Project Report

On

Retail Management System

Submitted To:

Mr. Sarabjit singh
Assistant Professor

Submitted By:

Sarbjot singh
Roll No: 2011CSC1086
MCA (FYIC)-6TH SEM

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

G.N.D.U, AMRITSAR.

INTRODUCTION

Management system is frame work of processes and procedures used to insure that an organisation can fulfil all tasks required to achieve its objective for instance an environmental management system enables organization to improve their environmental performance through a process of continuous improvement**.**

Store Management System enables retailers to streamline in-store activities, improve merchandise management, reduce labor costs, support remote store processes, and manage true store-level profit and loss. With our system, retailers can search order by order identity number, name, date, customer name, etc., accept, reject, and recall any order. One can even edit, add, and delete the categories and sub-categories of products within a store after getting the inputs from the Point of Sales Management System. Retailers can also get important statistics like sales conversion rates, commissions, third party sales and revenue reports

Indian retail sector has been growing by leaps and bounds in last decade or so. One research suggests that it will grow to $ 785 billion by 2015. Technology is and will play an important role in the Indian retail sector. Various groups in organized as well as the unorganised sector has taken to IT for supporting this growth.

Existing system

Before designing the new system, the existing system should be studied. The whole system developed first was not so much strong and totally file basis. I think that it is perfect in few years ago but now the days all work handle manually quite difficult and simply the previous fulfil all the requirements of the user. Software does not

The information of sold quantity and other details are stored in different files and so takes a long time to retrieve the data.it takes a long time to find the information about a relevant person.in case of a Delivery, the delivery will be held back.this results in a sharp drop in sales, unhappy customers and a bad impression on the company.

.

Problems in existing system

Some of the main drawbacks of file based system are as follows:

1. ***Data redundancy:***

   The same data has to be kept on several files which results in unnecessary data entry again and again, additional storage, more maintenance.

2. ***Lack of security:***

   The data stored using file based system is not secure as the data stored in books can be easily read by anyone with no restrictions.

3. ***Inaccuracies:***

   Since all the data entry is done manually in books or records, inaccuracies are common. The manual system is more prone to errors and inaccuracy.

4. ***Expensive:***

   Large number of personnel is required for each and every part of the manual system. We have to buy lot of books and paper for keeping the information. So the existing system is very expensive.

5. ***Inefficiency:***

   Large volume of data and unformatted outputs lead to inefficient decision making. Inefficiency in the existing system is also caused by the lack of proper communication among the people of the organization.

6. ***Time and effort:***

Large amount of time and efforts may be required in manual file based system. For searching a particular data, we have to search all the paper documents, its corresponding book and then find the data.

7. All the searching are done manually:

In the existing system, all the work like entering the details of a Customer, Booking details, Stock Details etc. are done manually. This is time consuming.

8. Updating problem:

In the existing system, since everything is stored in the books so every time the changes are required we have to update a particular record from every book maintaining that customer's record.

Proposed System

Retail management system is a database centred system which can be used for easy and efficient point of sale process . It allows to store all their information electronically including information on sales, customers, vendors and even allows for expansion of database scope to feature user developments. Most importantly this information is easily accessible to authorised user and provides a stone hard wall to unauthorised access.

It is highly flexible and can be configured to meet most individual's business needs. It controls the point of sale and inventory system. It include user log in feature and easy to use design for fast learning. It quickly generates customer histories and statements etc. Preview and print all sorts of reports including sales purchase customer reports etc and update notifications.

Key features

1) Complete insights into key products and service drivers. With the help of tables and symbols, marketers can effectively track and analyse factors that have an effect on important bottom lines like profitability. Store Managers can also effectively optimise product mix across channels, lines and brands with the product scorecards available. Some of the different KPIs that managers can avail of from product performance metrics are product sales by region, change in sales and margin per product, ROI per product, top competitor by product category and much more.
2) The entire organisation can access the same store data simultaneously and thus everyone has an understanding of what the customer wants. Managers can better monitor progress, respond immediately to customer needs, adjust parameters for continuous improvement, and exercise greater control over the organisation.
3) One can record and analyze inventory results and merchandise processes daily to know whether business decisions are based on timely, accurate information.
4) The interface is very user-friendly and use to use, hence it requires minimal training and IT experience for potential users.
5) Store Management System, being completely web-based, requires minimal software or hardware investment.

Interface Features

RETAIL CHAIN MANAGEMENT SYSTEM is a complete POS solution for small and midsize retailers operating everything from a single store to a whole chain. It is comprehensive and scalable for those businesses focusing on significant growth.

It offers advanced functionality such as multidimensional inventory management, customised reporting, advanced purchasing and receivables management .it integrates with well- known financial packages you may be using now.

It can be customised for your specific business requirements, providing you with investment protection of a flexible solution that adapts to meet demanding retail needs.

It as two modules/interfaces-

Admin interface

- Managing Stores
- Managing Store logins
- Managing Sections, categories, brands, products
- Managing Stock under various sections, categories, products
- Managing employees and attendance
- Announcing packages discount
- Managing Accounts of store
- Auditing accounts and inventory

Outlet interface

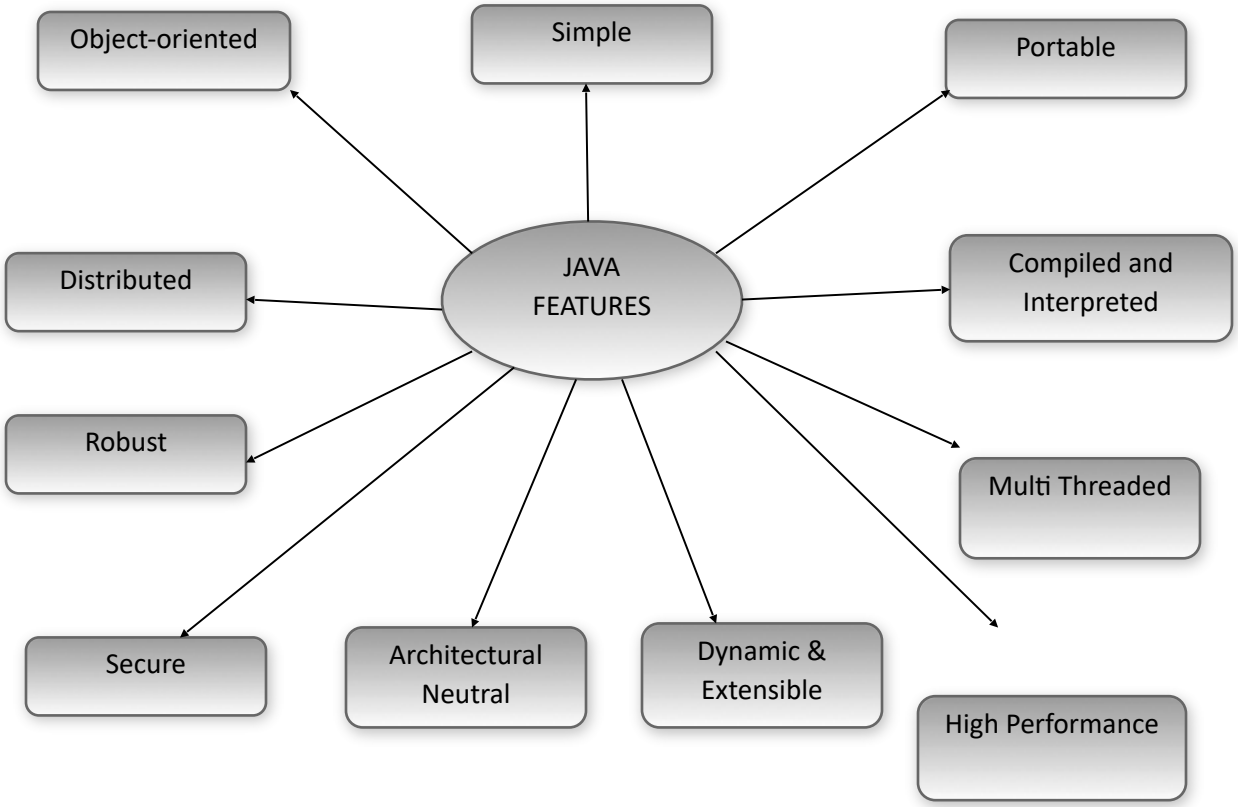
- Making customer membership
- Dealing with billing points schema
- Recording entry and exit of members
- Billings based on Barcodes
- Service desk management
- Sending schemes via sms and email
- Announcing scheme on public website
- Renewing customer membership

Front End

Java is a general purpose and the most popular object-oriented programming language. Java was developed by James Gosling and his colleagues at Sun Microsystems in the early 1990's.

Due to its simplicity and easy to learn and advanced features, we opted this language for our six months industrial training. This language supports many interesting features that make it an ideal language for software development. In addition to the object oriented features, it also provides features such as platform independence, security, multithreading, portability; etc which makes it well suited for the web and networked services, applications, platform-independent desktops, robotics and any other embedded devices.

Features of Java:

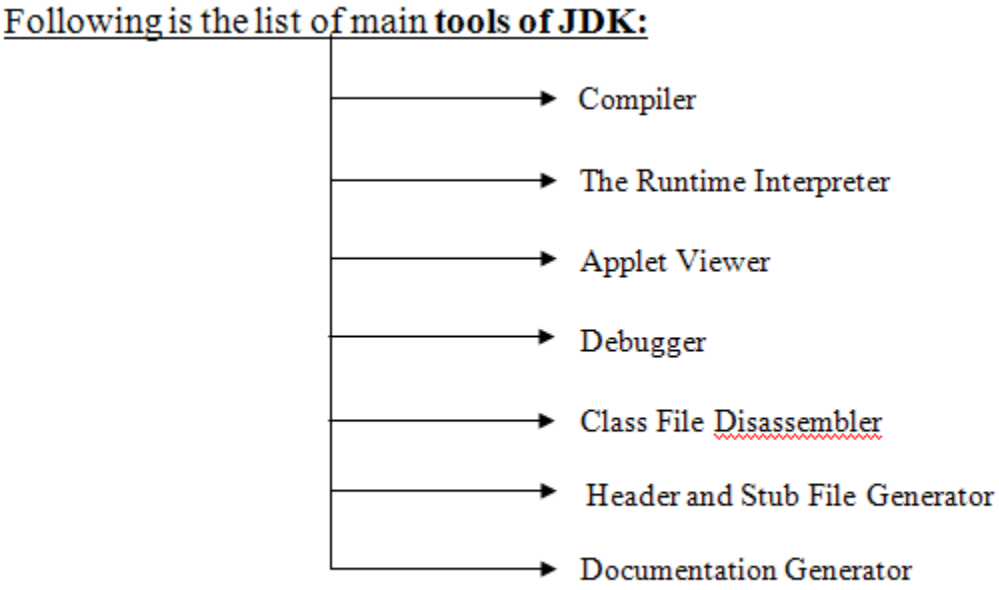


- **Simple**: Java is a compact and simple language. Programs are easy to write and debug as it omits many clumsy, poorly understood and confusing features of other programming languages such as C++.
- **Object-oriented:** Java is purely object-oriented language because programming in java is centered on creating objects; manipulating objects and making objects work together.
- **Distributed:** Java is a distributed language which means that the programs can be designed to run on computer networks. Java provides an extensive library of classes for communicating using TCP/IP protocols such as HTTP and FTP. This makes creating network connections much easier.

- **Robust:** Java is designed for writing programs that are highly robust. By robust, we mean reliable.
- **Secure:** As java is intended to be used in networked/distributed environments so it implements several security mechanisms to protect you against malicious code that might try to invade your file system.
- **Architectural Neutral:** This means that the programs written on one platform can run on any other platform without having to rewrite or recompile them. It follows 'Write-once-run-anywhere' approach.
- **Portable:** In Java, the size of the primitive data types is machine independent. These consistencies make java program portable among different platforms such as Windows, UNIX and Mac.
- **Interpreted:** Java is such a language that is both compiled and interpreted. The two steps of compilation and interpretation allow extensive code checking and improved security.
- **High performance:** Java programs are complied with portable intermediate form known as byte codes, rather than to native machine level instructions and JVM executes java byte codes on any machine on which it is installed. This architecture means that java programs are faster.
- **Multithreaded:** Java is also a multithreaded programming language. It allows you to write a program that can do many tasks simultaneously.
- **Dynamic:** Java is designed to be dynamic. Classes are stored in separate files and are loaded into the Java Interpreter only when they are needed.

JAVA DEVELOPMENT KIT (JDK)

The Java Development Kit (JDK) is a software package that sun has made available to public. It includes all the basic components that makeup the java environment. These include the Java compiler, Java Interpreter, an applet viewer that lets you see applets without opening a Java-compatible web browser.



APPLICATIONS OF JAVA

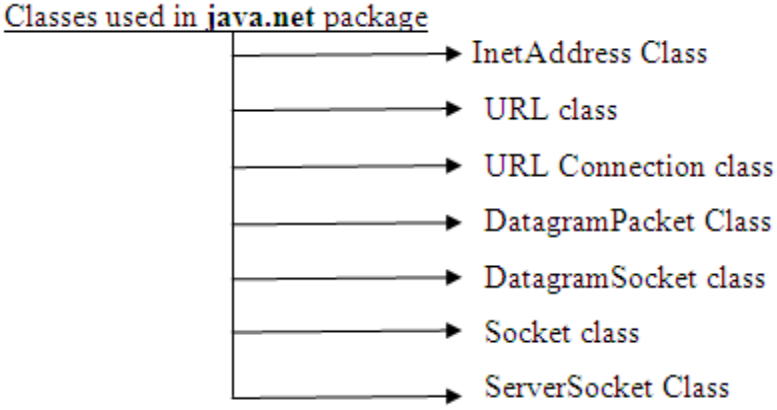
Java has evolved from a simple language providing interactive dynamic content for webpage's to a predominant enterprise-enables programming language suitable for developing significant and critical applications.

Today, Java is used for many applications like:

- Web based applications
- Financial applications
- Gaming applications
- Embedded applications
- Distributed enterprise applications
- Mobile applications
- Image applications
- E-business applications
- Desktop applications and many more.

JAVA.NET Package

The java.ne **on and working with network res**



Back End

MySQL

MySQL is a open source Relational Database Management System. MySQL is very fast reliable and flexible Database Management System. It provides a very high performance and it is multi threaded and multi user Relational Database management system.

MySQL is one of the most popular relational database Management System on the web. The MySQL Database has become the world's most popular open source Database, because it is free and available on almost all the platforms. The MySQL can run on Unix , window, and Mac OS. .

MySQL source code is available that's why now you can recompile the source code.

Features:

The following list describes some of the important **Features of MySQL** Database Software.

Internals and Portability

Works on many different platforms.

- The **MySQL** code is tested with Purify (a commercial memory leakage detector) as well as with Valgrind, a GPL tool
- The server is available as a separate program for use in a client/server networked environment. It is also available as a library that can be embedded (linked) into standalone applications. Such applications can be used in isolation or in environments where no network is available.

- Column Types
    - Many column types: signed/unsigned integers 1, 2, 3, 4, and 8 bytes long, FLOAT, DOUBLE, CHAR, VARCHAR, TEXT, BLOB, DATE, TIME, DATETIME, TIMESTAMP, YEAR, SET, ENUM, and OpenGIS spatial types.
    - Fixed-length and variable-length records.

- Statements and Functions
    - Full operator and function support in the SELECT and WHERE clauses of queries. For example:
    - Full support for SQL GROUP BY and ORDER BY clauses. Support for group functions (COUNT(), COUNT(DISTINCT ...), AVG(), STD(), SUM(), MAX(), MIN(), and GROUP_CONCAT()).
    - Support for LEFT OUTER JOIN and RIGHT OUTER JOIN with both standard SQL and ODBC syntax.
    - Support for aliases on tables and columns as required by standard SQL.
    - DELETE, INSERT, REPLACE, and UPDATE return the number of rows that were changed (affected). It is possible to return the number of rows matched instead by setting a flag when connecting to the server.
- Security
    - A privilege and password system that is very flexible and secure, and that allows host-based verification. Passwords are

secure because all password traffic is encrypted when you connect to a server.

- Scalability and Limits
    - Handles large databases. We use **MySQL** Server with databases that contain 50 million records. We also know of users who use **MySQL** Server with 60,000 tables and about 5,000,000,000 rows.
- Connectivity
    - Clients can connect to the **MySQL** server using TCP/IP sockets on any platform. On Windows systems in the NT family (NT, 2000, XP, or 2003), clients can connect using named pipes. On Unix systems, clients can connect using Unix domain socket files.
    - The Connector/J interface provides **MySQL** support for Java client programs that use JDBC connections. Clients can be run on Windows or Unix. Connector/J source is available.
- Localization
    - The server can provide error messages to clients in many languages.

HARDWARE REQUIREMENTS:

Hardware requirements include that hardware which is required for its working. It includes:

- ➢ Core2 due intel processor
- ➢ 512 MB RAM
- ➢ High Speed Internet Connection(DSL/Cable)

SOFTWARE REQUIREMENTS

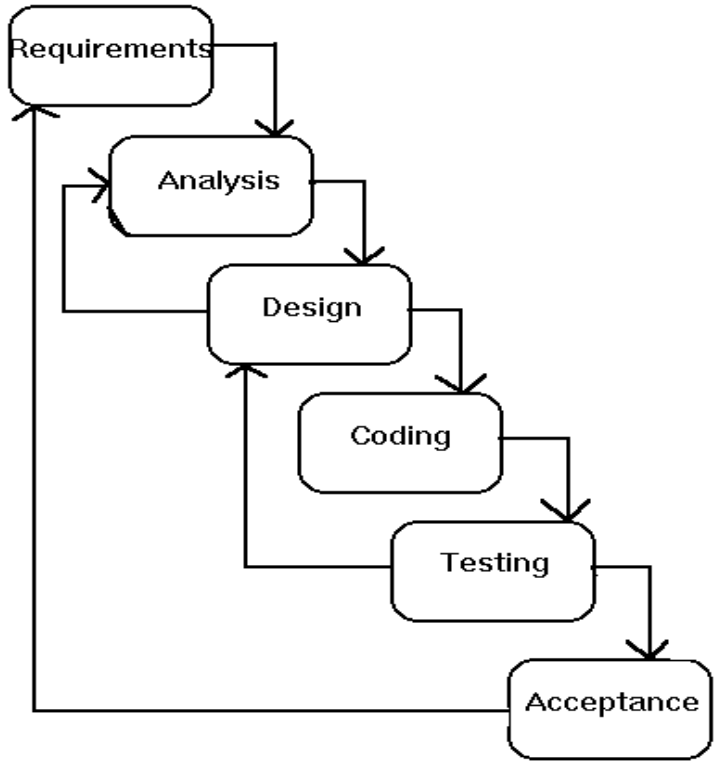
The technical specifications of requirements for the software are as follows:

- ➢ Any Operating System (Windows, Linux, MAC)
- ➢ Java run time environment
- ➢ Netbeans (Java IDE)
- ➢ Java SDK (Software Development Kit)
- ➢ Any web browser(Chrome , Firefox , etc)

Software Model Followed

Waterfall Model

The waterfall model is a sequential design process, often used in software development processes, in which progress is seen as flowing steadily downwards through the phases of Conception, Initiation, Analysis, Design, Coding, Testing, Implementation and Maintenance



The waterfall model (Systems Development Life Cycle)

In waterfall model, the following phases are followed in order:

- Requirements specification
- Feasibility Study
- Design & Coding
- Integration & Testing
- Implementation
- Maintenance

Requirement Analysis & Definition*:* All possible requirements of the system to be developed are captured in this phase. Requirements are set of functionalities and constraints that the end-user expects from the system. The requirements are gathered from the end-user by consultation, these requirements are analyzed for their validity and the possibility of incorporating the requirements in the system to be development is also studied.

System & Software Design: Before a starting for actual coding, it is highly important to understand what we are going to create and what it should look like? The requirement specifications from first phase are studied in this phase and system design is prepared which helps in specifying hardware and system requirements and also helps in defining overall system architecture.

Implementation & Unit Testing: On receiving system design documents, the work is divided in modules/units and actual coding is started. The system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality; this is referred to as Unit Testing. Unit testing mainly verifies if the modules/units meet their specifications.

Integration & System Testing: The units earlier created are now integrated into a complete system during and tested to check if all modules/units coordinate between each other and the system as a whole behaves as per the specifications. After successfully testing the software, it is delivered to the customer.

Operations & Maintenance: This phase is virtually never ending phase. Generally, problems with the system developed come up after its practical use starts, so the issues related to the system are solved after deployment of the system. Not all the problems come in picture directly but they arise time to time and needs to be solved; hence this process is referred as Maintenance.

Advantages

The advantage of waterfall development is that it allows for departmentalization and managerial control. A schedule can be set with deadlines for each stage of development and a product can proceed through the development process. Development moves from concept, through design, implementation, testing, installation, troubleshooting, and ends up at operation and maintenance. Each phase of development proceeds in strict order, without any overlapping or iterative steps.

Disadvantages

The disadvantage of waterfall development is that it does not allow for much reflection or revision. Once an application is in the testing stage, it is very difficult to go back and change something that was not well-thought out in the concept stage.

Why we use waterfall model?

As it is a minor project and being a beginner, we already have the requirements for our ongoing project. Waterfall model is considered to be of downward approach and we don't have to look up to the previous level that frequently, it is beneficial for our project. Also waterfall model with modern approach provides a provision to have a check onto every level, once a cycle is completed. Thus if we want to modify anything within our project after acceptance, we can start from the initial phase. Thus it does not freeze the possibility for any kind of change.

DataBase Tables

1). Admin/employee login

Purpose: login information of employee and admin

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
username	VARCHAR(123)	☑	☑	☐	☐	☐	☐	☐	
password	VARCHAR(45)	☐	☑	☐	☐	☐	☐	☐	
emailid	VARCHAR(45)	☐	☑	☐	☐	☐	☐	☐	
phonenumber	BIGINT(20)	☐	☑	☐	☐	☐	☐	☐	
gender	VARCHAR(45)	☐	☑	☐	☐	☐	☐	☐	
address	VARCHAR(45)	☐	☑	☐	☐	☐	☐	☐	
sec_ques	VARCHAR(45)	☐	☑	☐	☐	☐	☐	☐	
sec_ans	VARCHAR(45)	☐	☑	☐	☐	☐	☐	☐	
photo	VARCHAR(200)	☐	☐	☐	☐	☐	☐	☐	NULL

2). Attendance

Purpose: attendance record of various employees

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
attendenceid	INT(11)	☑	☑	☐	☐	☐	☐	☑	
username	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	NULL
date_of_attendence	TIMESTAMP	☐	☑	☐	☐	☐	☐	☐	CURRENT_TIMESTA...
		☐	☐	☐	☐	☐	☐	☐	

3. Section

Purpose: record of various sections in their business.

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
section_name	VARCHAR(32)	☑	☑	☐	☐	☐	☐	☐	
description	VARCHAR(145)	☐	☐	☐	☐	☐	☐	☐	NULL
location	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	NULL
incharge	VARCHAR(45)	☐	☑	☐	☐	☐	☐	☐	

4. Categories

Purpose: record of various categories under a section.

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI
category_name	VARCHAR(123)	☑	☑	☐	☐	☐	☐	☐
category_description	VARCHAR(45)	☐	☑	☐	☐	☐	☐	☐
category_section	VARCHAR(45)	☐	☑	☐	☐	☐	☐	☐

5. Products

Purpose: record of various products available for selling.

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
product_id	INT(11)	☑	☑	☐	☐	☐	☐	☑	
product_name	VARCHAR(45)	☐	☑	☐	☐	☐	☐	☐	
barcode	VARCHAR(45)	☐	☑	☐	☐	☐	☐	☐	
price	DECIMAL(6,2)	☐	☑	☐	☐	☐	☐	☐	
discount	DECIMAL(6,2)	☐	☐	☐	☐	☐	☐	☐	NULL
netprice	DECIMAL(6,2)	☐	☐	☐	☐	☐	☐	☐	NULL
category	VARCHAR(45)	☐	☑	☐	☐	☐	☐	☐	
brands	VARCHAR(45)	☐	☑	☐	☐	☐	☐	☐	
photo	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	NULL

6. Brands

Purpose: brands available under different products.

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
brand_name	VARCHAR(45)	☑	☑	☐	☐	☐	☐	☐	
brand_description	VARCHAR(45)	☐	☑	☐	☐	☐	☐	☐	
brand_pic	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	NULL

7. Stock

Purpose: availability of stock of different products.

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI
stock_id	INT(11)	☑	☑	☐	☐	☐	☐	☑
product_id	INT(11)	☐	☑	☐	☐	☐	☐	☐
quantity	DECIMAL(10,0)	☐	☑	☐	☐	☐	☐	☐
date_of_stockadd	DATE	☐	☑	☐	☐	☐	☐	☐

8. Bad stock

Purpose: contain record of defected products.

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI
bad_stock_id	INT(11)	☑	☑	☐	☐	☐	☐	☑
stock_id	INT(11)	☐	☑	☐	☐	☐	☐	☐
quantity	DECIMAL(10,0)	☐	☑	☐	☐	☐	☐	☐
supervisedby	VARCHAR(45)	☐	☑	☐	☐	☐	☐	☐

9. Bill

Purpose: information of sold quantity.

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
billid	INT(11)	☑	☑	☐	☐	☐	☐	☑	
date	TIMESTAMP	☐	☑	☐	☐	☐	☐	☐	CURRENT_TIMESTA...
totalprice	DECIMAL(10,0)	☐	☑	☐	☐	☐	☐	☐	
supervisor	VARCHAR(45)	☐	☑	☐	☐	☐	☐	☐	

10. Bill details

Purpose: detailed information of generated bill.

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI
billdetailid	INT(11)	☑	☑	☐	☐	☐	☐	☑
bill_id	INT(11)	☐	☑	☐	☐	☐	☐	☐
barcode	VARCHAR(45)	☐	☑	☐	☐	☐	☐	☐
quantity	DECIMAL(10,0)	☐	☑	☐	☐	☐	☐	☐
price	DECIMAL(10,0)	☐	☑	☐	☐	☐	☐	☐

11. Members

Purpose: members detail which are connected to our bussiness

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
memberid	INT(11)	☑	☑	☐	☐	☐	☐	☑	
name	VARCHAR(45)	☐	☑	☐	☐	☐	☐	☐	
email	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	NULL
mobile	VARCHAR(15)	☐	☑	☐	☐	☐	☐	☐	
dateofbirth	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	NULL
gender	VARCHAR(5)	☐	☑	☐	☐	☐	☐	☐	
marital_status	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	NULL
anniversary_date	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	NULL

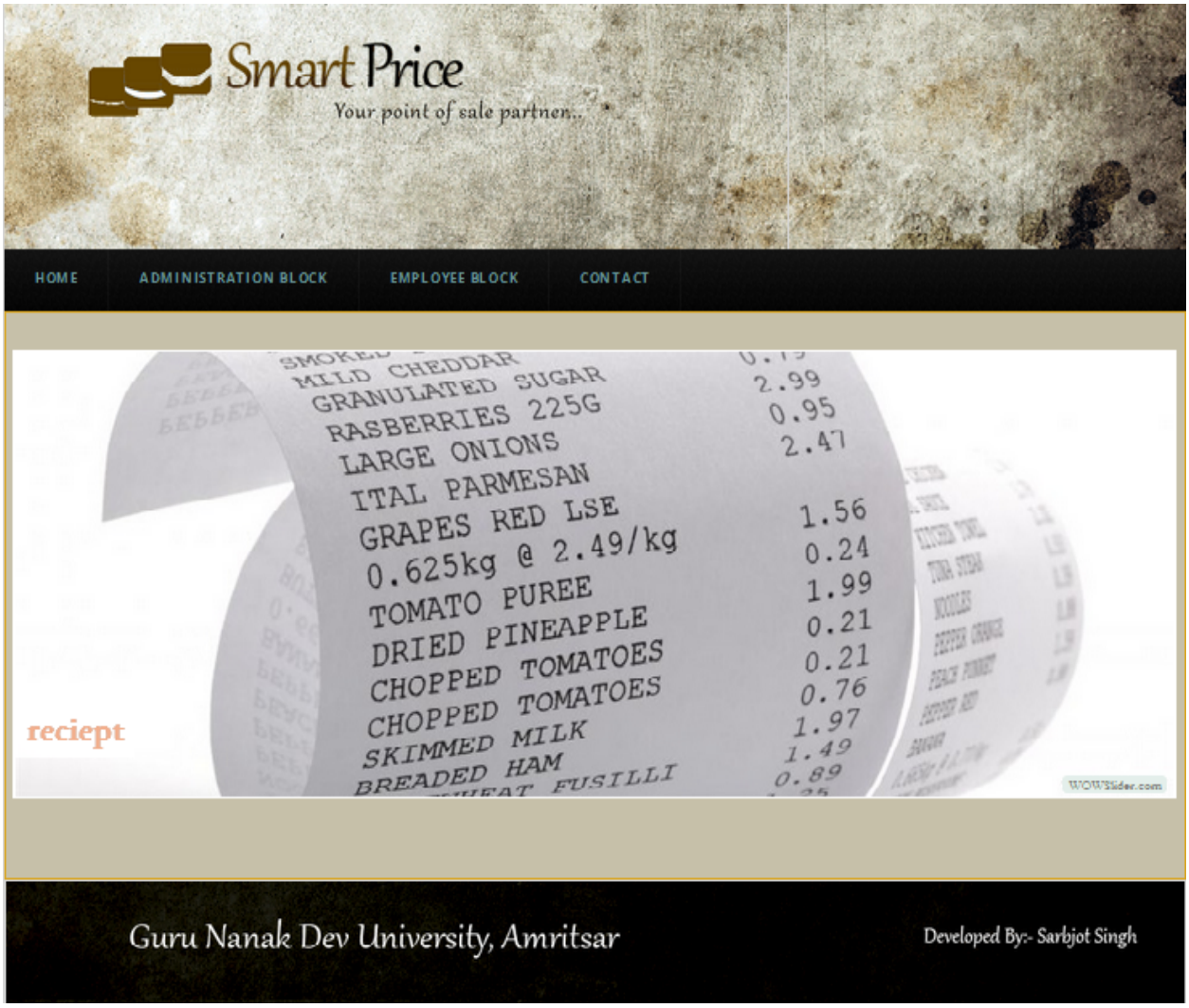
12. Points

Purpose: cointaning points of different members according to their bill.

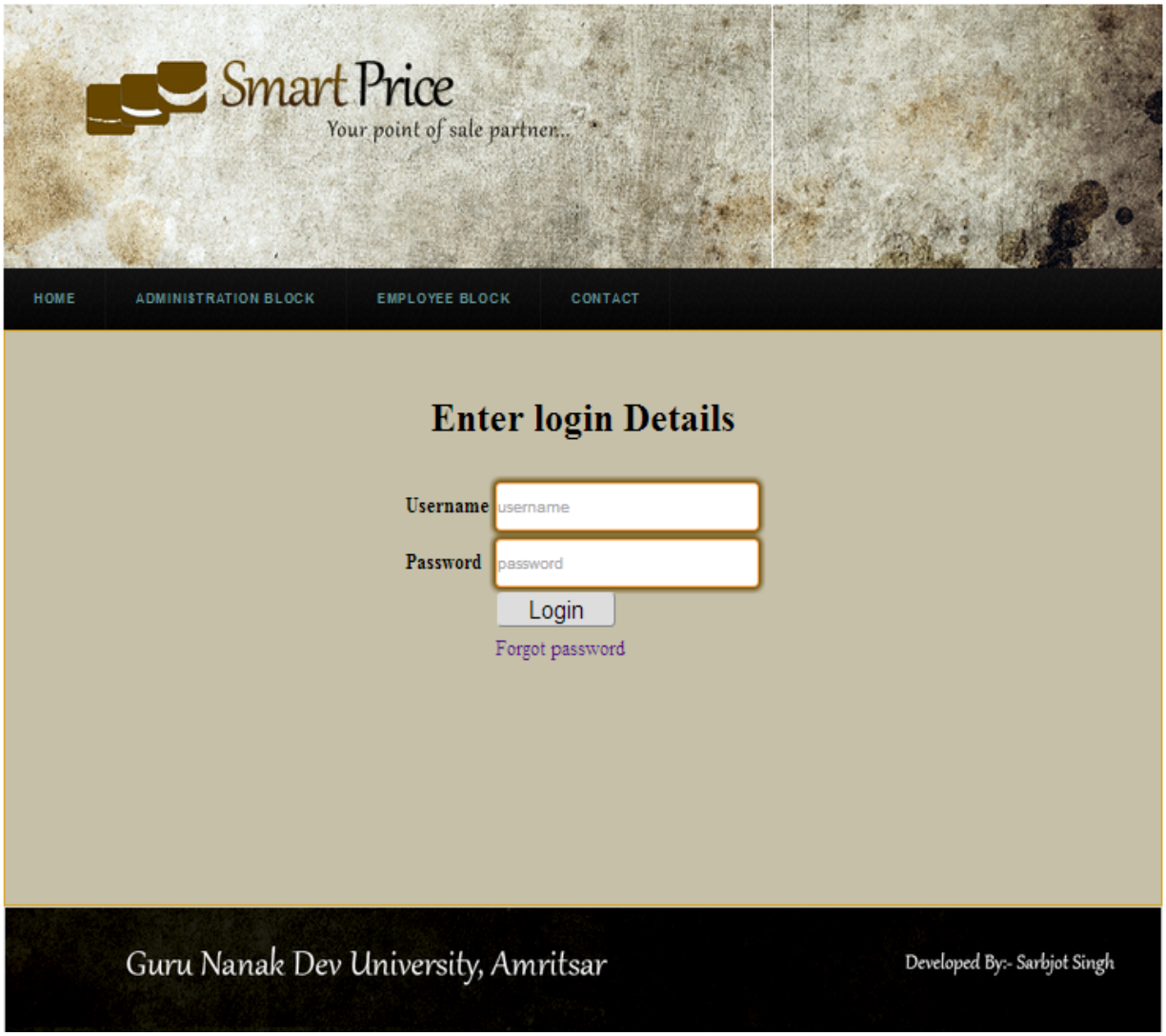
Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
pointsid	INT(11)	☑	☑	☐	☐	☐	☐	☑	
dateofpoints	TIMESTAMP	☐	☑	☐	☐	☐	☐	☐	CURRENT_TIMESTA...
billid	INT(11)	☐	☑	☐	☐	☐	☐	☐	
points	INT(11)	☐	☑	☐	☐	☐	☐	☐	
mid	INT(11)	☐	☑	☐	☐	☐	☐	☐	

SCREEN SHORTS

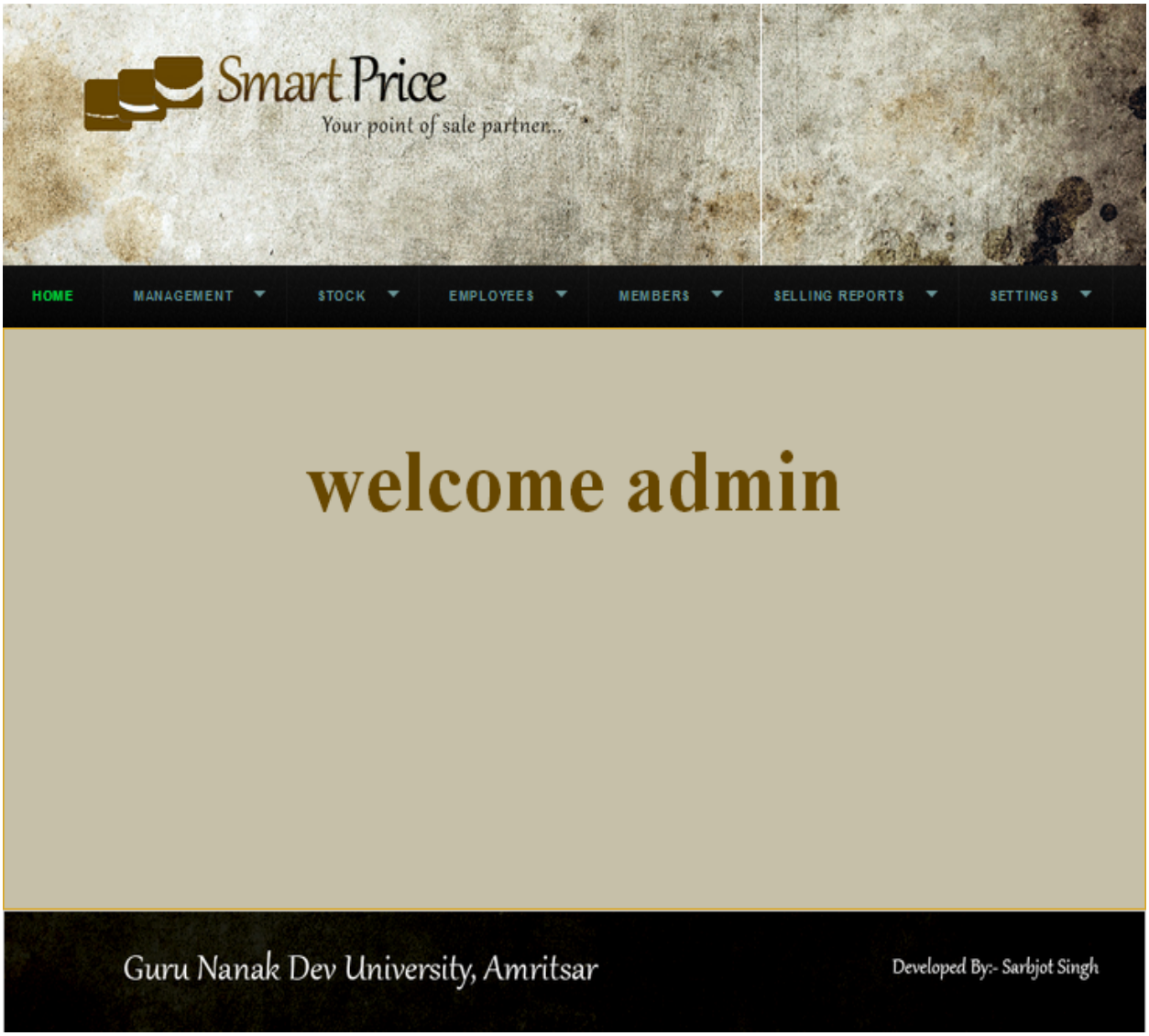
HOME



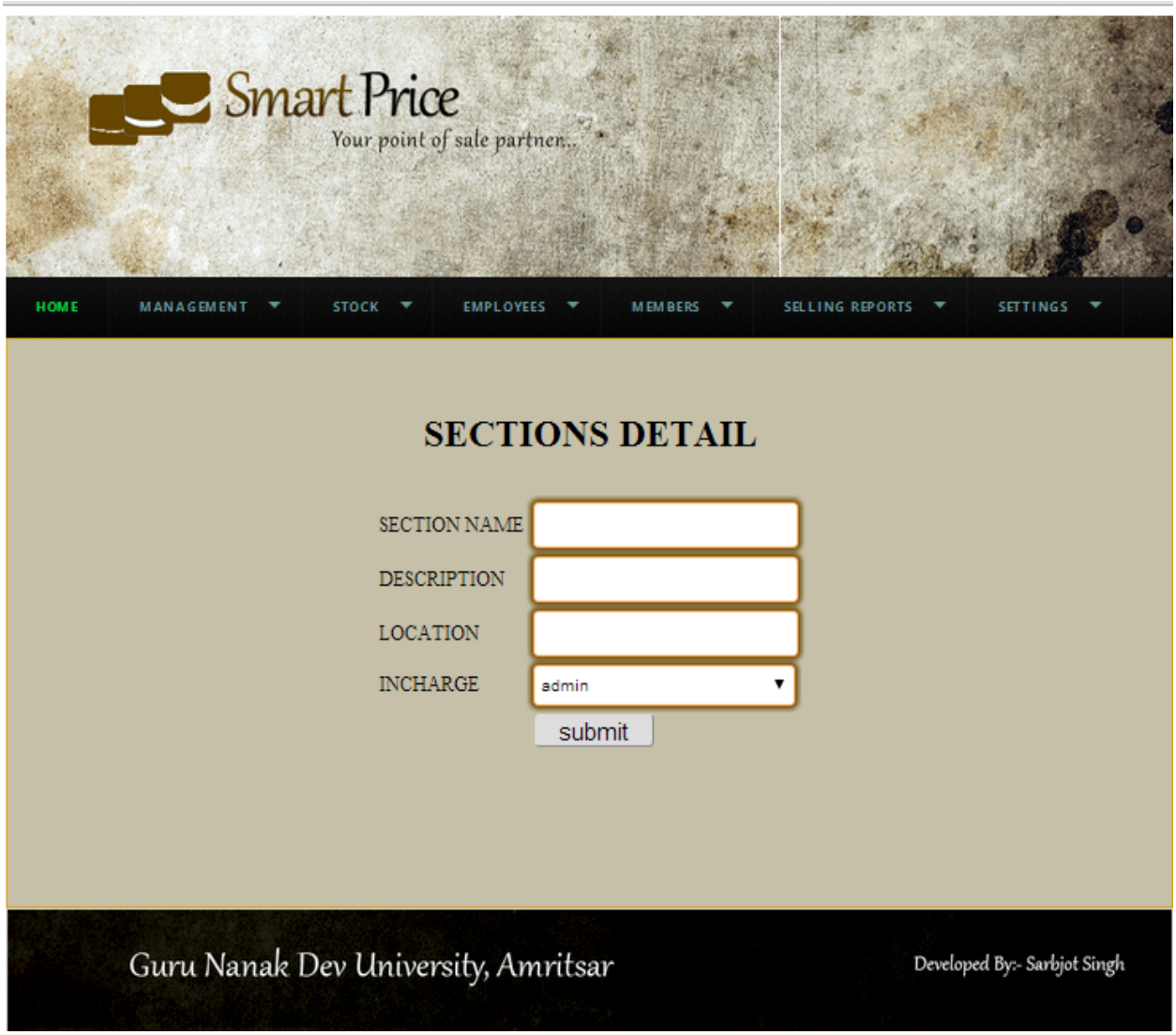
Login


Smart Price
Your point of sale partner...
HOME
ADMINISTRATION BLOCK
EMPLOYEE BLOCK
CONTACT
Enter login Details
Username
username
Password
password
Login
Forgot password
Guru Nanak Dev University, Amritsar
Developed By:- Sarbjot Singh

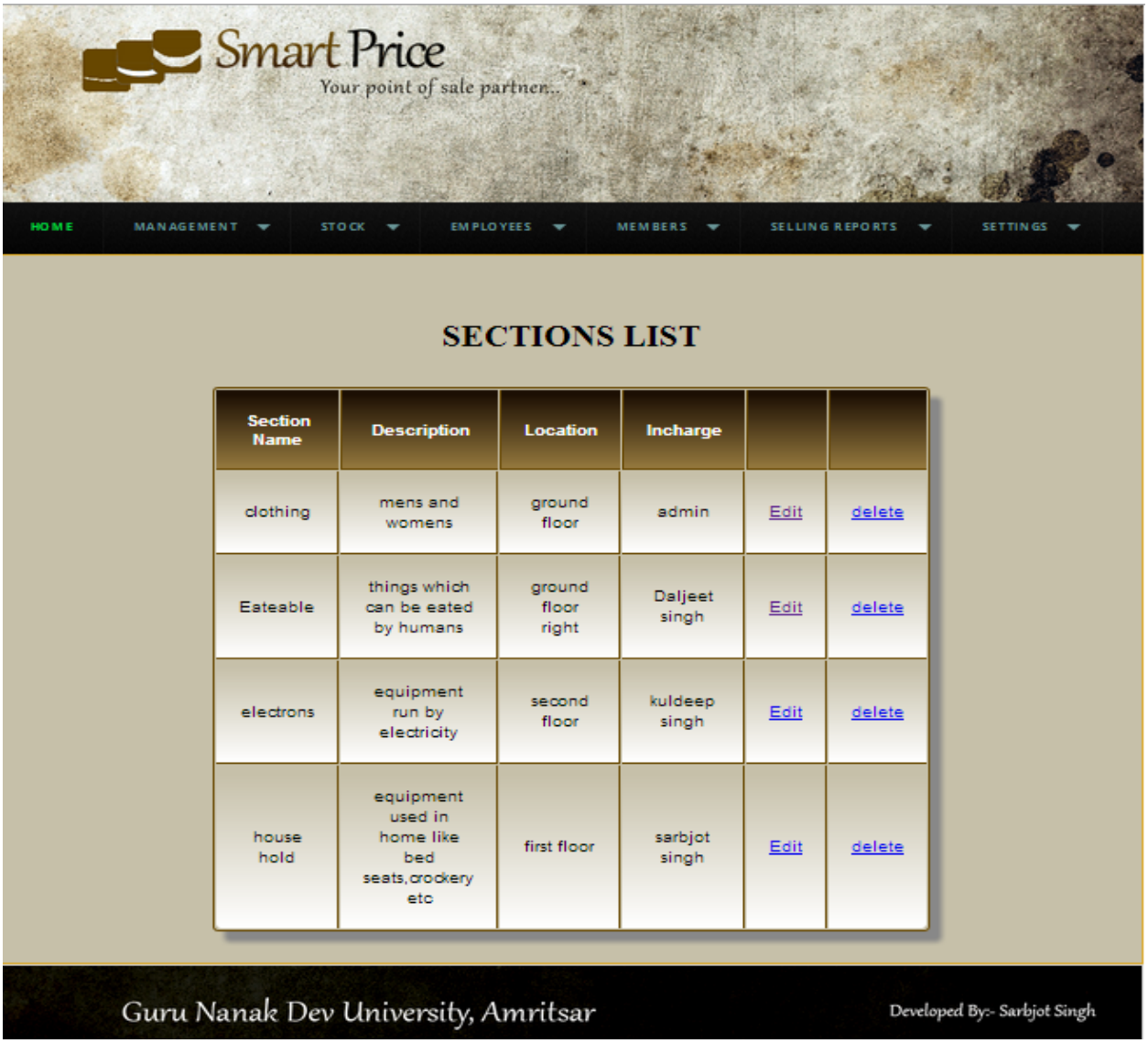
Admin Home


Smart Price
Your point of sale partner...
HOME
MANAGEMENT
STOCK
EMPLOYEES
MEMBERS
SELLING REPORTS
SETTINGS
welcome admin
Guru Nanak Dev University, Amritsar
Developed By:- Sarbjot Singh

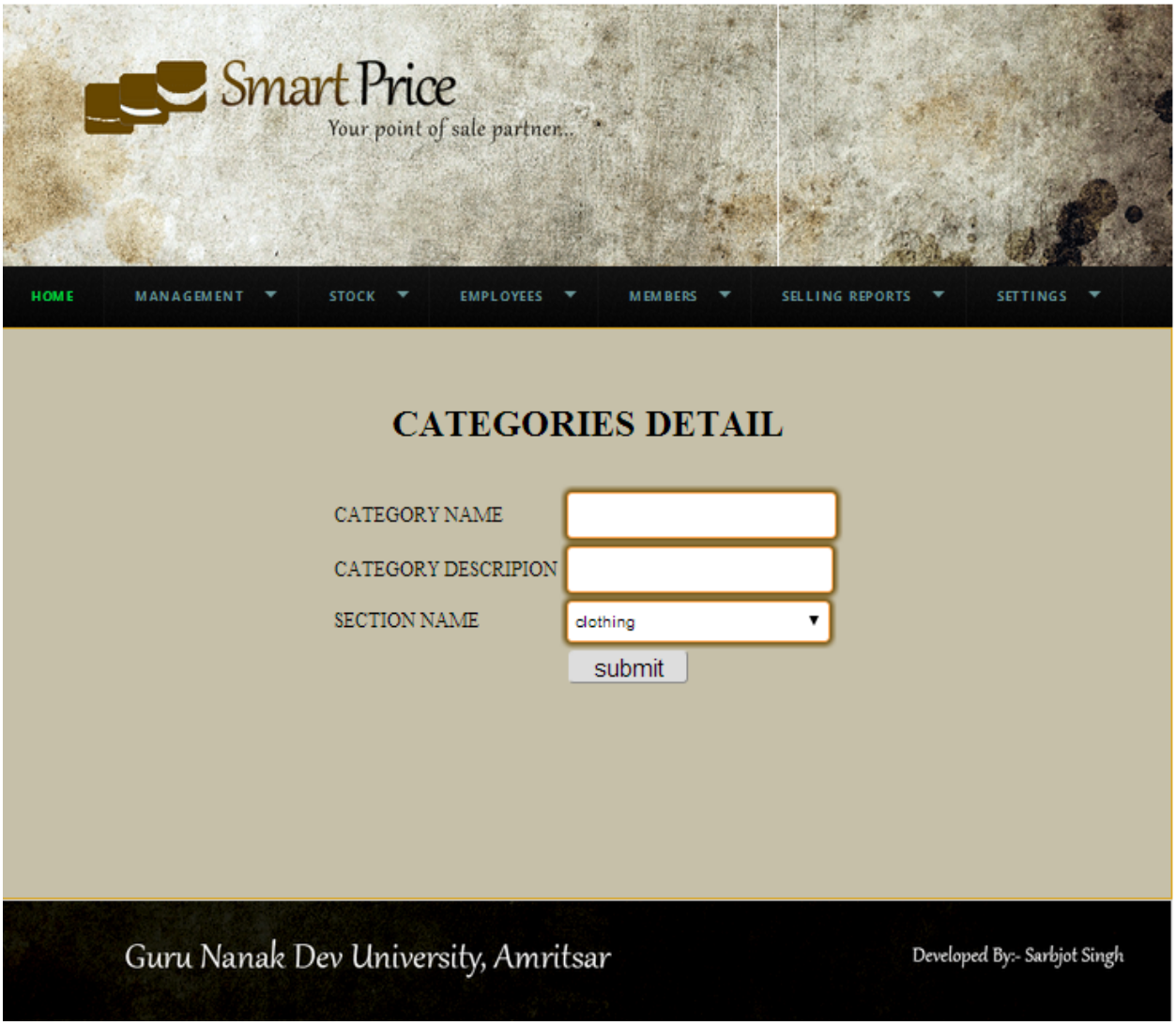
Add Sections



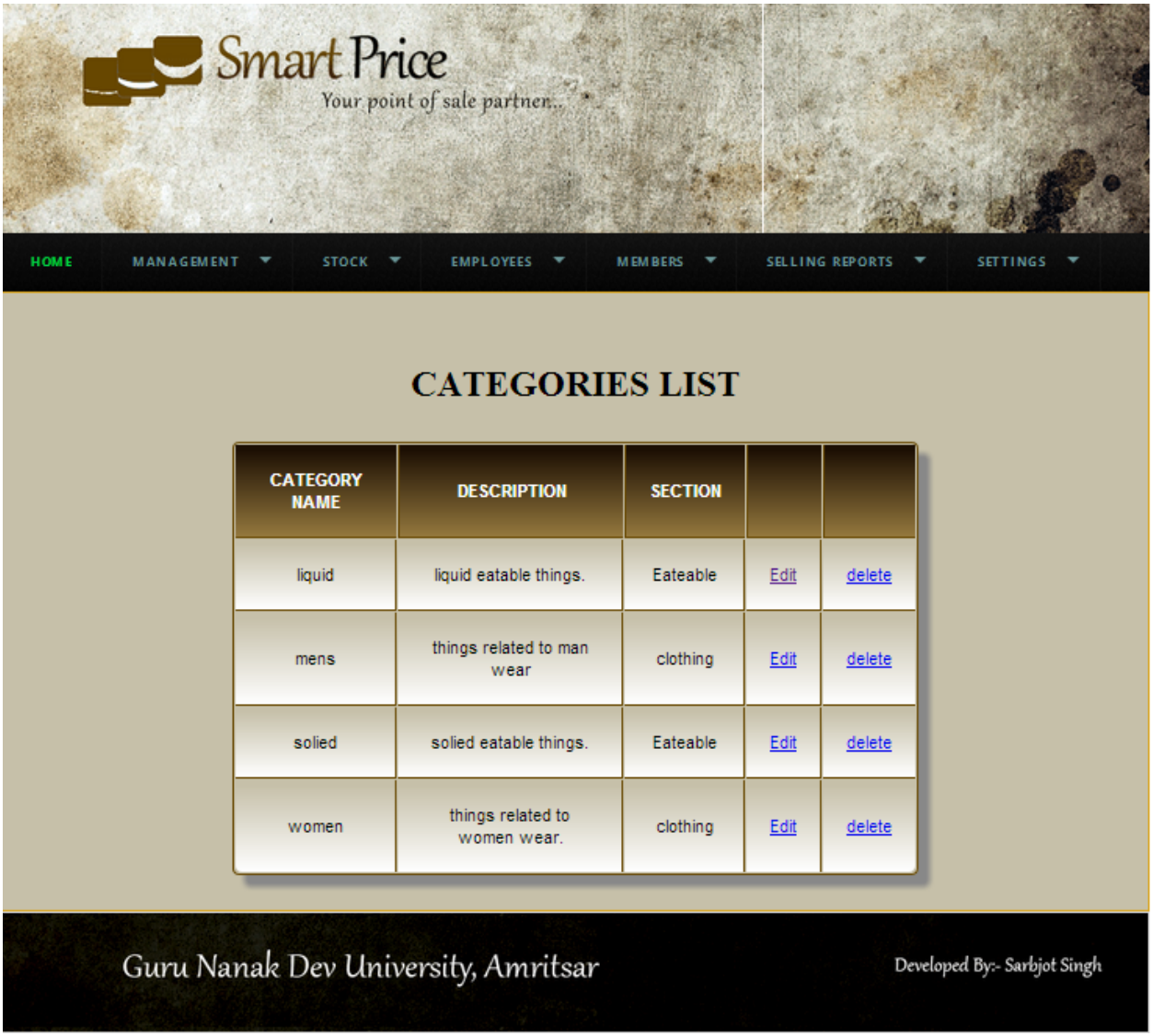
View Sections



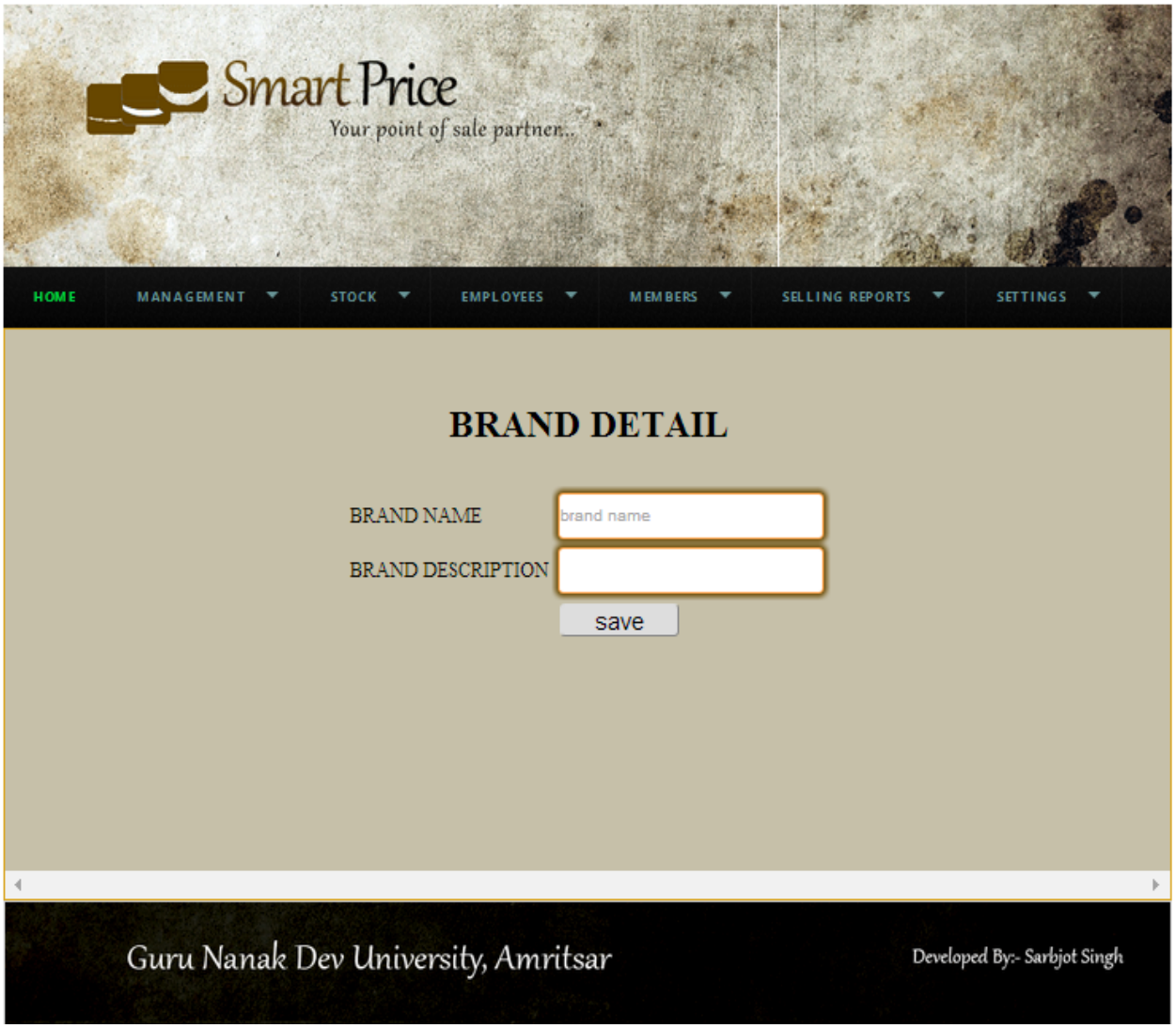
Add Categories



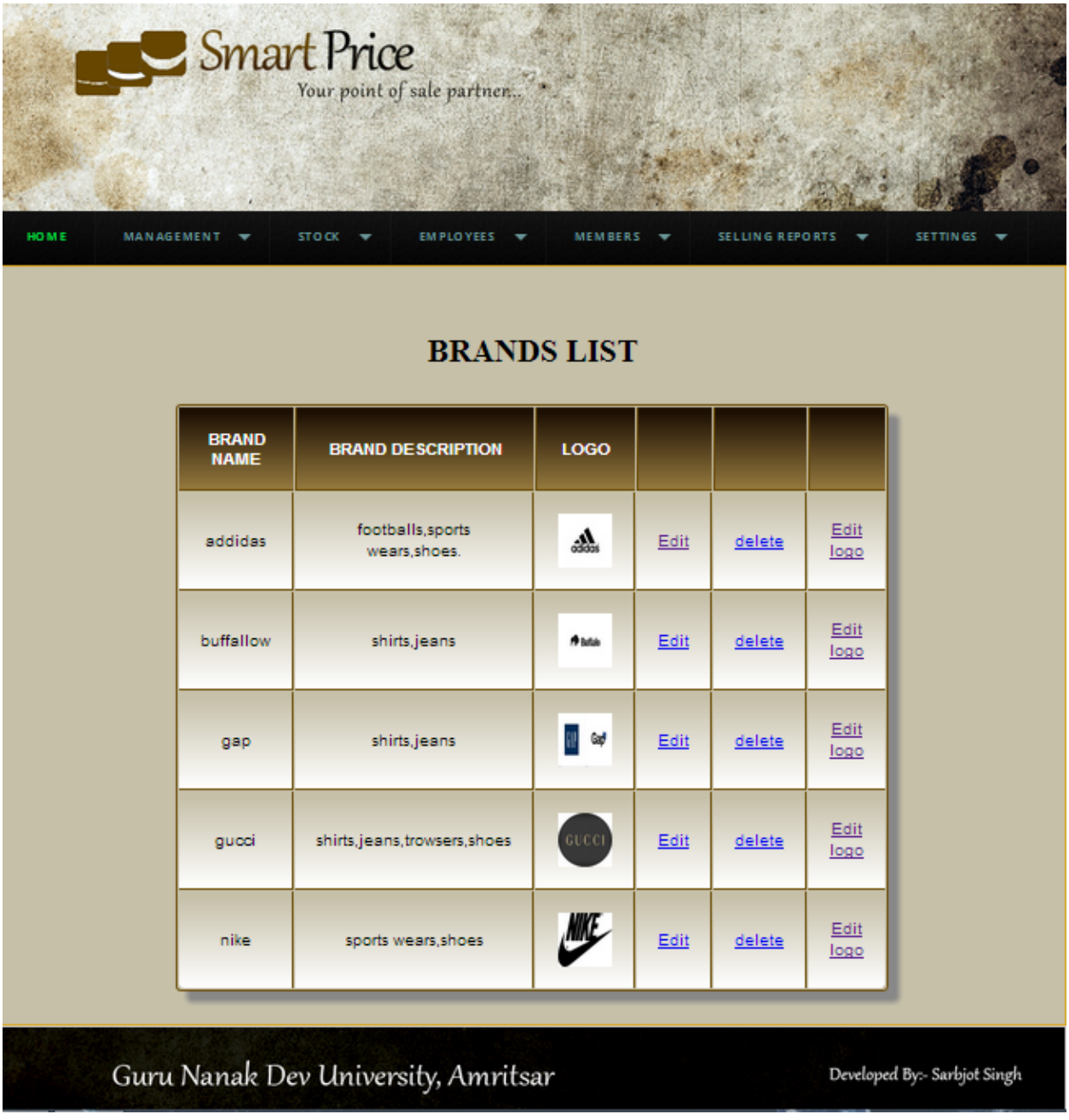
View Categories



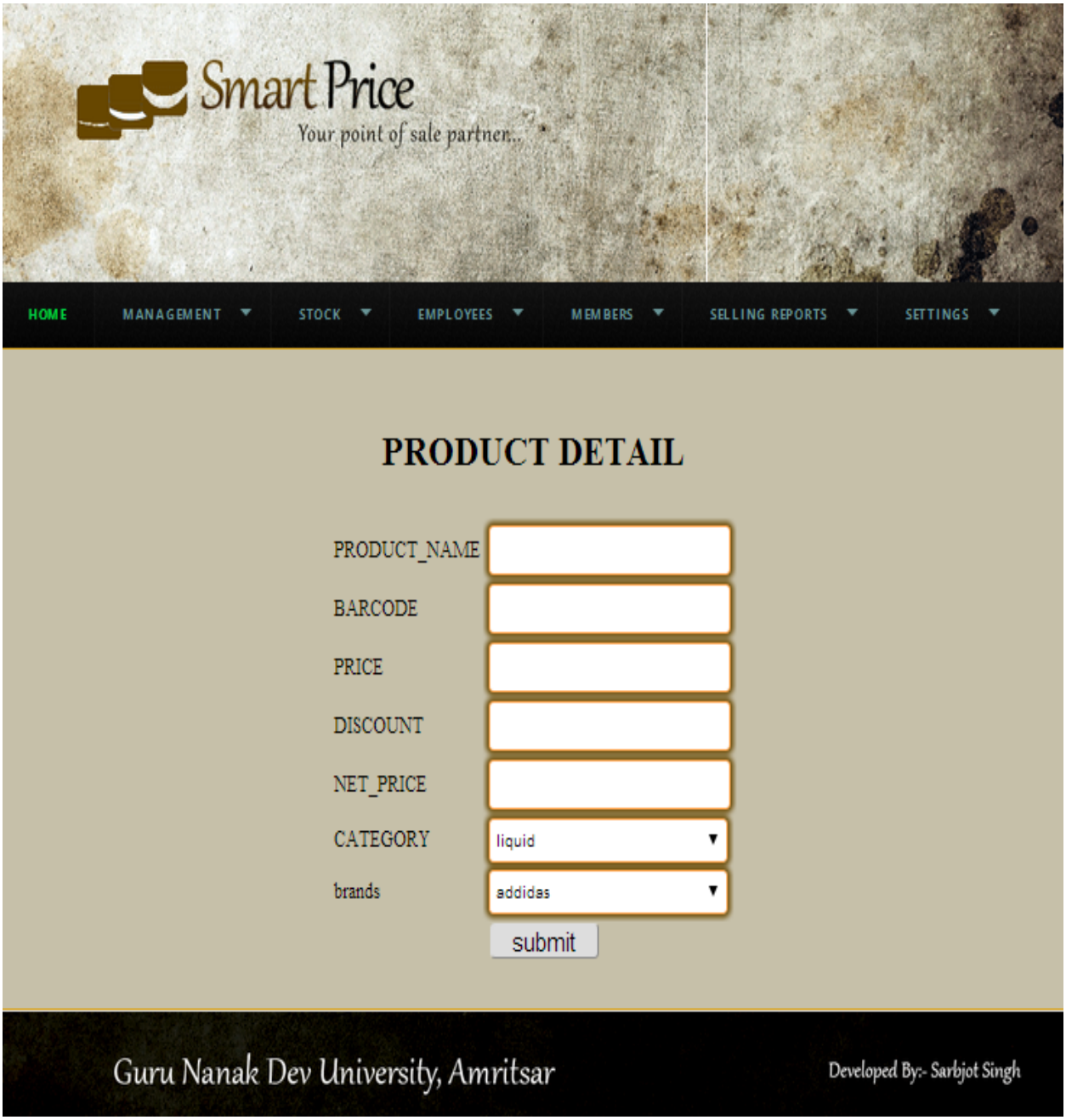
Add Brands


Smart Price
Your point of sale partner...
HOME
MANAGEMENT
STOCK
EMPLOYEES
MEMBERS
SELLING REPORTS
SETTINGS
BRAND DETAIL
BRAND NAME
brand name
BRAND DESCRIPTION
save
Guru Nanak Dev University, Amritsar
Developed By:- Sarbjot Singh

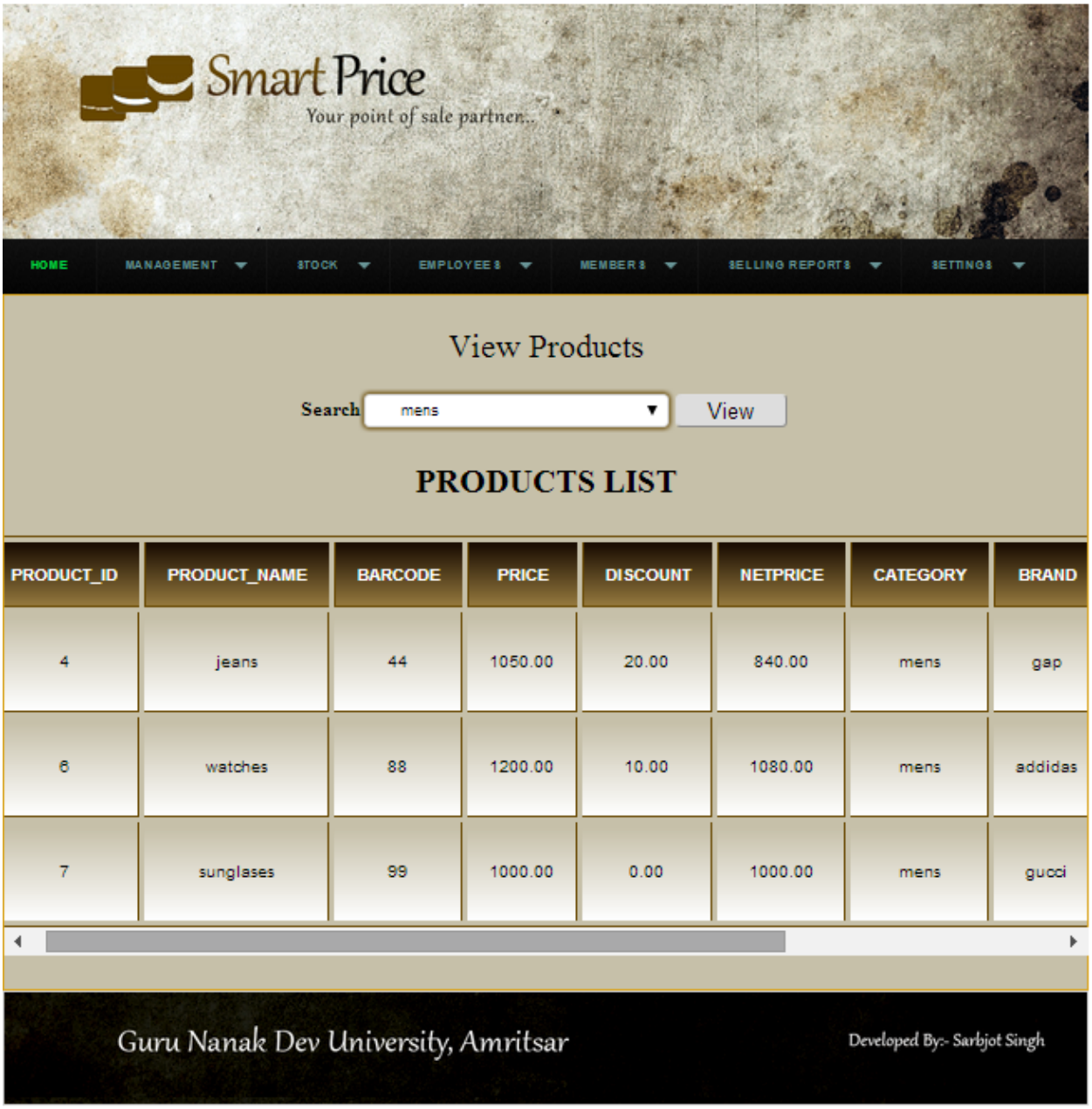
View Brands


Smart Price
Your point of sale partner...
HOME
MANAGEMENT
STOCK
EMPLOYEES
MEMBERS
SELLING REPORTS
SETTINGS
BRANDS LIST
BRAND NAME
BRAND DESCRIPTION
LOGO
addidas
footballs,sports wears,shoes.
Edit
delete
Edit logo
buffallow
shirts,jeans
Edit
delete
Edit logo
gap
shirts,jeans
Edit
delete
Edit logo
gucci
shirts,jeans,trowsers,shoes
Edit
delete
Edit logo
nike
sports wears,shoes
Edit
delete
Edit logo
Guru Nanak Dev University, Amritsar
Developed By:- Sarbjot Singh

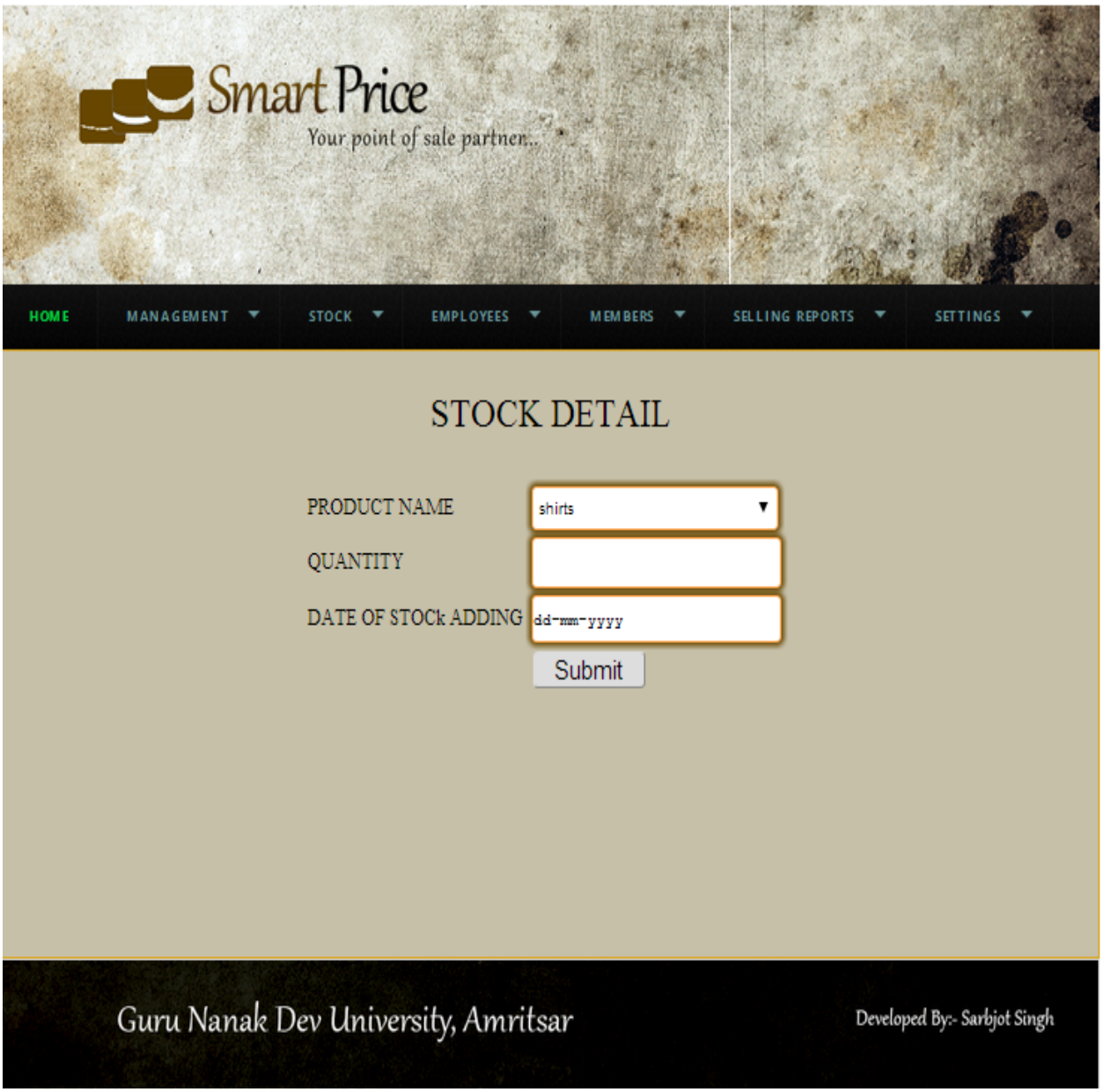
Add Products



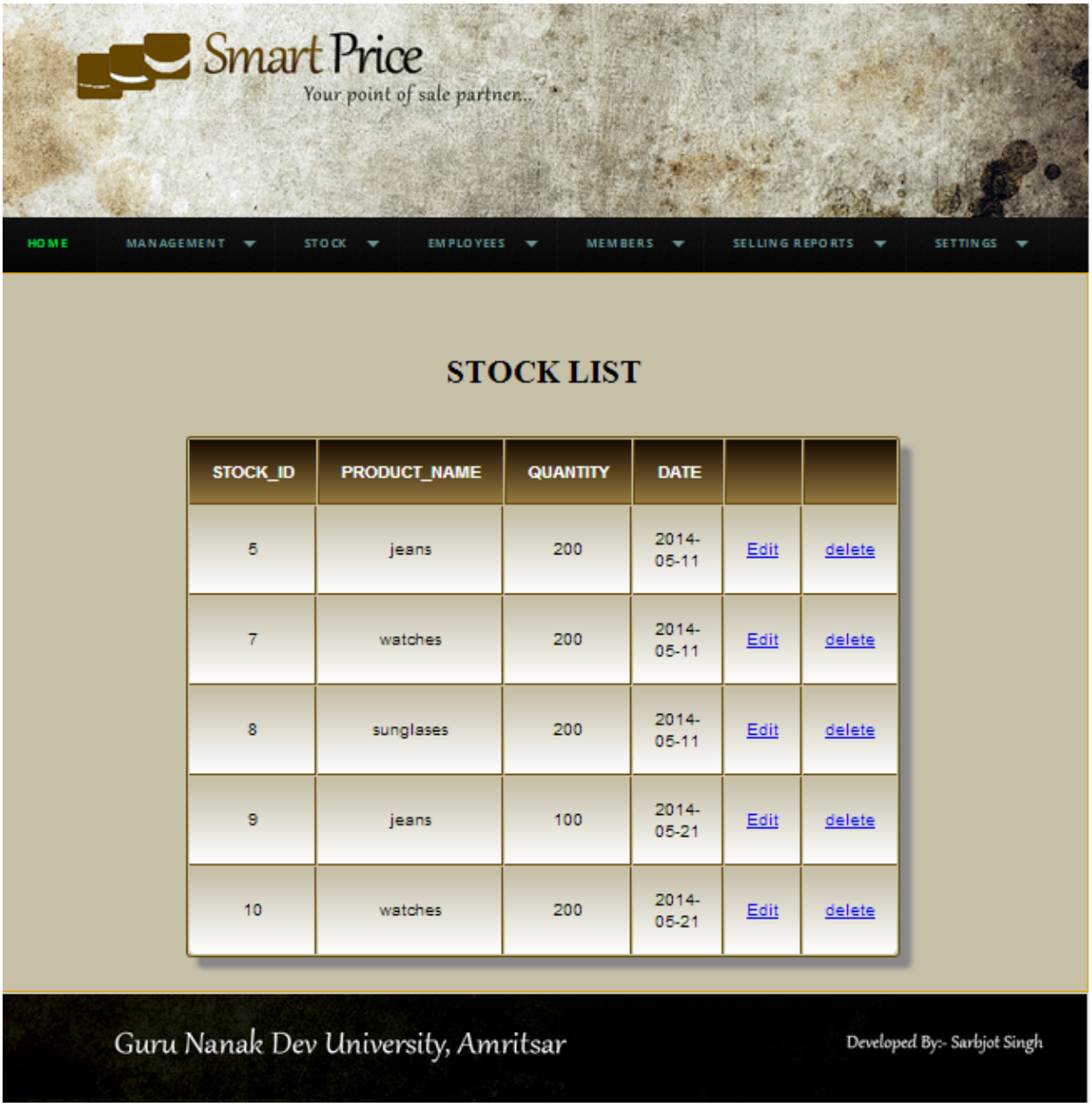
View Products



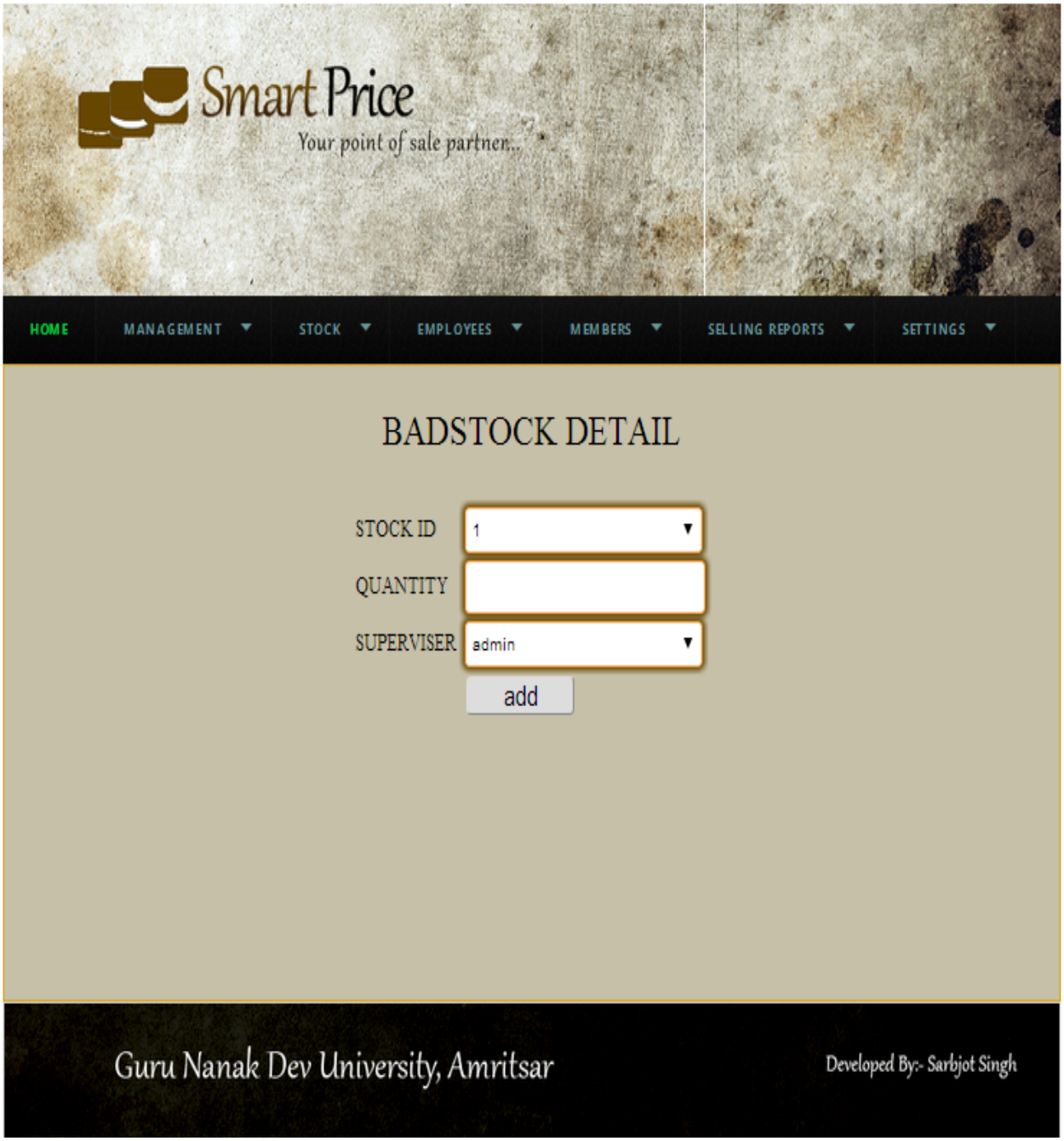
Add Stock


Smart Price
Your point of sale partnen...
HOME
MANAGEMENT
STOCK
EMPLOYEES
MEMBERS
SELLING REPORTS
SETTINGS
STOCK DETAIL
PRODUCT NAME
shirts
QUANTITY
DATE OF STOCk ADDING
dd-mm-yyyy
Submit
Guru Nanak Dev University, Amritsar
Developed By:- Sarbjot Singh

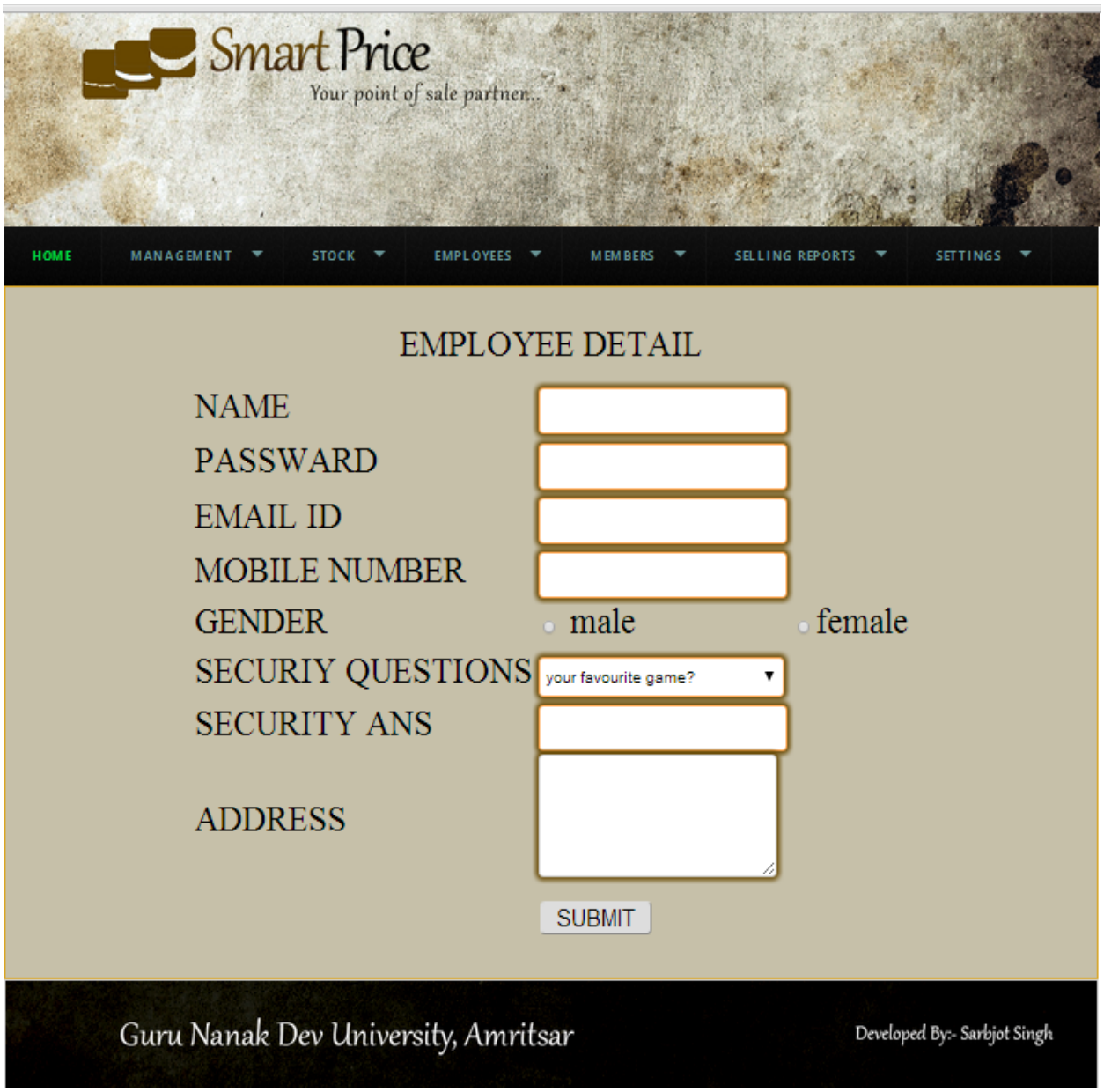
View Stock


Smart Price
Your point of sale partnen...
HOME
MANAGEMENT
STOCK
EMPLOYEES
MEMBERS
SELLING REPORTS
SETTINGS
STOCK LIST
STOCK_ID | PRODUCT_NAME | QUANTITY | DATE
5 | jeans | 200 | 2014-05-11 | Edit | delete
7 | watches | 200 | 2014-05-11 | Edit | delete
8 | sunglases | 200 | 2014-05-11 | Edit | delete
9 | jeans | 100 | 2014-05-21 | Edit | delete
10 | watches | 200 | 2014-05-21 | Edit | delete
Guru Nanak Dev University, Amritsar
Developed By:- Sarbjot Singh

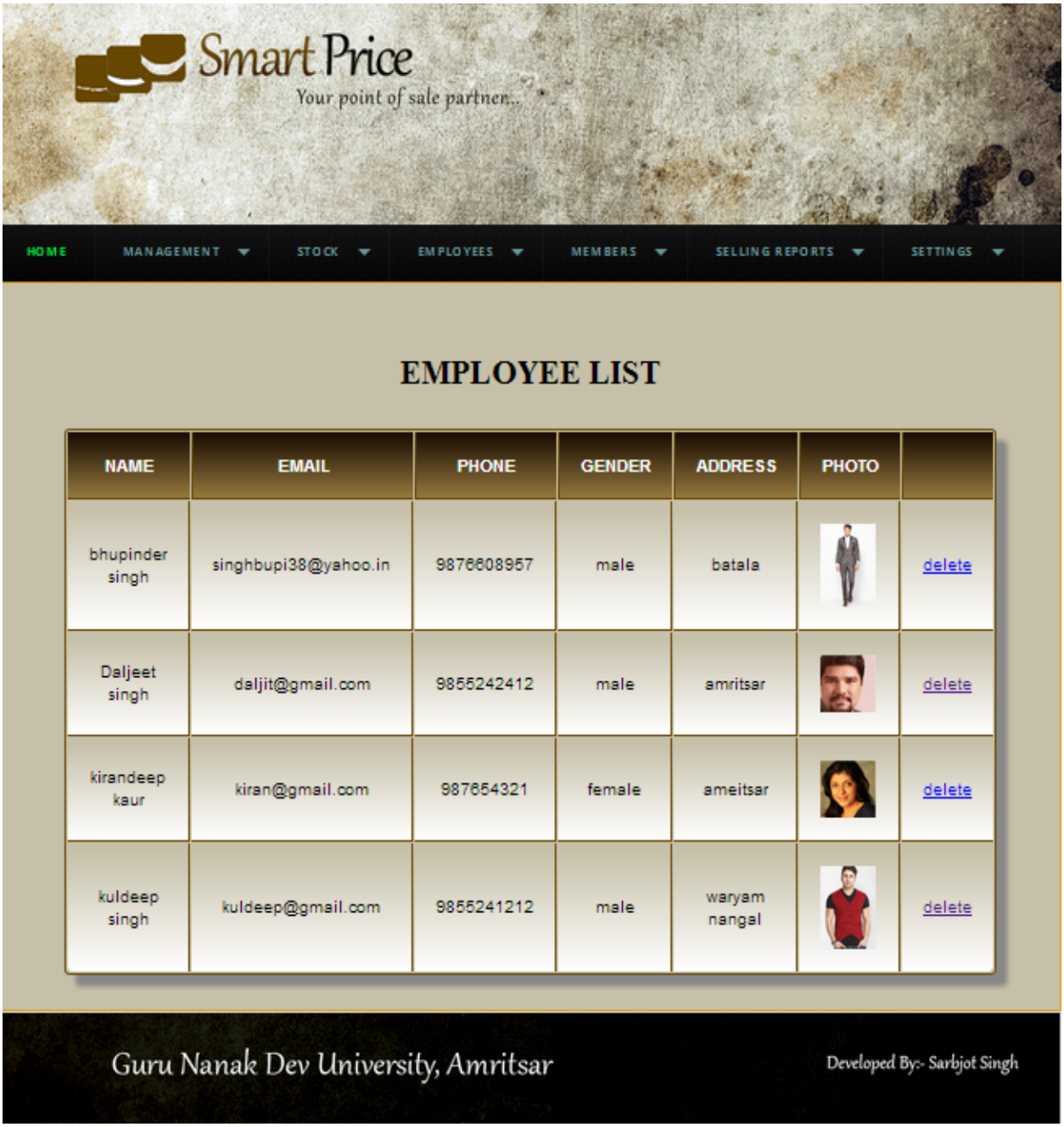
Add Badstock



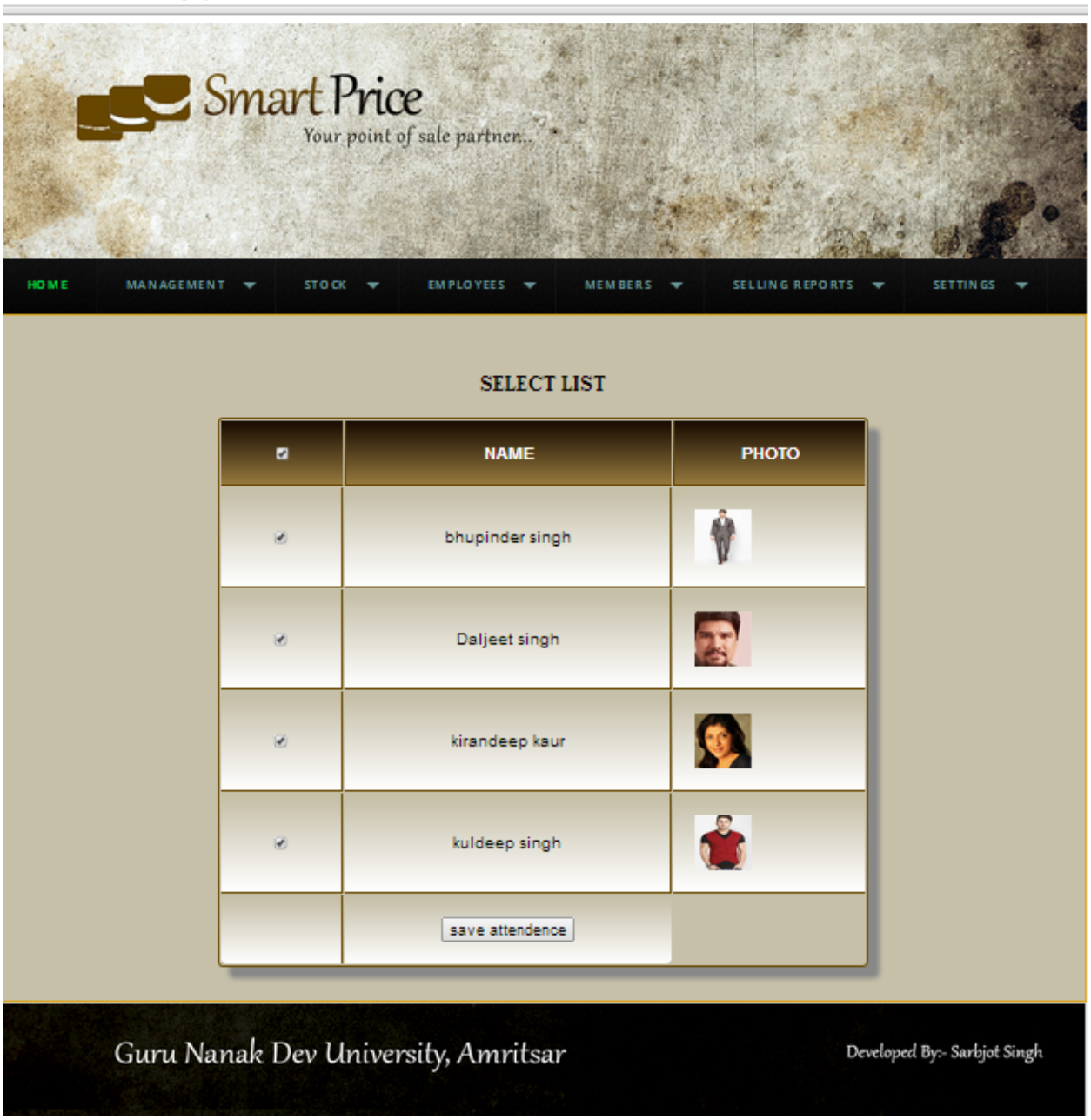
Add Employee



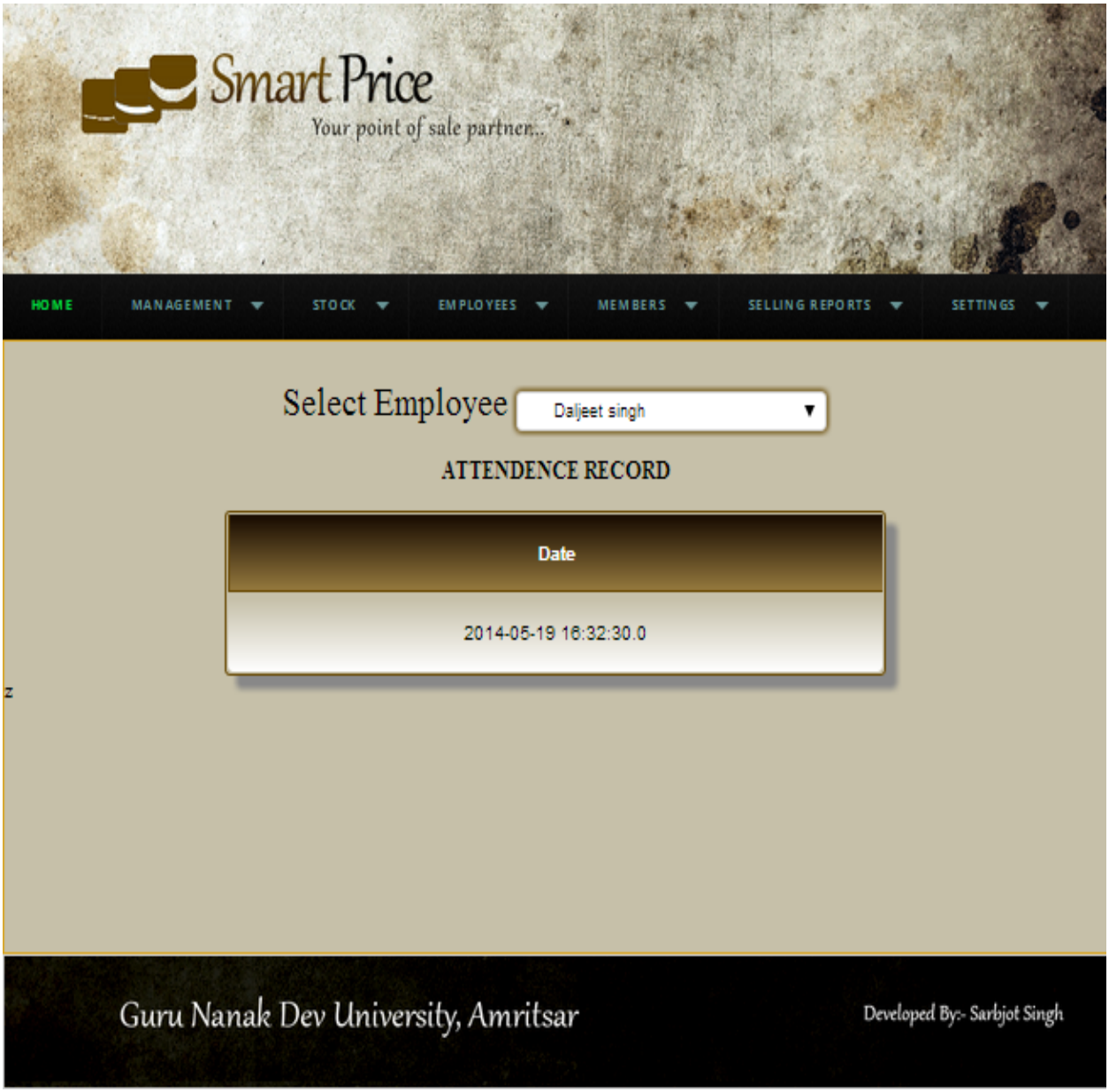
View Employee



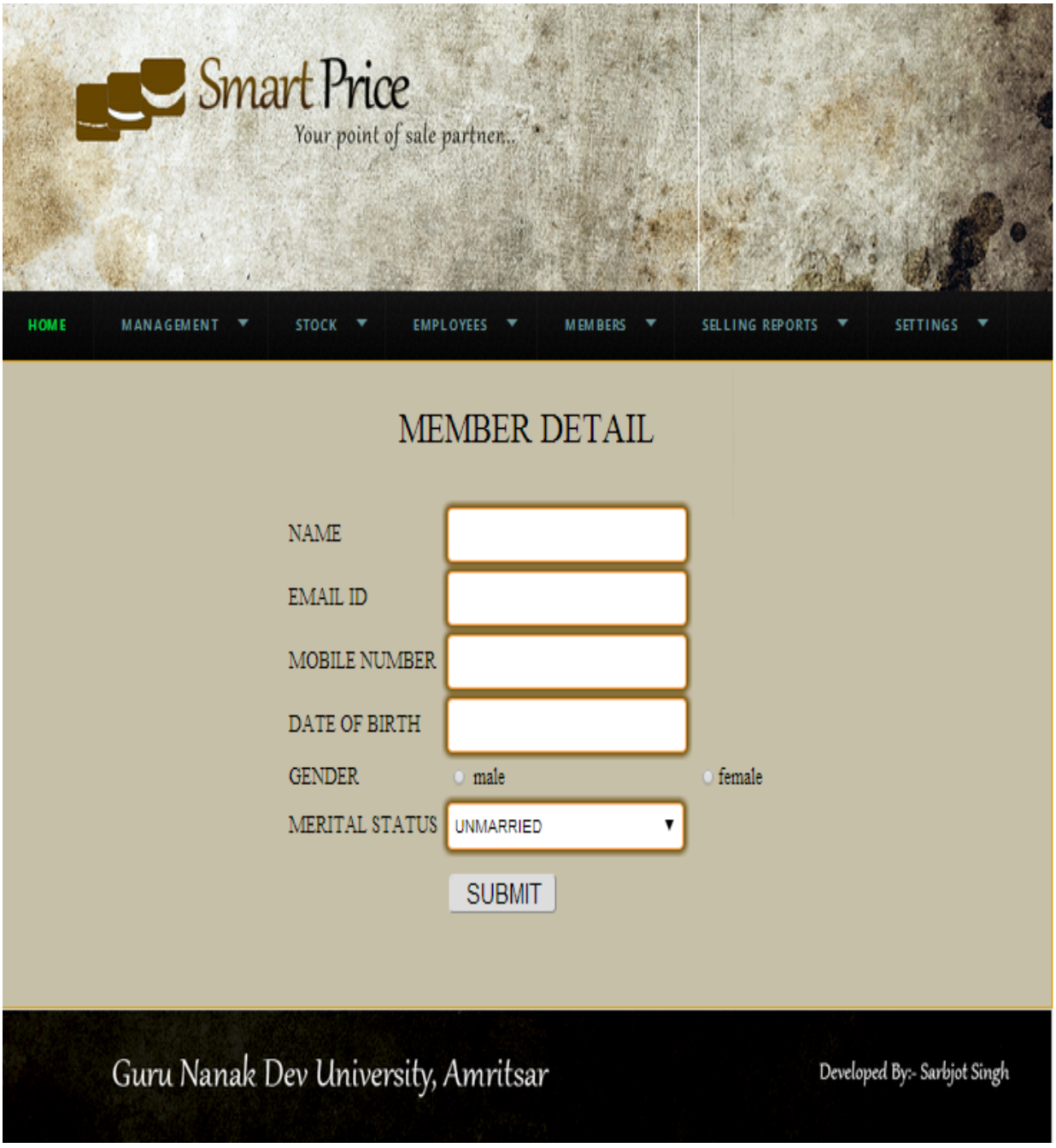
Employee Attendance



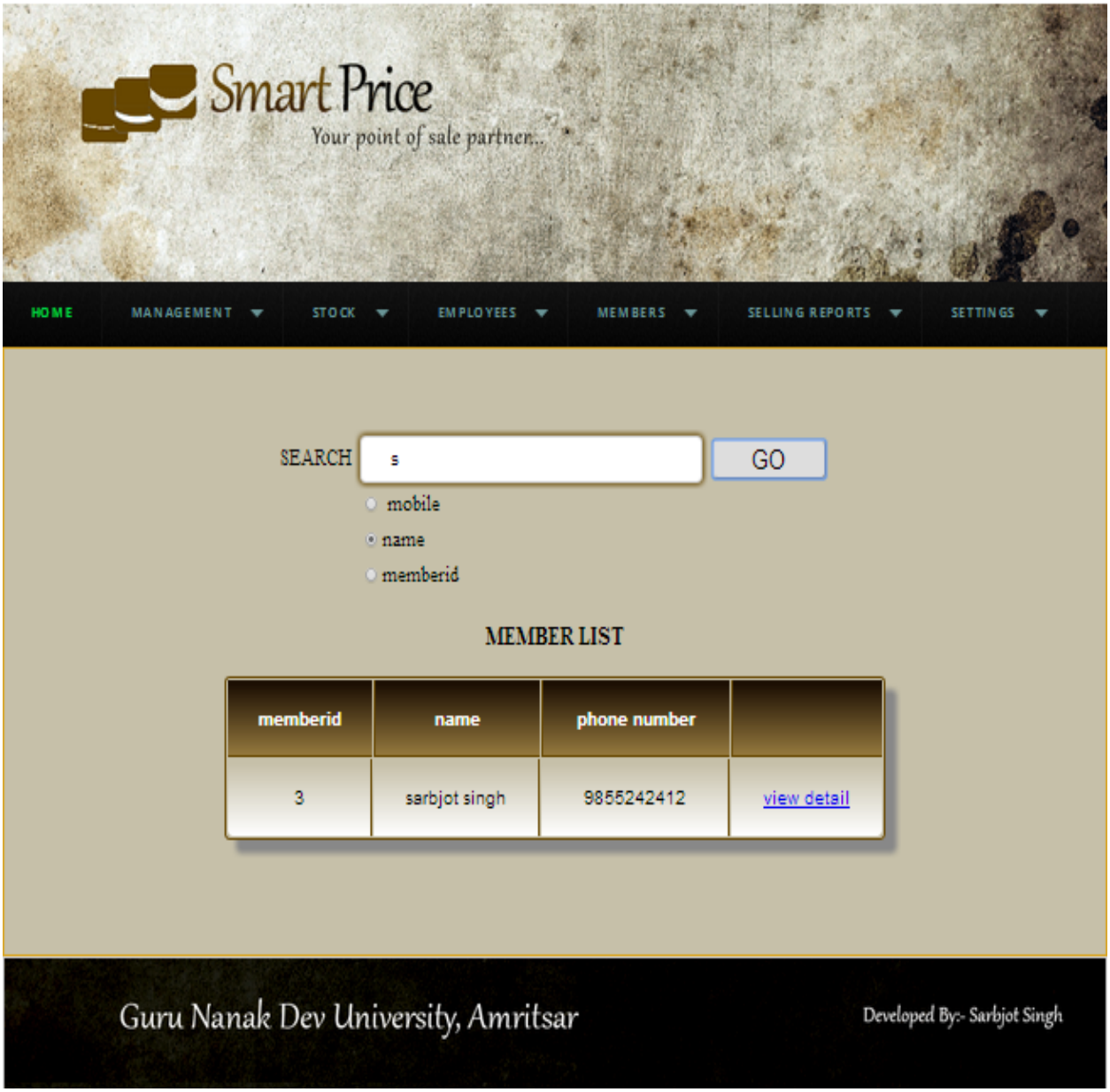
View Attendance



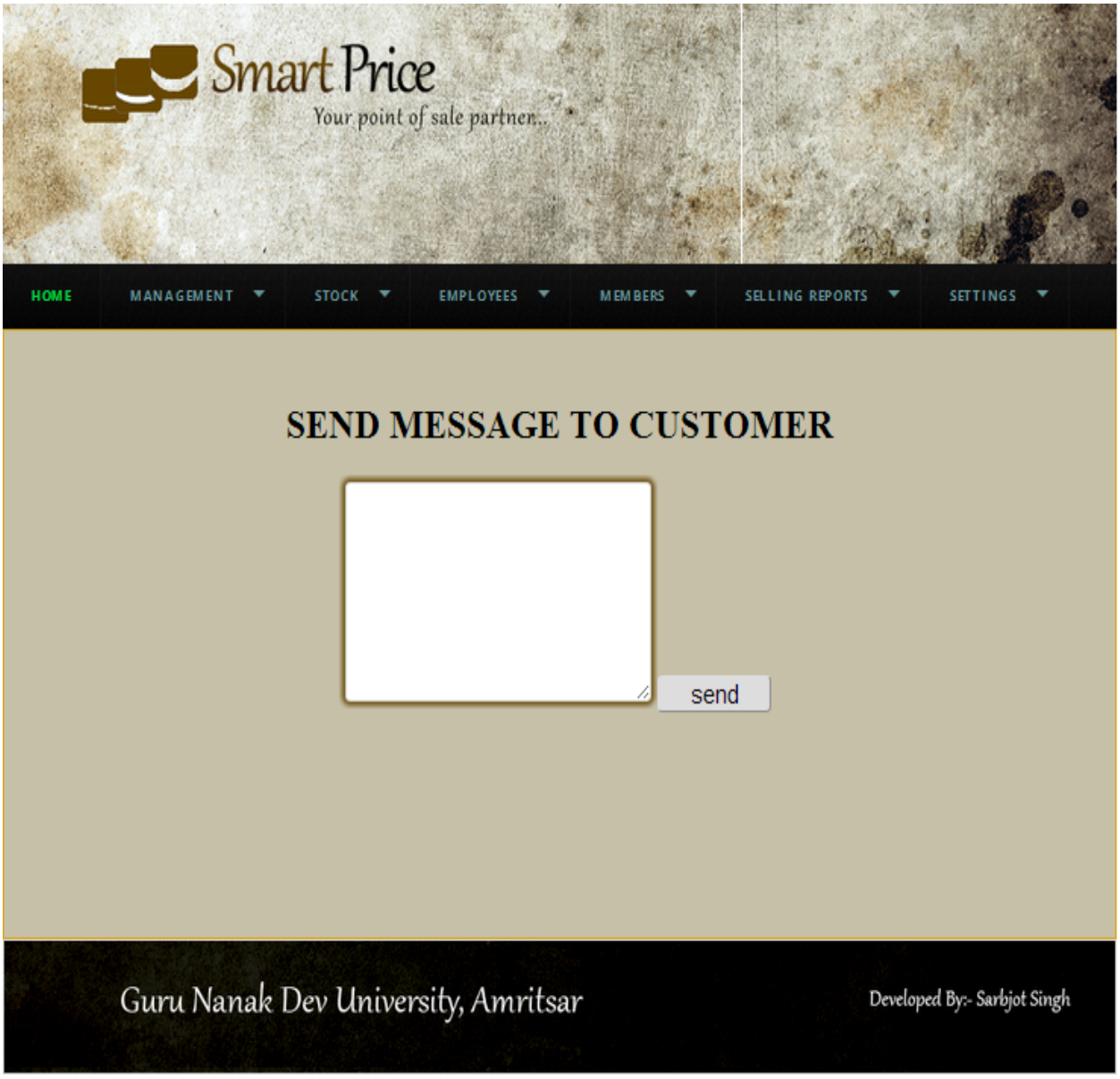
Add Member


Smart Price
Your point of sale partner...
HOME
MANAGEMENT
STOCK
EMPLOYEES
MEMBERS
SELLING REPORTS
SETTINGS
MEMBER DETAIL
NAME
EMAIL ID
MOBILE NUMBER
DATE OF BIRTH
GENDER
male
female
MERITAL STATUS
UNMARRIED
SUBMIT
Guru Nanak Dev University, Amritsar
Developed By:- Sarbjot Singh

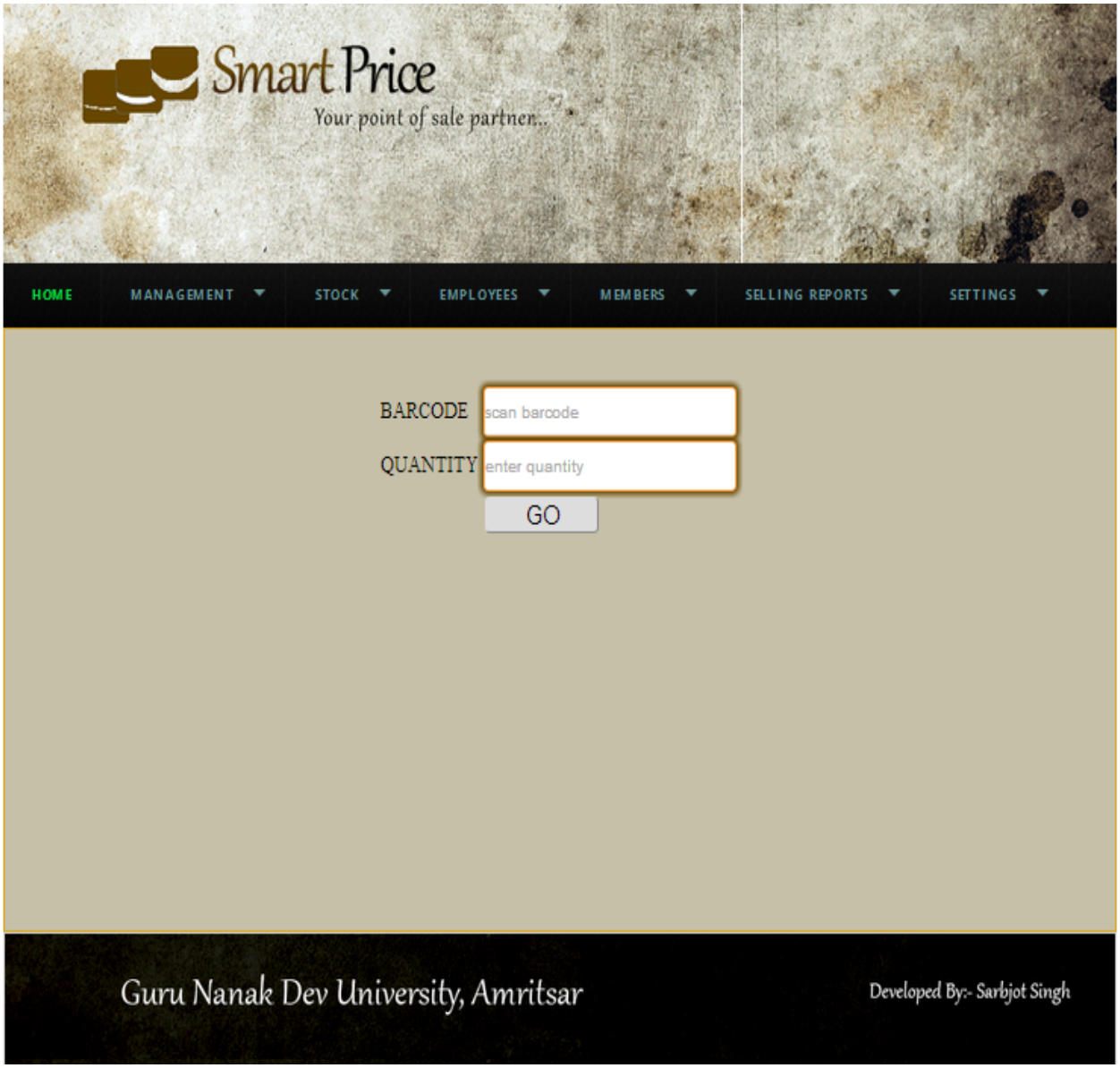
View Member


Smart Price
Your point of sale partner...
HOME
MANAGEMENT
STOCK
EMPLOYEES
MEMBERS
SELLING REPORTS
SETTINGS
SEARCH
s
GO
mobile
name
memberid
MEMBER LIST
memberid
name
phone number
3
sarbjot singh
9855242412
view detail
Guru Nanak Dev University, Amritsar
Developed By:- Sarbjot Singh

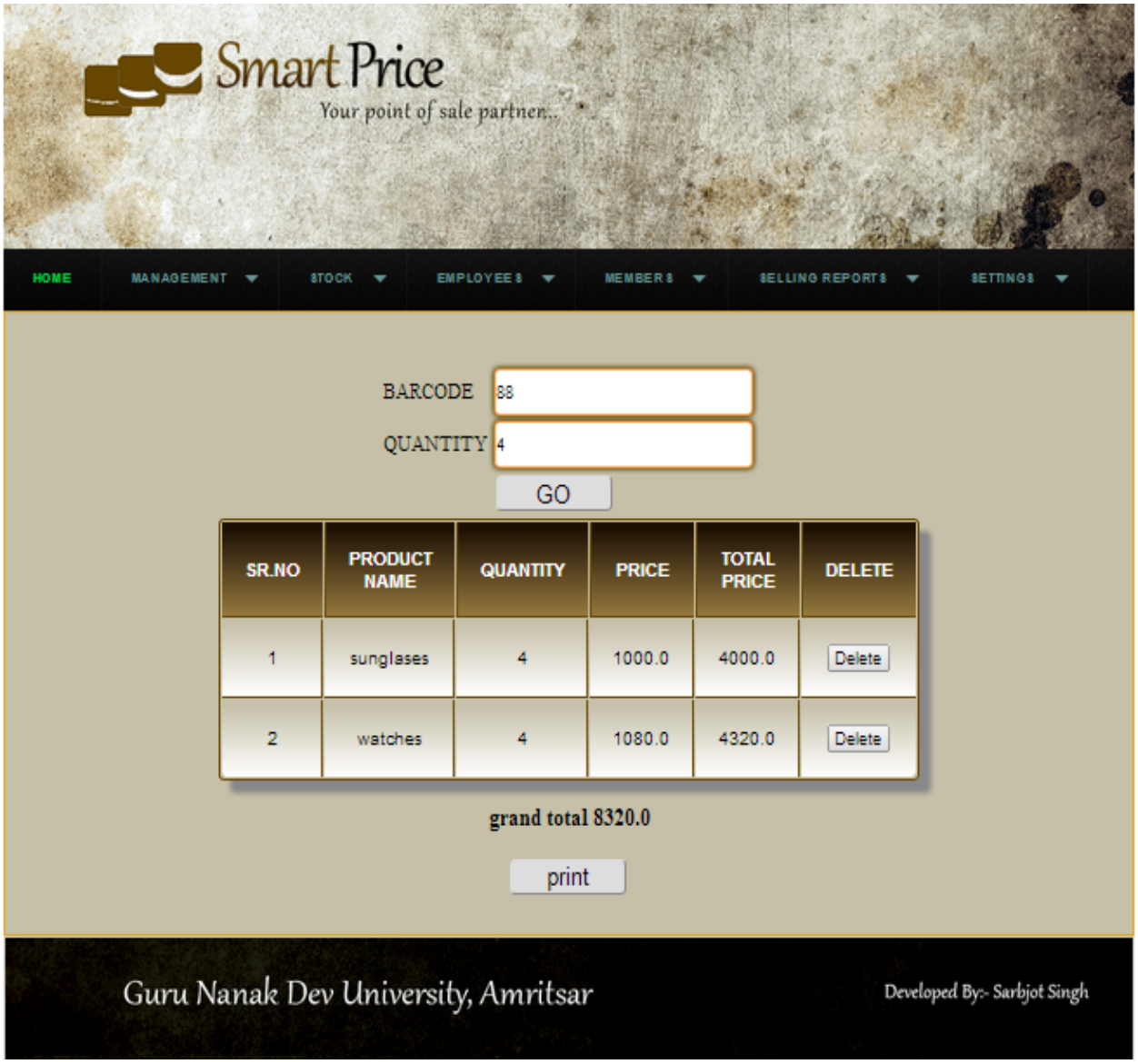
Send Messages



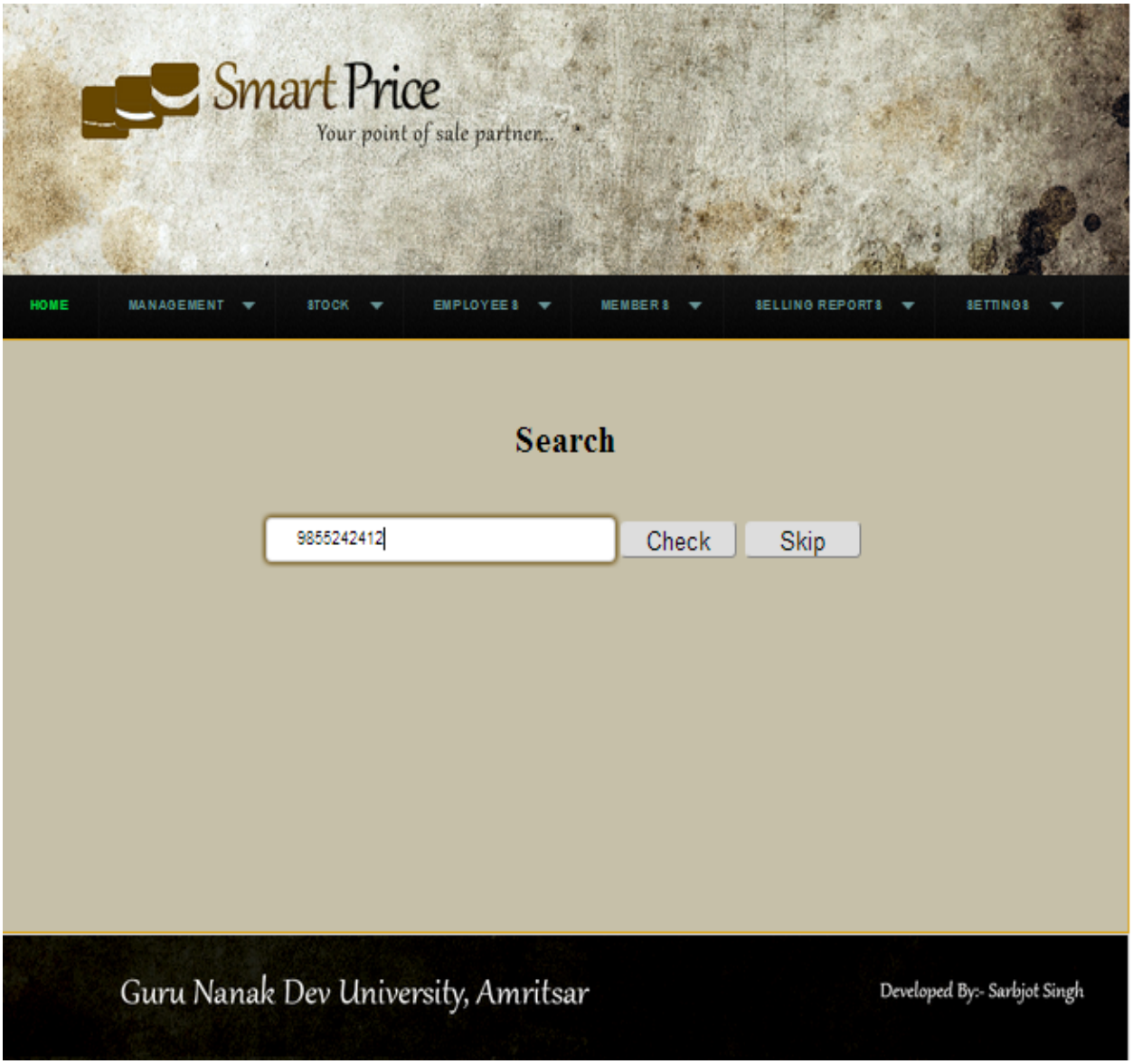
Barcode Scanner



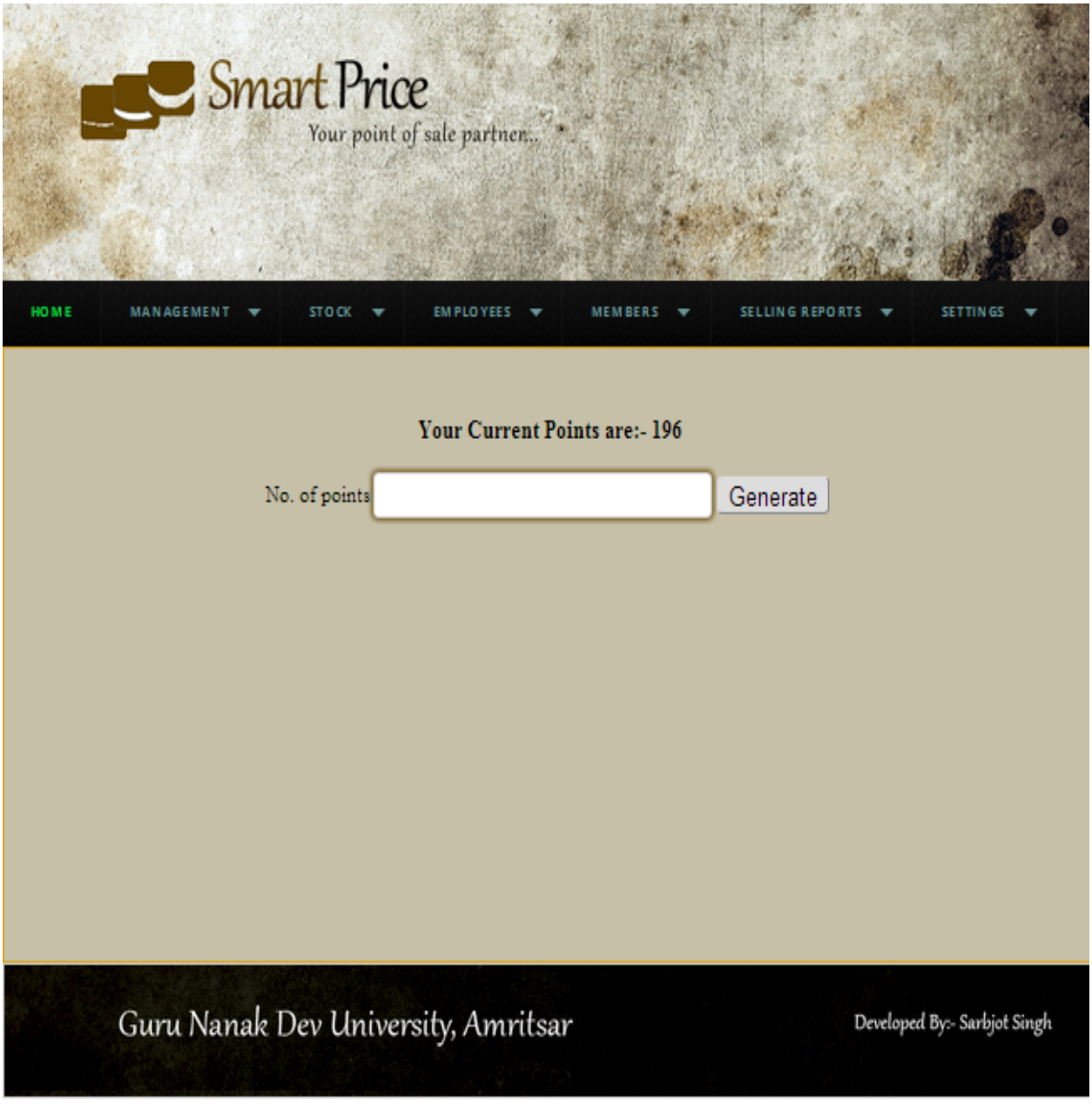
Generate Bill



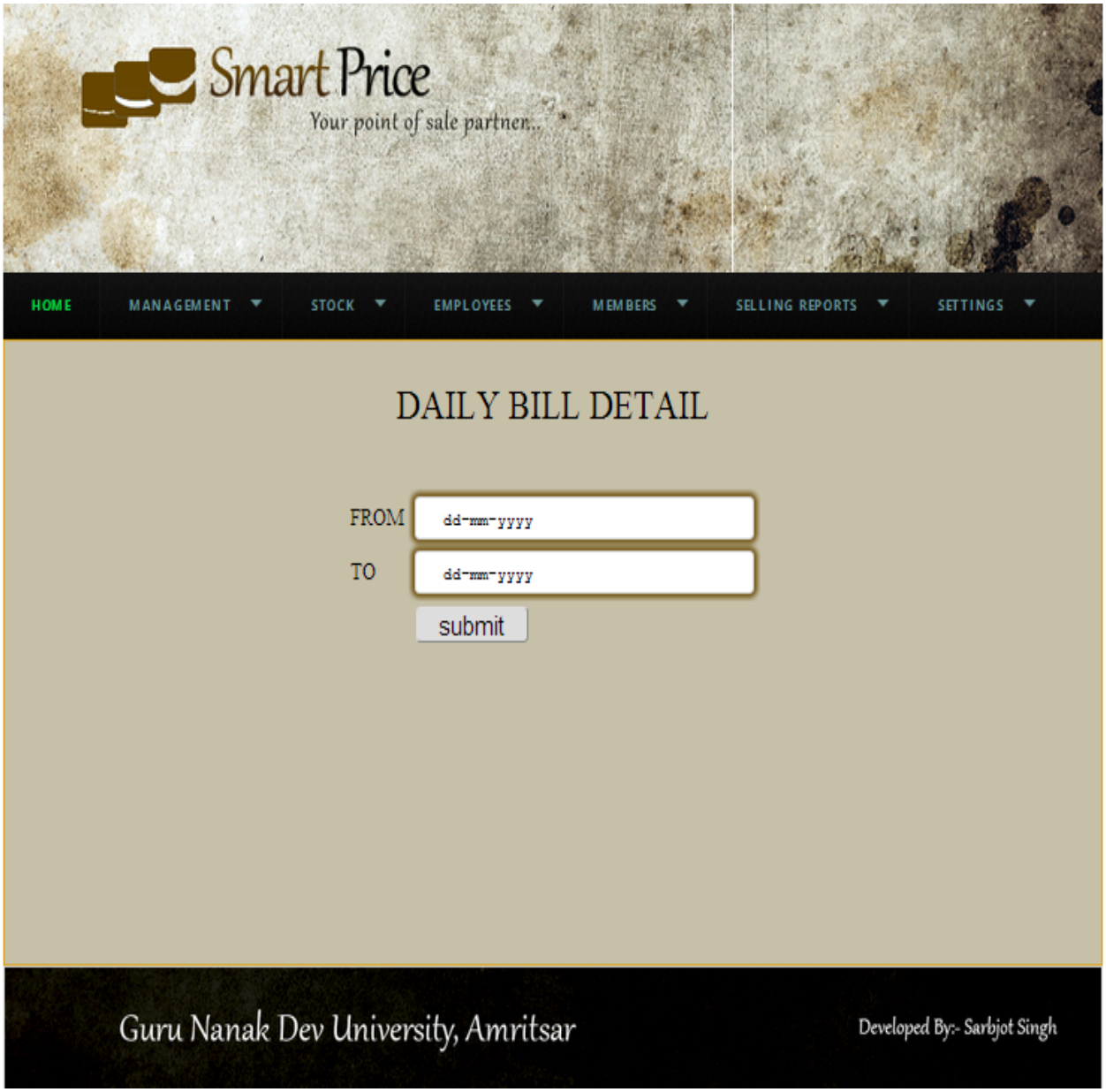
Add Points


Smart Price
Your point of sale partnen...
HOME
MANAGEMENT
STOCK
EMPLOYEES
MEMBERS
SELLING REPORTS
SETTINGS
Search
9855242412
Check
Skip
Guru Nanak Dev University, Amritsar
Developed By:- Sarbjot Singh

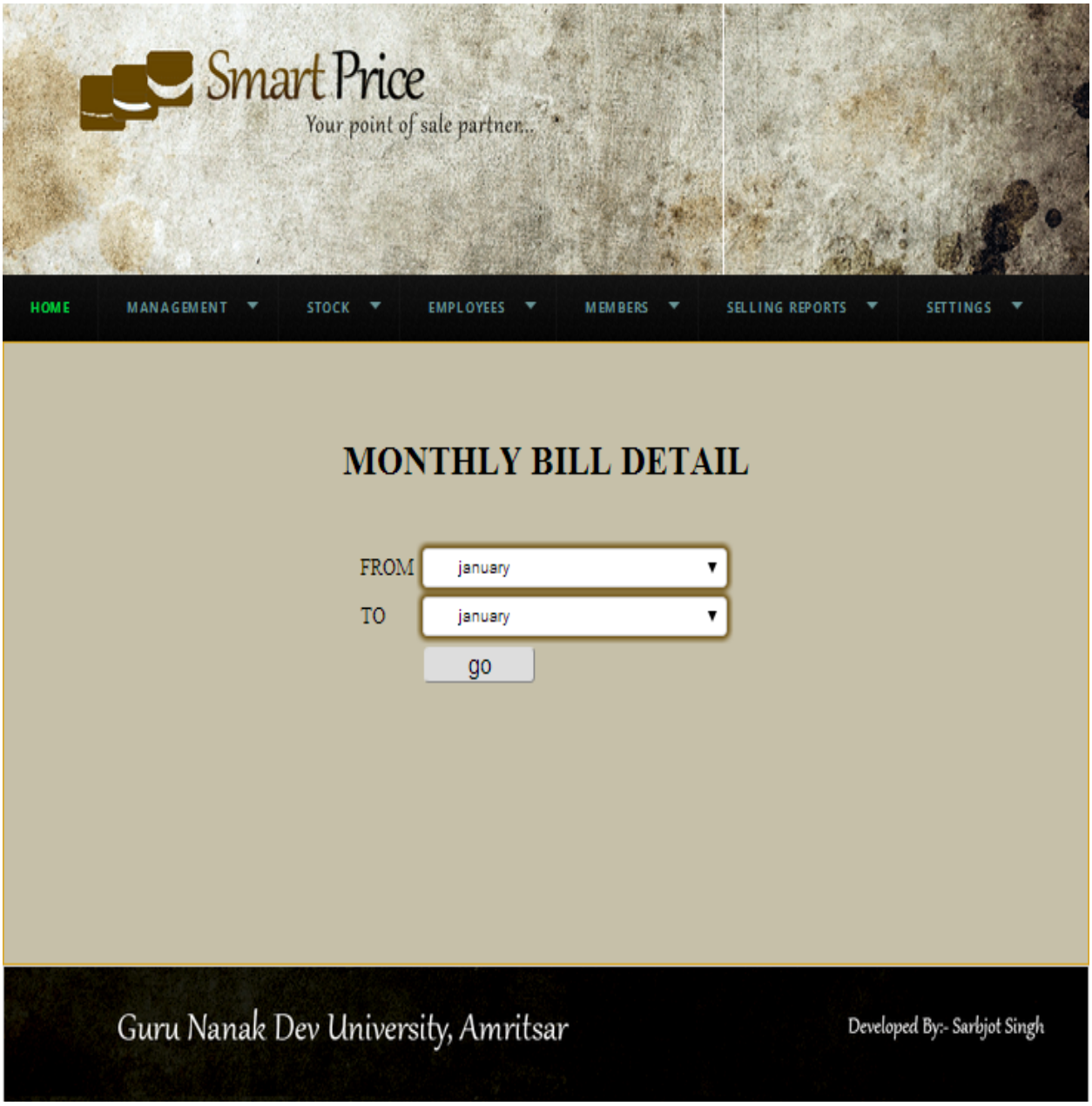
Check Points



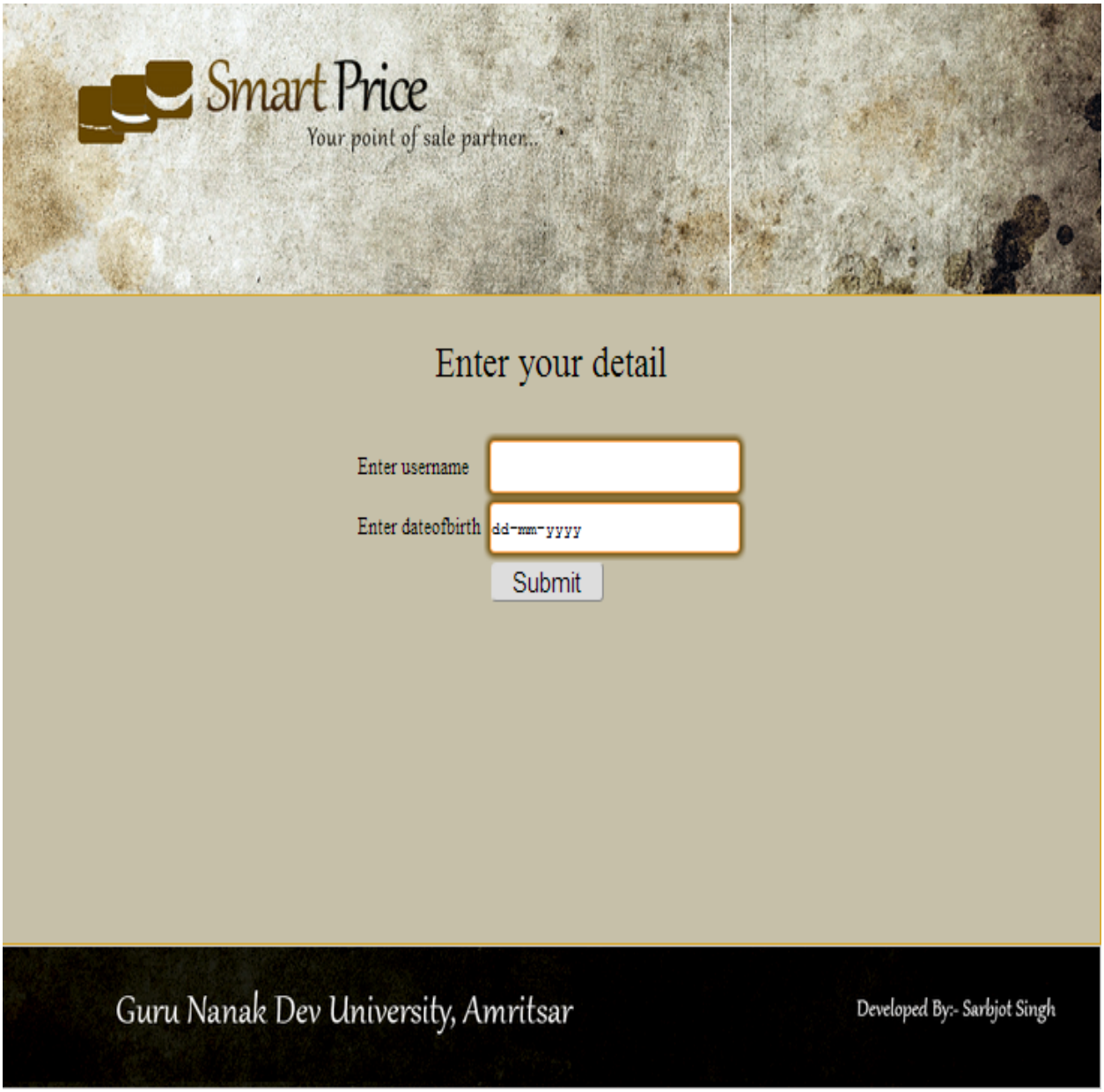
Daily Bill Report



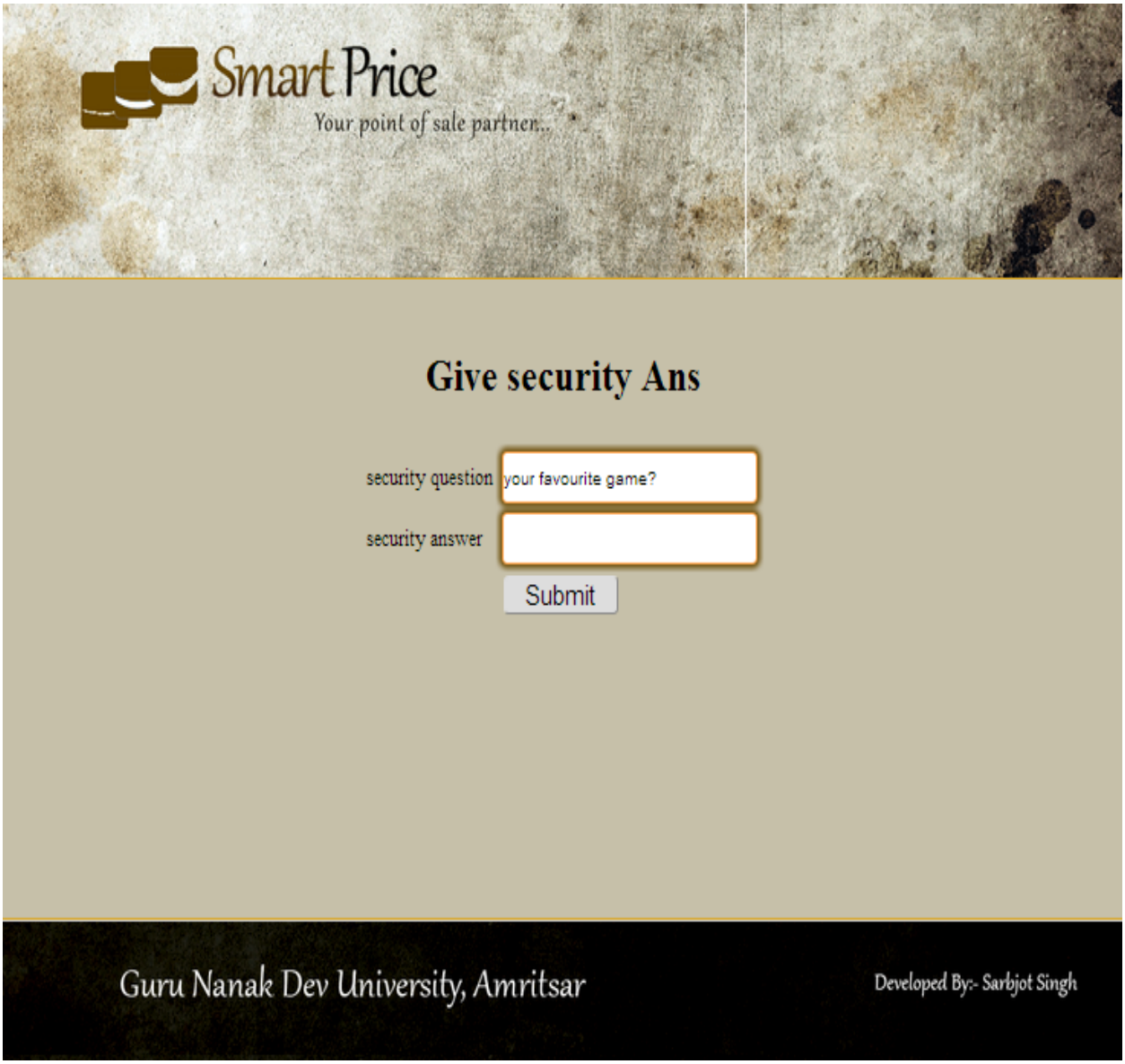
Monthly Bill Report



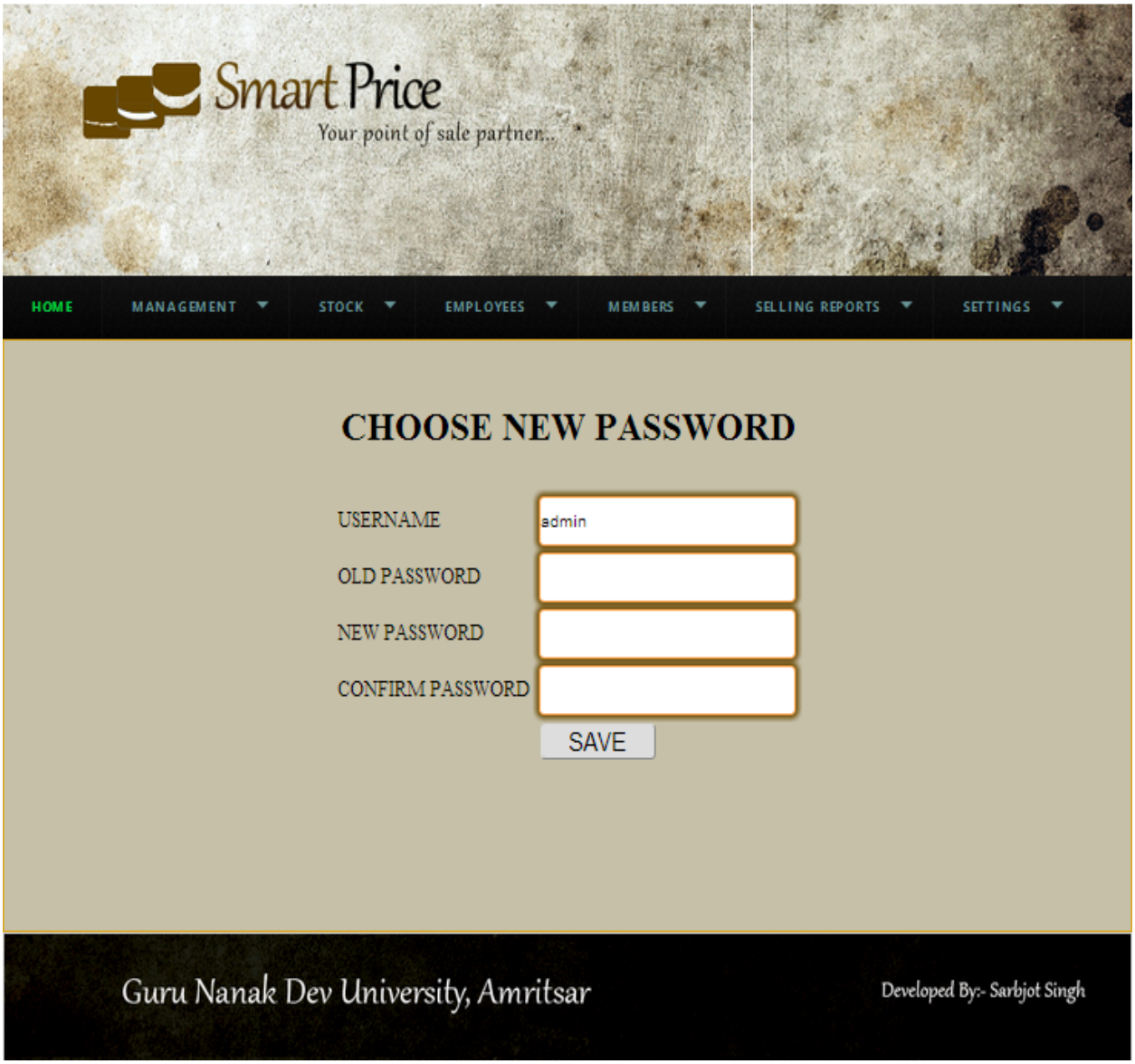
Forget Password



Security Answer



Change Password



Testing Phase

The basic goal of the software development process is to produce software that has no errors or very few errors. In an effort to detect errors soon after they are introduced, each phase ends with verification activity such as a review.

As testing is the last phase before the final software is delivered, it has the enormous responsibility of detecting any type of error that may in the software. Any software typically undergoes changes even after it has been delivered. And to validate that a change has not affected some old functionality of software regression testing is performed

Levels of Testing:

The basic levels of testing are unit testing, integration testing and system and acceptance testing. These different levels of testing attempt to detect different types of faults.

Code/Unit Testing:

Code testing and implementation is a critical process that can even consume more than sixty percent of the development time.

Testing:

The system development life cycle involves the phases of testing and debugging after the requirement analysis, designing and coding. The project in question was tested , debugged and implemented successfully.

Two strategies of software testing adopted for the new system are as follows:

1. Code testing
2. Specification testing
3. Unit Testing

Code testing:

Code testing was carried out to see the correctness of the logic involved and the correctness of the modules. Tests were conducted based upon sample and live data as well. All the modules are checked separately for assuming correctness and accuracy in all the calculations.

Specification testing:

It examines the specification stating about what program should do and how it performs under various conditions. This testing strategy is better strategy since it focuses on the way the software is expected to work.

Unit Testing:

During the phase of unit testing different constituent modules were testing against the specifications produced during the design for the modules. Unit testing is essentially for the verification of the code produced during the coding the phase, and goal is to test the internal logic of the modules. The modules once tested were then considered for integration and use by others.

Overview of Testing

1. **Testing:** Testing involves executing the program (or part of it) using sample data and inferring from the output whether the software performs correctly or not. This can be done either during module development (unit testing) or when several modules are combined (system testing).
2. **Defect Testing:** Defect testing is testing for situation where the program does not meet its functional specification. Performance testing tests a system's performance or reliability under realistic loads. This may go some way to ensuring that the program meets its non-functional requirements.
3. **Debugging:** Debugging is a cycle of detection, location, repair and test. Debugging is a hypothesis testing process. When a bug is detected, the tester must form a hypothesis about the cause and location of the bug. Further examination of the execution of the program (possible including many returns of it) will usually take place to confirm the hypothesis. If the hypothesis is demonstrated to be incorrect, a new hypothesis must be formed. Debugging tools that show the state of the program are useful for this, but inserting print statements is often the only approach. Experienced debuggers use their knowledge of common and/or obscure bugs to facilitate the hypothesis testing process. After fixing a bug, the system must be reset to ensure that the fix has worked and that no other bugs have been introduced. This is called regression testing. In principle, all tests should be performed again but this is often too expensive to do.

Test Planning

Testing needs planning, as it must be cost and time effective. Planning is setting out standards for tests. Test plans set out the context in which individual engineers can place their own work. Typical test plan contains:

1. Requirements trace ability (to ensure that all requirements are tested)
2. List of item to be tested
3. Schedule
4. Recording procedures so that test results can be audited
5. Hardware and software requirements
6. Constraints

Overview of Testing Strategies

Large systems are usually tested using a mixture of strategies. Different strategies may be needed for different parts of the system or stages of the process.

1. **Top-down testing:** This approach tests high levels of system before detailed components. This is appropriate when developing the system top-down and is likely to show up structural design errors early (and therefore cheaply). But this often has advantage that a limited, working system is available early on. Validation (as distinct from verification) can begin early. Its disadvantage is that stubs need to be generated (extra effort) and might be impracticable if component is complex (e.g. converting an array to a linked list; unrealistic to generate random list; therefore end up implementing unit anyway). Test output may be difficult to observe (needs creation of artificial environment). This is not appropriate for OO systems (except within a class).
2. **Bottom-up testing:** This is opposite of top-down testing. This tests low-level units then works up the hierarchy. Its advantages and disadvantages mirror those of top-down testing. In this testing there is need to write test

drivers for each unit. These are as reusable as the unit itself. Combining top-down development with bottom-up testing means that all parts of system must be implemented before testing can begin, which does not accord with incremental approach discussed above. Bottom-up testing is less likely to reveal architectural faults early on. However, bottom-up testing of critical low-level components is almost always necessary. Appropriate for OO systems.

3. **Stress testing:** Tests system's ability to cope with a specified load (e.g. transactions per second). Tests should be planned to increase load incrementally. This type of testing goes beyond design limit until system fails (this test is particularly important for distributed systems like checking degradation of performance a s network traffic increases).
4. **Back-to-back testing:** Comparison of test results from different versions of the system (e.g. compare with prototype, previous version or different configuration). Process – Run first system, saving test case results. Run second system, again saving its results. Compare result files. The key point to be noted is that 'no difference' does not mean there are no bugs. Both systems may have made the same mistake.
5. **Defect testing:** A successful defect test is a test that causes the system to behave incorrectly. Defect testing is not intended to show that a program meets its specification. If tests do not show up defects it may mean that the tests are not exhaustive enough. Exhaustive testing is not always practically applicable. Subset has to be defined (this should be part of the test plan, not left to the individual programmer).

Implementation

Implementation is the stage in the project where the theoretical design is turned into the working system and is giving confidence to the new system for the users

i.e. will work efficiently and effectively. It involves careful planning, investigation of the current system and its constraints on implementation, design of method to achieve the change over, an evaluation, of change over methods. A part from planning major task of preparing the implementation is education of users. The more complex system is implemented, the more involved will be the system analysis and design effort required just for implementation. An implementation coordinating committee based on policies of individual organization has been appointed. The implementation process begins with preparing a plan for the implementation for the system. According to this plan, the activities are to be carried out, discussions may regarding the equipment has to be acquired to implement the new system.

Implementation is the final and important phase. The most critical stage is in achieving a successful new system and in giving the users confidence that the new system will work and be effective. The system can be implemented only after thorough testing is done and if it found to working according to the specification. This method also offers the greatest security since the old system can take over if the errors are found or inability to handle certain types of transaction while using the new system.

The major elements of implementation plan are test plan, training plan, equipment installation plan, and a conversion plan.

Maintenance

Once the website is launched, it enters the maintenance phase. All systems need maintenance. Maintenance is required because there are often some residual errors remaining in the system that must be removed as they are discovered. Maintenance

involves understanding the effects of the change, making the changes to both the code and the documents, testing the new parts and retesting the old parts that were not changed. Maintenance is mainly of two types:

- **Corrective Maintenance:**

Almost all software that is developed has residual errors or bugs in them. Many of these surfaces only after the system have been in operation, sometimes for a long time. These errors once discovered need to be removed, leading to the software to be changed. This is called Corrective Maintenance.

- **Adaptive Maintenance:**

Even without bugs, software frequently undergoes change. The software often must be upgraded and enhanced to include more features and provide more services. This requires modification of the software. This type of maintenance is known as the Adaptive Maintenance

Bibliography

Websites

. **www.google.com**

. wikipedia

. www.w3school.com/html

. www.codeacademy.com

Books

- Complete reference by Herbert schildi
- Head first html with css and xhtml by elizabedth freeman