | Relation | varchar | 50 | DEFAULT NULL | Store relation. |
| 19 | bank\_name | varchar | 50 | DEFAULT NULL | Store bank name. |
| 20 | account\_no | varchar | 40 | DEFAULT NULL | Store account no. |
| 21 | ifsc\_code | varchar | 20 | DEFAULT NULL | Store ifsc code. |
| 22 | created\_by | int | 10 | DEFAULT NULL | Store created by. |
| 23 | Rank | int | 10 | DEFAULT 0 | Store rank. |
| 24 | Photo | varchar | 150 | DEFAULT NULL | Store photo. |
| 25 | created\_date | date |  | NOT NULL | Store created date. |