Chapter 1

Preamble

1.1Introduction

1.11 Database Management System (DBMS)

Following the Technology progress in the areas of processors, computer memory, computer storage, and computer networks, the sizes, capabilities, and performance of database and their respective dbms have grown in orders of magnitude. The development of database technology can be divided into three ears based on data model or structure: navigational/relational, and post-relational. The two main early navigational data models were the hierarchical model, epitomized by IBM'S IMS system and the coadsyl model, implemented in a number of products such as IDMS

The relational model employs set of ledger-style tables, each used for a different type
Of entity. Only in the mid-1980 did computing hardware become powerful enough to allow the wide deployment of relational system.by the early 1990 however, relational system dominated in all large-scale data processing applications, and as of 2015 they remain dominant: IBM DB2, ORACLE, MYSQL, and Microsoft sql server are the top dbms. The dominant database languages standardized sql for the relational model, has influenced database languages foe other data models.

1.1.2 My Sql

MySQL is and open-source relational database management system(rdms).
MySQL is written in C and C++. Its SQL parser is written in yacc, but it uses a home-brewed lexicalanalyzer.MySQLworksonmany systemplatforms,including AIX, BSDi, FreeBSD, HPUX, eComStation, i5/OS, IRIX, Linux, macOS, MicrosoftWindows, NetBSD, NovelNetWare, OpenBSD, OpenSolaris, OS/2 Warp, QNX, OracleSolaris, Symbian, Sun OS, SCO OpenServer, SCO UnixWare, Sanos and Tru64. A port of MySQL to OpenVMS also exists.

MySQL was originally developed to handle large databases quickly. Although MySQL is typically installed on only one machine, it is able to send the database to multiple locations, as users are able to access it via different MySQL client interfaces. These interfaces send SQL statements to the server and then display the results.

MySQL enables data to be stored and accessed across multiple storage engines, including InnoDB, CSV, and NDB. MySQL is also capable of replicating data and partitioning tables for better performance and durability. MySQL users aren't required to learn new commands; they can access their data using standard SQL commands.

For security, MySQL uses an access privilege and encrypted password system that enables host-based verification. MySQL clients can connect to MySQL Server using several protocols, including TCP/IP sockets on any platform. MySQL also supports a number of client and utility programs, command-line programs and administration tools such as MySQL Workbench.

1.1.3 PHP

PHP: Hypertext Pre-processor (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 Pre-processor.

PHP code may be executed with a command line interface (CLI), embedded into HTML code, or 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 outputs the results of the interpreted and executed PHP code, which may be any type of data, such as generated HTML code or binary image data. PHP can be used for many programming tasks outside of the web context, such as standalone graphical applications[9] 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.

1.1.4 Normalization

Normalization is a process of organizing the data in database to avoid data redundancy, Insertion anomaly, update anomaly and deletion anomaly .to overcome these anomalies we need to normalize the data. There are 4 basic types of normalizations. They are

- First normal form(1nf)
- Second normal form(2nf)
- Third normal form(3nf)
- Boyce & Codd normal form(bcnf)

First normal form(1nf) is defined as per rule as: an attribute (column) of a table cannot hold multiple values .it should hold only atomic values. This means that there shouldn't be repetition of data in the tables.

A table is said to be in 2nf if the two conditions stated are satisfied. the table is in first normal form and the entire non-prime attribute are dependent on the proper subset of any candidate key of table. The attribute that is not part of any candidate key are known as non-prime attribute.

A table design is said to be in 3nf if the table is in 2nf and transitive functional dependency of non -prime attribute on any super key are removed.

Boyce Codd normal form (bcnf) is the advance version of 3nf that's why it is also referred as 3.5nf, bcnf is stricter than 3nf.a table complies with bcnf if it is in 3nf
And for every functional dependency x->y, x should be the super key of the table.

1.1.5 PROPOSED SYSTEM

The proposed Home Management System provides various services like storing of data regarding various cases of expenditures. We can also retrieve the data and view reports according to our requirements that are; we can view daily reports, monthly reports, yearly reports etc. It is easy to access the application through the simple interactive website where the users can register themselves and gain access to the storage of database. And also the data is secured as the system has user authentication facility where only registered users can retrieve the desired data by requesting the application server to the access of the database.

EXISTING SYSTEM

The existing System includes not much technological user friendly application to maintain day-to-day expenses. One can easily lose track of their expenditures and it's difficult to maintain these accounts and reviews them as we need. There might be certain mobile applications fulfilling the need but they prove to be less secured and quick accessible.

1.1Objectives

In this system we aim in providing the clients personalized views of the data required, by designing the efficient user interface. The users can add there day- to- day expenses under specific tables and each of the records are secured and accessible to them whenever they need.

The common everyday expenses include buying of groceries, medicines or payment of variety of bills etc. User can log into the website and store the data along with respective details like date, amount spent etc. so that they don’t lose track of any calculations in their account maintenance.

Chapter 2

Requirement Specification

2.1 SOFTWARE SPECIFICATION

- Design and Interface: HTML, CSS
- Programming language: PHP
- Database: MySQL Server
- XAMPP Server control Panel v-3.2.4

2.2 HARDWARE REQUIREMENT

- Operating System: Windows XP, 7 OR 8 OR Windows10
- Processor: Intel Core Duo 2.0 GHz or more
- RAM: 1 GB or more
- Hard Disc: 80 GB or more
- Monitor: 15 inches CRT or LCD Monitor
- Keyboard: Normal or multimedia keyboard
- Mouse: Compatible Mouse

2.3 USER CHARACTERISTICS

Every user:

Should be comfortable with basic working of the computer

Must have basic knowledge of English

Must carry a User-Name and password used for authentication

Chapter 3

System Design and Implementation

3.1 Introduction

Systems design is the process or art of defining the architecture, components, modules, interface and data for a system to satisfy specified requirements. Once could see it as the application of system theory to product development.

This project is implemented using php, which is proven to be a very efficient scripting language for web development. Interface to program is provided with the help of MySQL database

3.2 ER Diagram

An Entity–relationship model (ER model) describes the structure of a database with the help of a diagram, which is known as Entity Relationship Diagram (ER Diagram)**.** An ER model is a design or blueprint of a database that can later be implemented as a database. The main components of E-R model are: entity set and relationship set.

An entity-relationship model or the ER Diagram describes inter-related things of interest in a specific domain of knowledge, An ER model is composed of entity types and specifies relationships that can exist between instances of those entity types.

IN terms of DBMS, an entity is a table or attribute of a table in database, so by showing relationship among tables and their attributes, ER diagram shows thecomplete logical structure of a database.

3.3 Schema diagram

The schema diagram of a database system is its structure described in a formal language supported by the database management system (DBMS).the formal definition of a database schema is a set of formulas called integrity constraints imposed on a database.

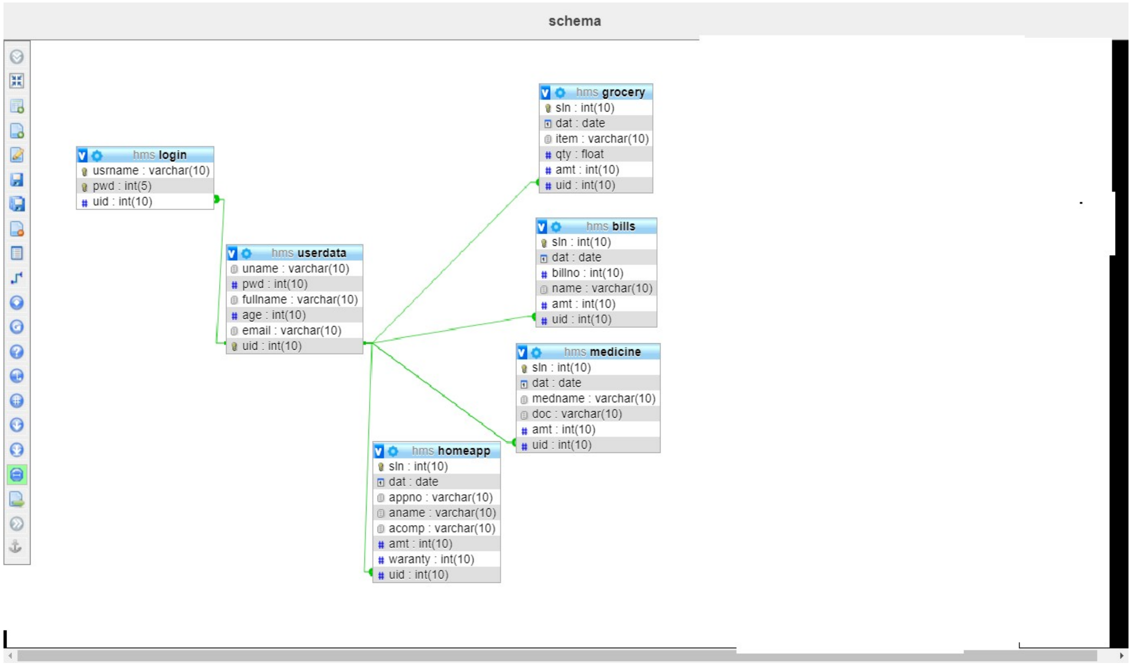


Figure 3.2 schema is defined for Homes management system. All the various table used are described in the following schema. The necessary Primary Key's and the corresponding Foreign keys are also represented.

The term "schema" refers to the organization of data as a blueprint of how the database is constructed. these integrity constraints ensure compatibility between parts of the schema. All constraints are expressible in the same language. A database can be considered a structure in realized of the database language. The states of a created conceptual schema are transformed into an explicit mapping, the database schema. This describes how real-world entities are modelled in the database.

3.4 Queries

The below mentioned are all the queries used to perform various tasks in MySQL such as insert, delete, update. A short description of the query is also provided.

Query: create table userdata (
```
uname  varchar(10),
 pwd int,
fullname varchar(10),
age int,
email varchar(10),
uid varchar(10)   primary key);
```

Description: This query is used to create a table called userdata which will store all the user details of all registered users. This table is identified by primary key (uid).

Query: create table login(
```
uname references userdata(uname) on delete cascade,
pwd references userdata(pwd) on delete cascade,
uid references userdata(uid) on delete cascade,
primary key(uname,pwd);
```

Description: This query is used to create a table called login which will store all the users login credentials referring from userdata table.

Query: create table grocery(
```
sln int primary key auto increment,
dat date,
item varchar(10),
qty float,
amt int,
uid references userdata(uid) on delete cascade);
```

Description: This query is used to create a table called grocery which will store all the information pertaining to grocery details entered by user. This table is identified with the help of primary key(sln)which is also auto increemented.

Query: : create table bills(
sln int primary key auto increment,
dat date,
billno int,
name varchar(10),
amt int,
uid references userdata(uid) on delete cascade);

Description: This query is used to create a table called bills which will store all the information pertaining to bills payed by user. This table is identified with the help of primary key (sln)which is autoincreemented. The uid is the foreign key.

Query: : create table medicine(
sln int primary key auto increment,
dat date,
medname varchar(10),
doc varchar(10),
amt int,
uid references userdata(uid) on delete cascade);

Description: This query is used to create a table called medicine which will store all the information of medicines taken by user. This table is identified with the help of primary key(sln)which is auto incremented and uid is the foreign key.

Query: create table homeapp(
sln int primary key auto increment,
dat date,
appno int,
aname varchar(10),
acomp varchar(10),

warranty int,
amt int,
uid references userdata(uid) on delete cascade);

Description: This query is used to create a table called homeapp which will store all the information of home appliances taken by user. This table is identified with the help of primary key(sln)which is auto incremented and uid is the foreign key.

Query: : create table usercount(
dat date,
ct int primary key);

Description: This query is used to create a table called usercount which will store the number of users updated by a trigger. This table is identified with the help of primary key(ct).

Query: INSERT INTO `userdata`(`uname`, `pwd`, `fullname`, `age`, `email`, `uid`) VALUES (‘suni’,123,’Sunidhi’,20,’suni@’,’106’);

Description: This query is used to insert into table userdata as a new user. All the values to be inserted are mentioned in a chorological order and are inserted into the table according to the order.

Query: INSERT INTO `grocery`(`dat`, `item`, `qty`, `amt`, `uid`) VALUES (‘2019-12-12’,’fruits’,2.5,35,106);

Description: This query is used to insert into table grocery. All the values to be inserted are mentioned in a chorological order and are inserted into the table according to the order.sln is autoincreemented.

Query: INSERT INTO `bills`(`dat`, `billno`, `name`, `amt`, `uid`) VALUES (‘2019-12-12’,678,’gas’,350,106);

Description: This query is used to insert into table bills. All the values to be inserted are mentioned in a chorological order and are inserted into the table according to the order.sln is autoincreemented.

Query: INSERT INTO `medicine`(`dat`, `medname`, `doc`, `amt`, `uid`) VALUES (‘2019-12-12’,’nuro’,’karthik’,35,106);

Description: This query is used to insert into table medicine. All the values to be inserted are mentioned in a chorological order and are inserted into the table according to the order.sln is autoincreemented.

Query: INSERT INTO `homeapp`(`dat`, `appno`, `aname`, `acomp`,`amt`,`warranty`,`uid`) VALUES (‘2019-12-12’,890,’fridge’,’LG’,35000,8,106);

Description: This query is used to insert into table homeapp. All the values to be inserted are mentioned in a chorological order and are inserted into the table according to the order.sln is autoincreemented.

3.4.1 Stored Procedures

A stored procedure is nothing more than prepared sql code that the developer saves
So, the application can reuse the code over and over again. So, if the developer thinks
About a query that you write over and over again, intend of having to write a query
Each time you would save it as a stored procedure and then just call the stored procedure
To execute the sql code that you saved as a part of the stored procedure.

Query:

```
BEGIN
SELECT SUM(grocery.amt) as grocsum FROM grocery where grocery.uid= uids and
grocery.dat=CURRENT_DATE() ;
END
```

Description: this stored procedure is used to get the details of today's grocery summary of the user with uid.

Query:

```
BEGIN
SELECT SUM(bills.amt)  as billsum FROM bills where bills.uid= uids and
bills.dat=CURRENT_DATE();
END
```

Description: this stored procedure is used to get the details of today's bills summary of the user with uid.

Query:

```
BEGIN
SELECT SUM(medicine.amt) as medsum FROM medicine where medicine.uid= uids and
medicine.dat=CURRENT_DATE() ;
END
```

Description: this stored procedure is used to get the details of today's medicine summary of the user with uid.

Query:

```
BEGIN
BEGIN
SELECT SUM(homeapp.amt) as  homesum FROM homeapp where homeapp.uid= uids and homeapp.dat=CURRENT_DATE();
END
```

Description: this stored procedure is used to get the details of today’s homeapp summary of the user with uid.

3.4.2 Triggers

Triggers are the stored programs, which are automaticallyexecuted or fired when some Events occur. Triggers are stroed into database and invoked repeatedly, when specific Condition match.

Benefits of triggers

Triggers can be written for the following process

1.generating some derived columns and values automatically.

2.enforcing referential integrity.

3.event logging and storing information on table access.

4.auditing.

5.synchronous replications of tables.

6.imposing security authorizations.

7.preventing invalid transaction.

Query:

```
CREATE DEFINER=`root`@`localhost` TRIGGER `trig1` BEFORE INSERT ON
`userdata` FOR EACH ROW BEGIN UPDATE `usercount` SET `da`=CURRENT_DATE(),`
ct`=`ct`+1 WHERE `id`= 1; END
```

Description: The above trigger has been implemented for the userdata, which is automatically increments the count of users registerd when the new user has registered as and updates the usercount table in database.

3.5 Pseudo code

Pseudo code is an informal high-level description of the principle of a computer program or Other algorithm.it uses the structural conversational of a formal programminglanguage, butit is intended for human reading rather than machine reading.

3.5.1 Algorithm for Login

```
Step 1: BEGIN
Step 2: Enter username and password
Step 3: verify the credentials entered with that in the login table
Step 4: If credentials match, then proceed to the hello user page else show unsuccessfull.
Step 5: END
```

3.5.2 Algorithm for user options

```
Step 1: BEGIN
Step 2:if the ADD button is clicked,  go to ADD TO page
Step 3: if the VIEW button is clicked, go to VIEWS page
Step 4: if logout button is clicked, LOGOUT
Step 5: else come back to HOME page
Step 6: END
```

3.5.3 Algorithm for Insert

```
STEP 1: BEGIN
STEP 2: get all the necessary values required for insertion into textbox variables in html and POST this values through form action to a php page which establishes the connection to the database.
STEP 3: In PHP page after successful connection establishment with db insert into tables by executing  through sql queries which has user entered data.
STEP 4: execute the Query. If successful alert added! Else alert Error!!
STEP 5: END
```

3.5.3 Algorithm for views

STEP 1: BEGIN

STEP 2: click on required button to go to respective views.

STEP 3: if user clicks GROCERY,BILLS,MEDICINE,HOME APPLIANCES get repective user ids into uid and pass to php pages. Those php pages gives back results by selecting data.

STEP 4: if user clicks Today’s Summary,

CALL STORED PROCEDURES

call sumcal, call billcal, call medcal,call homecal.

STEP 5: if user clicks MONTHLY or BETWEEN, accept the user inputs and POST them to phps to access database and fetch required data.

STEP 6: END

3.5.5 Algorithm for New user Account

Step 1: BEGIN

Step 2: Clicks on NEW USER, go to CREATE ACCOUNT page.

Step 3: User should enter the required details with no fields empty and no constraints violated.

Step 4: Upon CREATE ACCOUNT , php page establishes connection with db to access the userdata table and also excecutes the **TRIGGER** to update user count details.

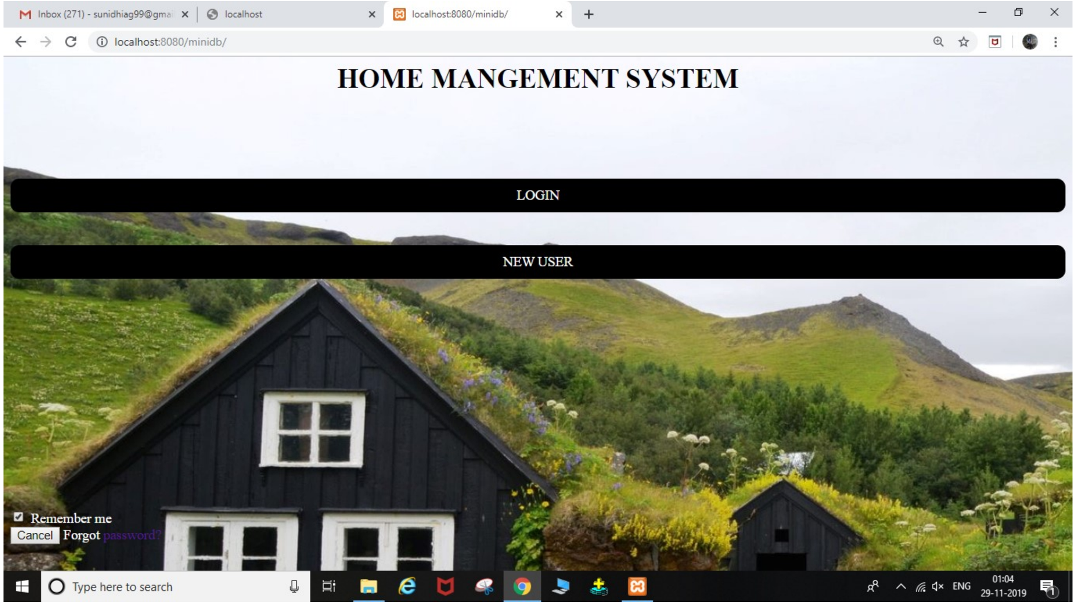
Step 5: If any violations and unsuccsessfull, alert the user saying Account cannot be created due to error.

Step 6: END

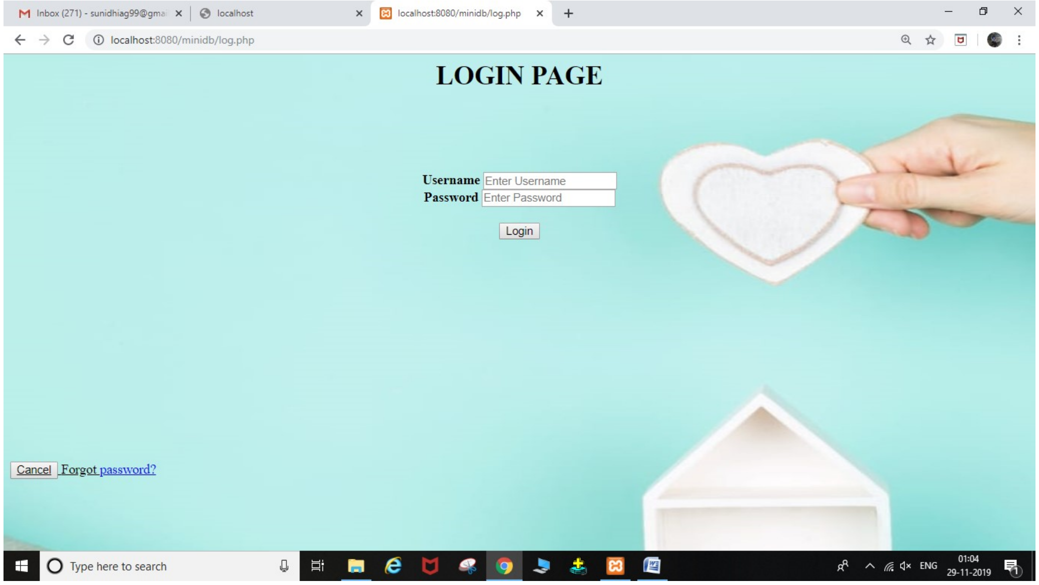
Chapter 4

Results and Discussions

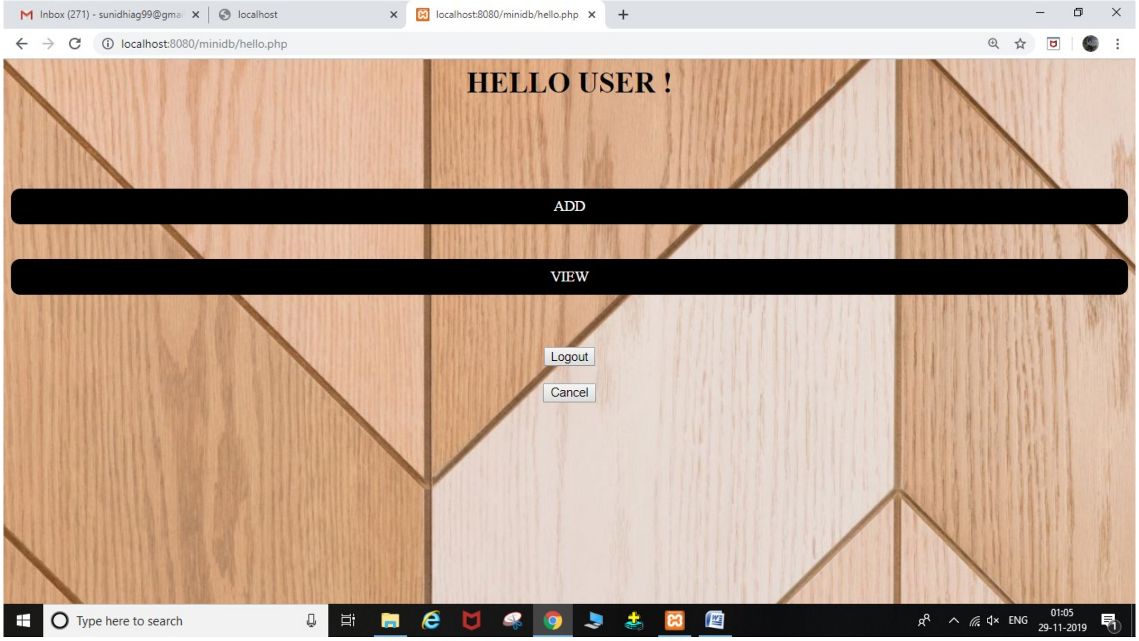
Snapshots:



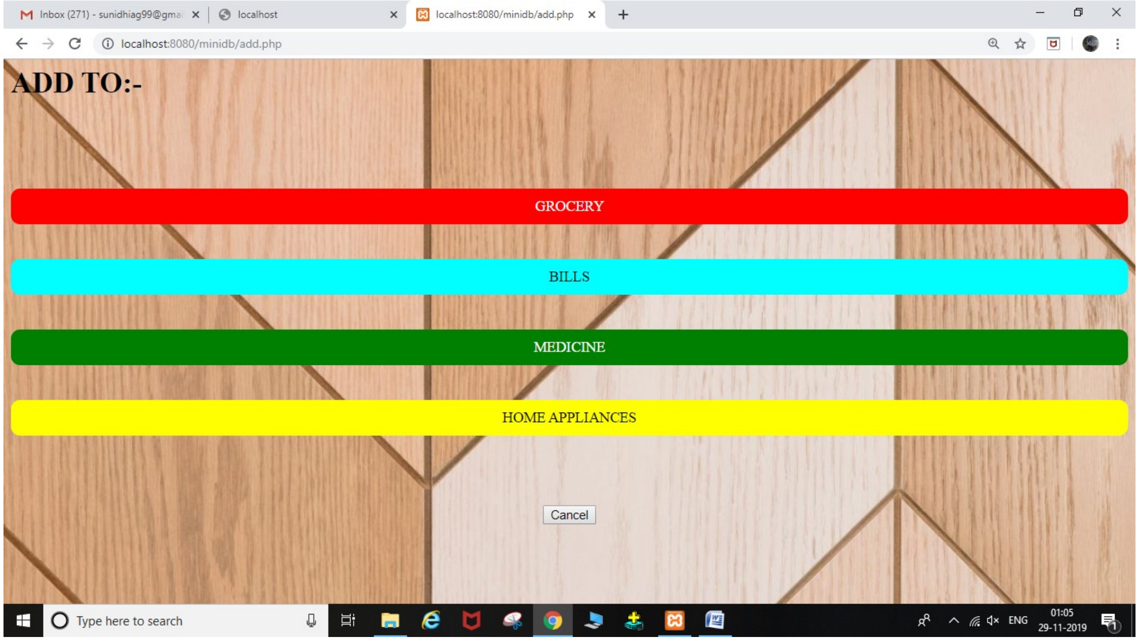
Homepage: The figure 4.1 shows the home page of the Home Management System.

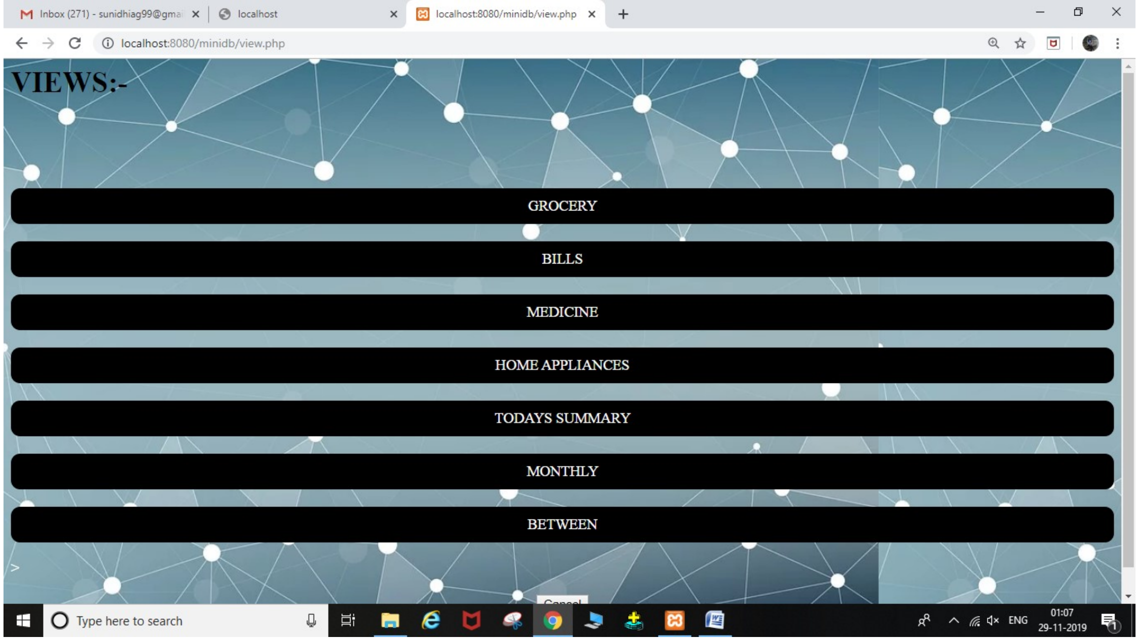


Login page: figure 4.2 display the login page.

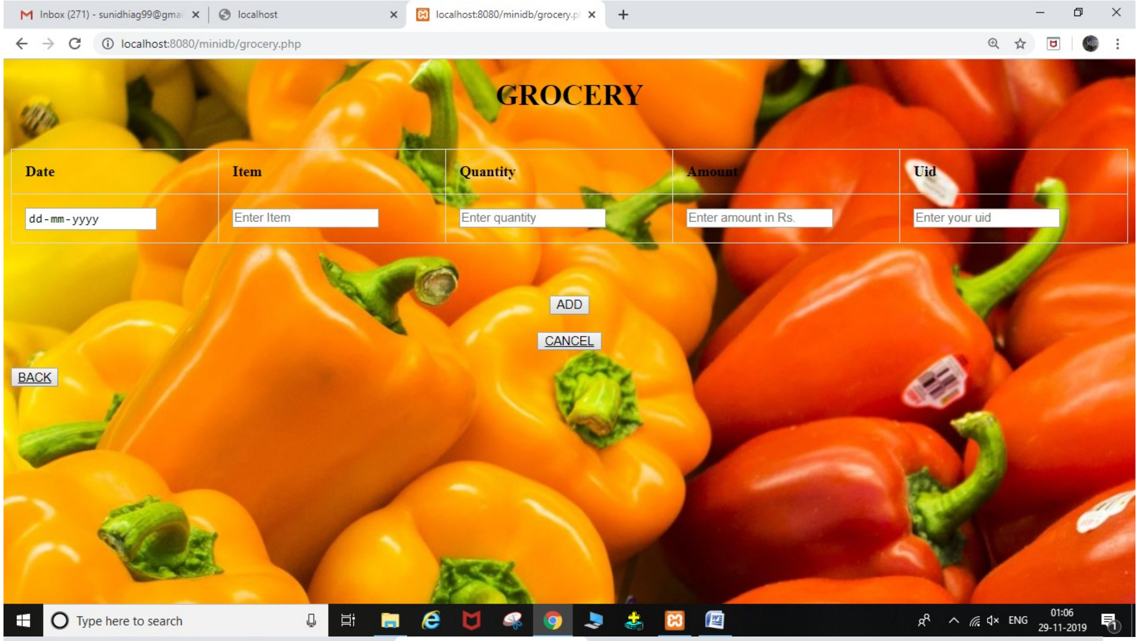

Inbox (271) - sunidhiag99@gma
localhost
localhost:8080/minidb/hello.php
localhost:8080/minidb/hello.php
HELLO USER !
ADD
VIEW
Logout
Cancel
Type here to search
ENG
01:05
29-11-2019

Hello User:figure 4.3 This page displays add and views options to user.



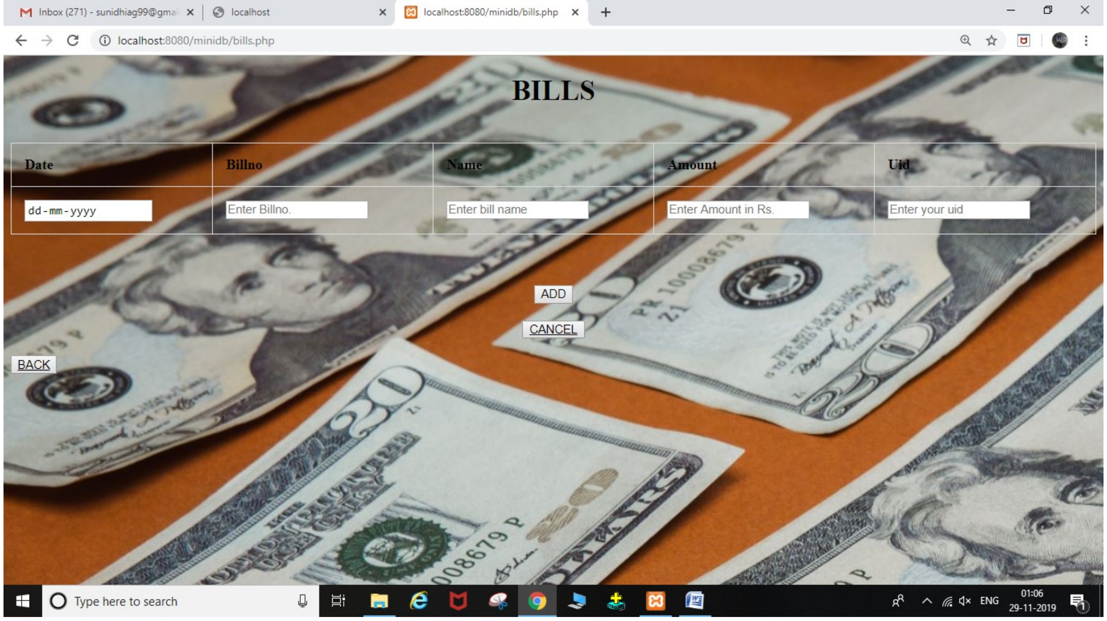


ADD TO and VIEWS: Figure 4.4 This two pages displays variants available to add and view.


Inbox (271) - sunidhiag99@gma
localhost
localhost:8080/minidb/grocery.p
localhost:8080/minidb/grocery.php
ADD
CANCEL
BACK
Type here to search
ENG
01:06
29-11-2019

GROCERY

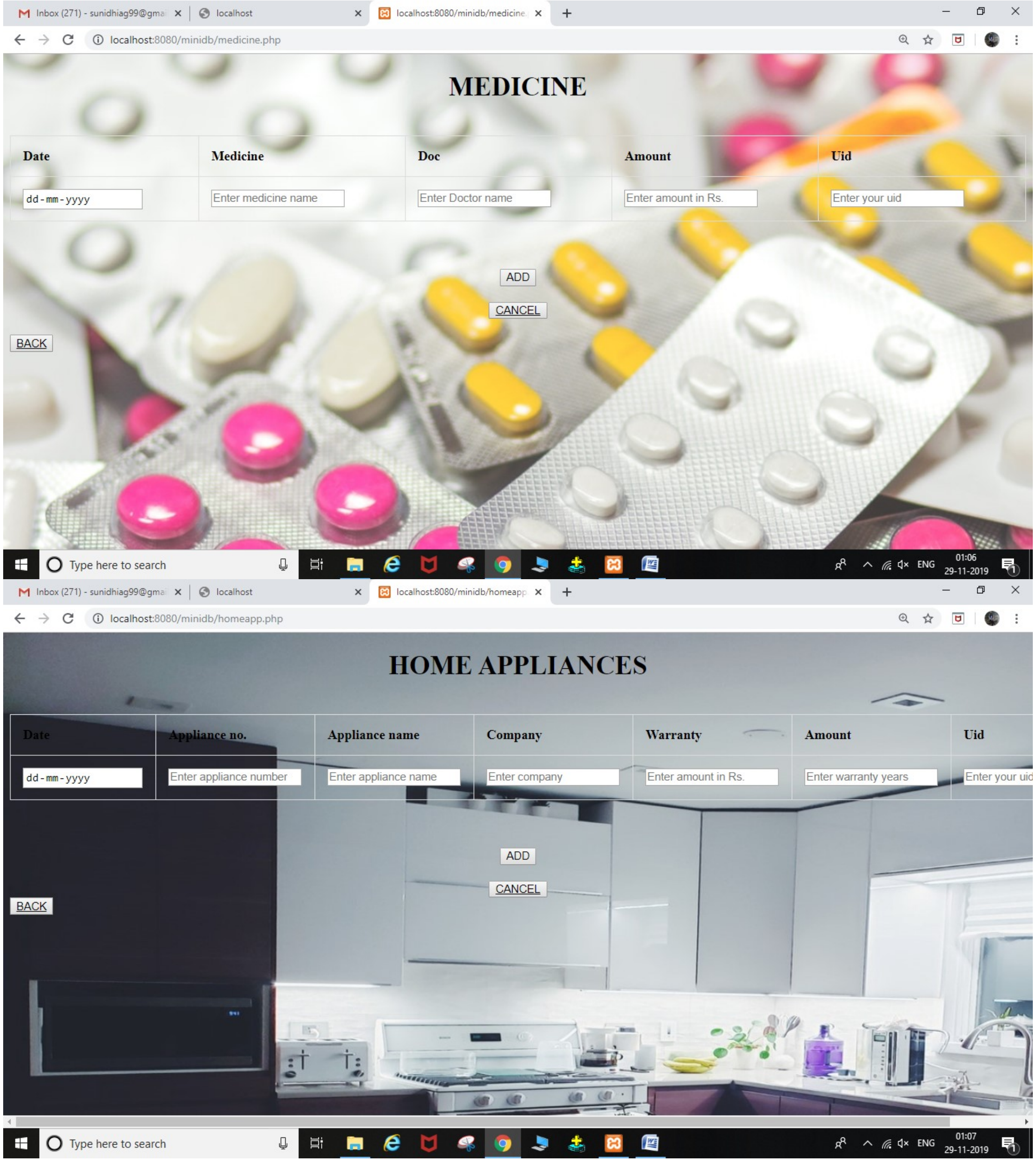
Date	Item	Quantity	Amount	Uid
dd-mm-yyyy	Enter Item	Enter quantity	Enter amount in Rs.	Enter your uid


Inbox (271) - sunidhiag99@gma
localhost
localhost:8080/minidb/bills.php
localhost:8080/minidb/bills.php
ADD
CANCEL
BACK
Type here to search
ENG
01:06
29-11-2019

BILLS

Date	Billno	Name	Amount	Uid
dd-mm-yyyy	Enter Billno.	Enter bill name	Enter Amount in Rs.	Enter your uid

GROCERY,BILLS,MEDICINE,HOME APPS: Figure 4.5 This page displays pages for user to add the details and store in database.


Inbox (271) - sunidhiag99@gma
localhost
localhost:8080/minidb/medicine
localhost:8080/minidb/medicine.php
ADD
CANCEL
BACK
Type here to search
ENG
01:06
29-11-2019

MEDICINE

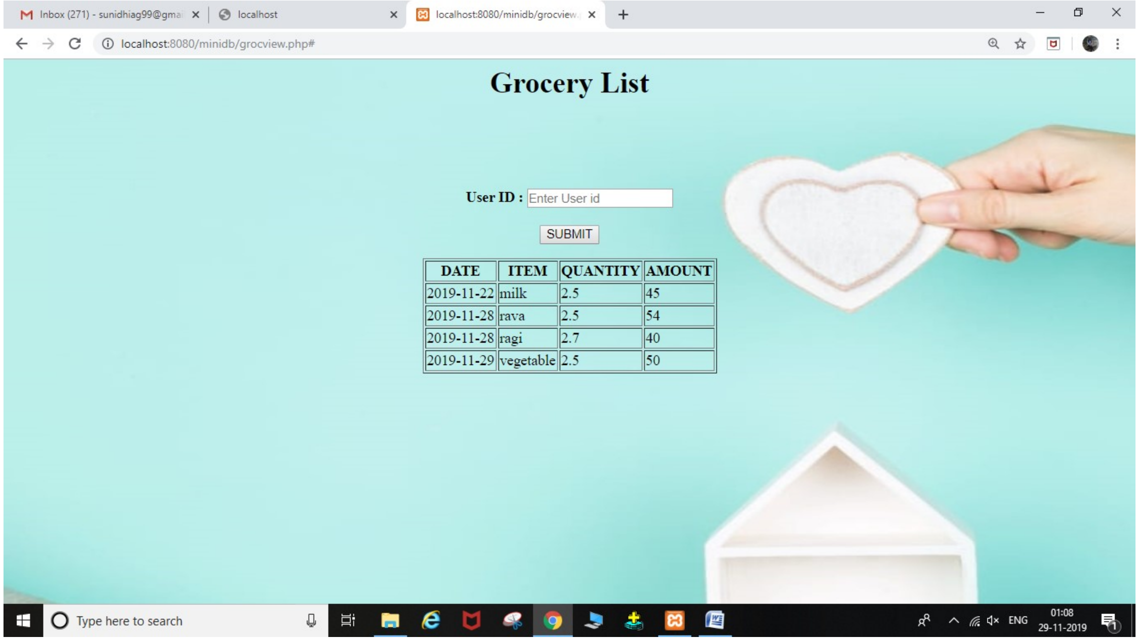
Date	Medicine	Doc	Amount	Uid
dd-mm-yyyy	Enter medicine name	Enter Doctor name	Enter amount in Rs.	Enter your uid

Inbox (271) - sunidhiag99@gma
localhost
localhost:8080/minidb/homeapp
localhost:8080/minidb/homeapp.php
ADD
CANCEL
BACK
Type here to search
ENG
01:07
29-11-2019

HOME APPLIANCES

Date	Appliance no.	Appliance name	Company	Warranty	Amount	Uid
dd-mm-yyyy	Enter appliance number	Enter appliance name	Enter company	Enter amount in Rs.	Enter warranty years	Enter your uid

GROCERY,BILLS,MEDICINE,HOME APPS: Figure 4.5 This page displays pages for user to add the details and store in database.

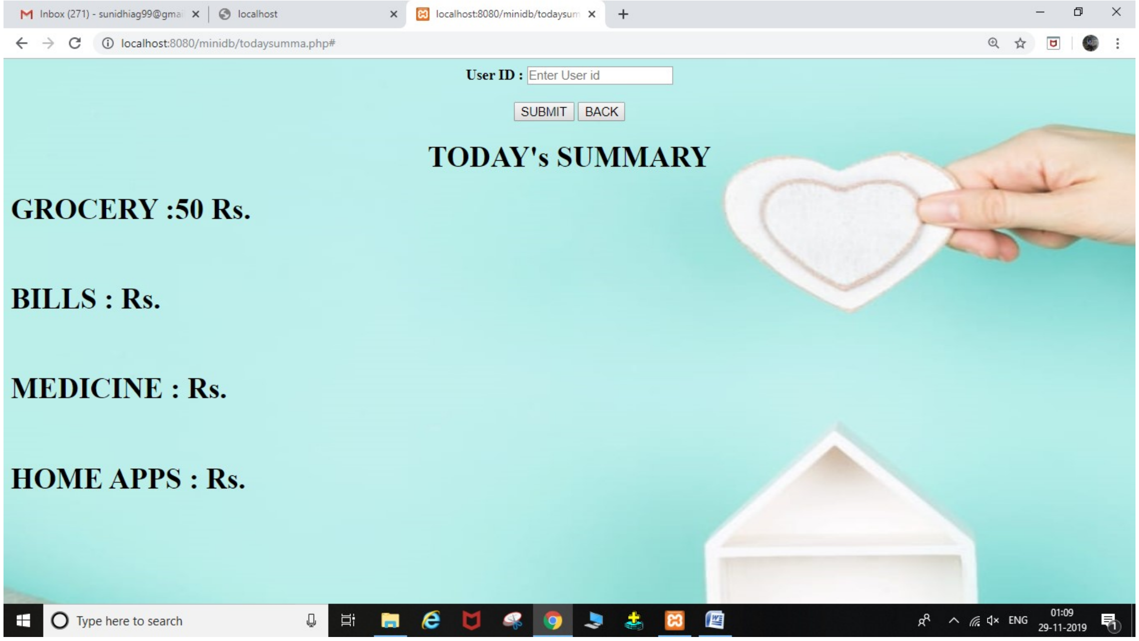

Inbox (271) - sunidhiag99@gma
localhost
localhost:8080/minidb/grocview
localhost:8080/minidb/grocview.php#
Type here to search
ENG
01:08
29-11-2019

Grocery List

User ID : Enter User id

DATE	ITEM	QUANTITY	AMOUNT
2019-11-22	milk	2.5	45
2019-11-28	rava	2.5	54
2019-11-28	ragi	2.7	40
2019-11-29	vegetable	2.5	50

Grocery View: figure 4.7 this page shows the grocery list added by the user with uid. Similarly for bills,medicines and homeapp it will appear.


Inbox (271) - sunidhiag99@gma
localhost
localhost:8080/minidb/todaysum
localhost:8080/minidb/todaysumma.php#
SUBMIT
BACK
Type here to search
ENG
01:09
29-11-2019

User ID : Enter User id

TODAY's SUMMARY

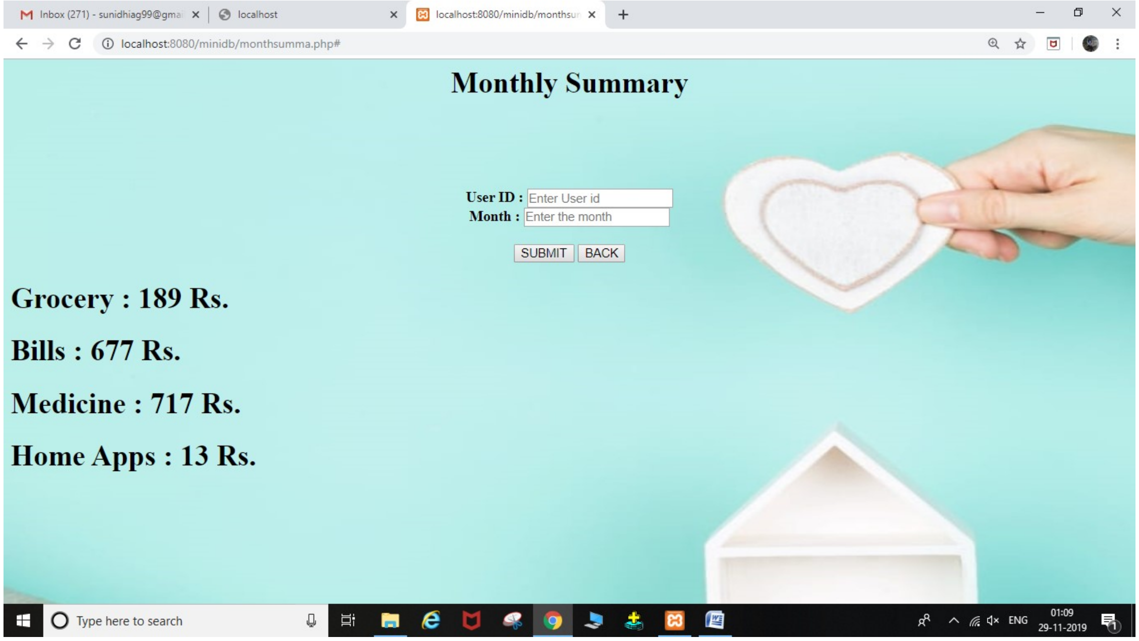
GROCERY :50 Rs.

BILLS : Rs.

MEDICINE : Rs.

HOME APPS : Rs.

Today's Summary: Figure 4.8 This page displays users total expense for today in each category.


Inbox (271) - sunidhiag99@gma
localhost
localhost:8080/minidb/monthsu
localhost:8080/minidb/monthsumma.php#
Enter User id
Enter the month
SUBMIT
BACK
Type here to search
ENG
01:09
29-11-2019

Monthly Summary

User ID :
Month :

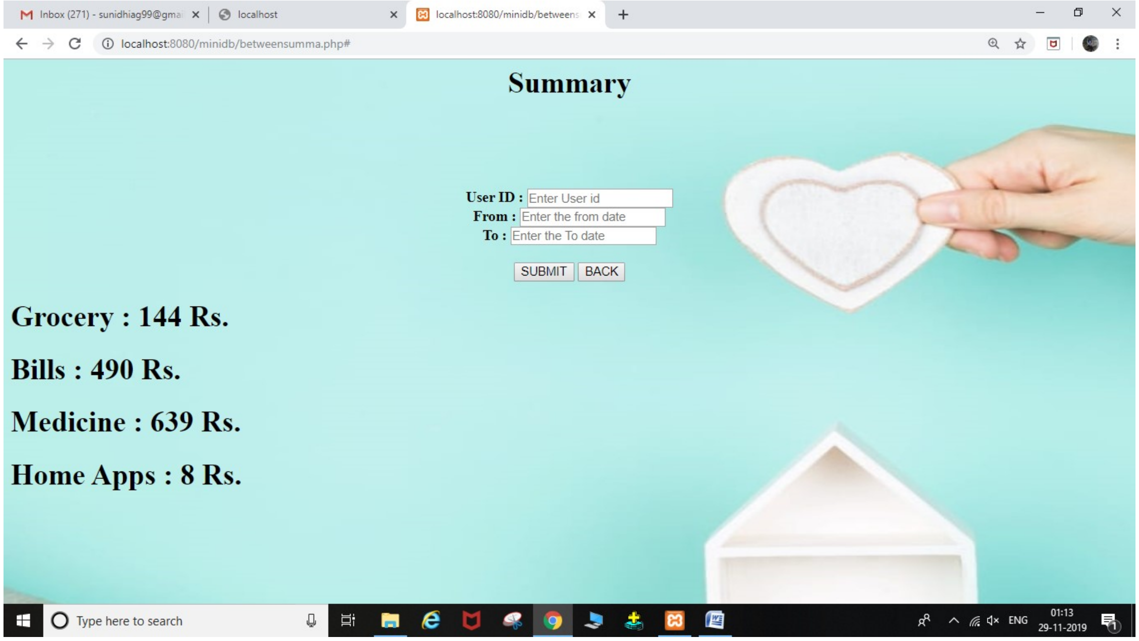
Grocery : 189 Rs.

Bills : 677 Rs.

Medicine : 717 Rs.

Home Apps : 13 Rs.

Monthly Summary: figure 5.0 this page shows the monthly summary of users total in each category in provided month.



Summary

User ID :
From :
To :

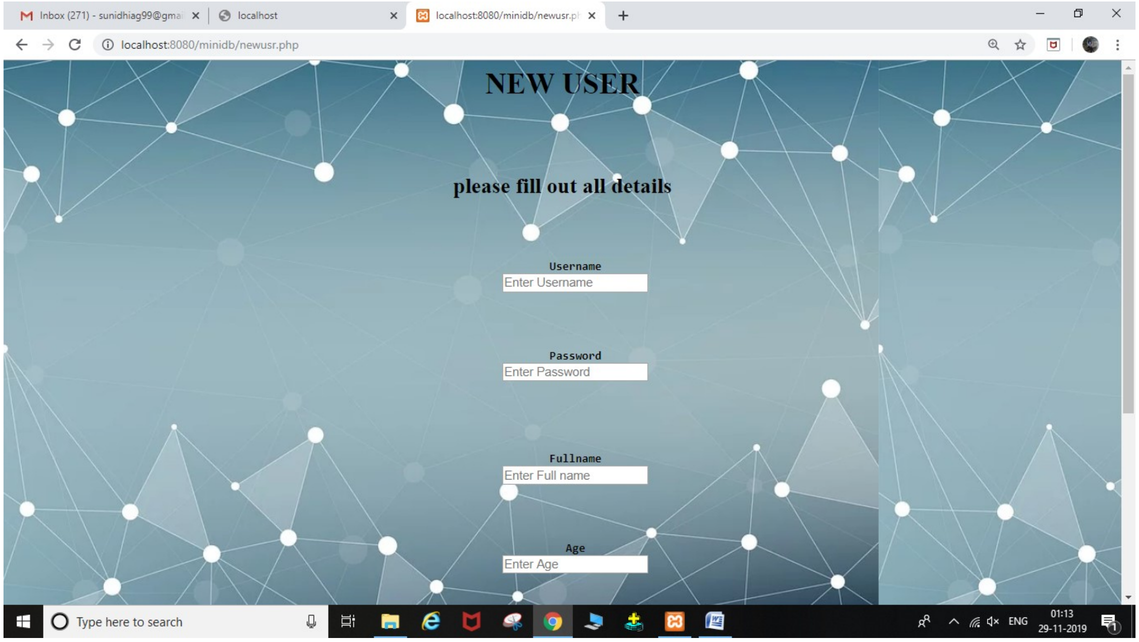
Grocery : 144 Rs.

Bills : 490 Rs.

Medicine : 639 Rs.

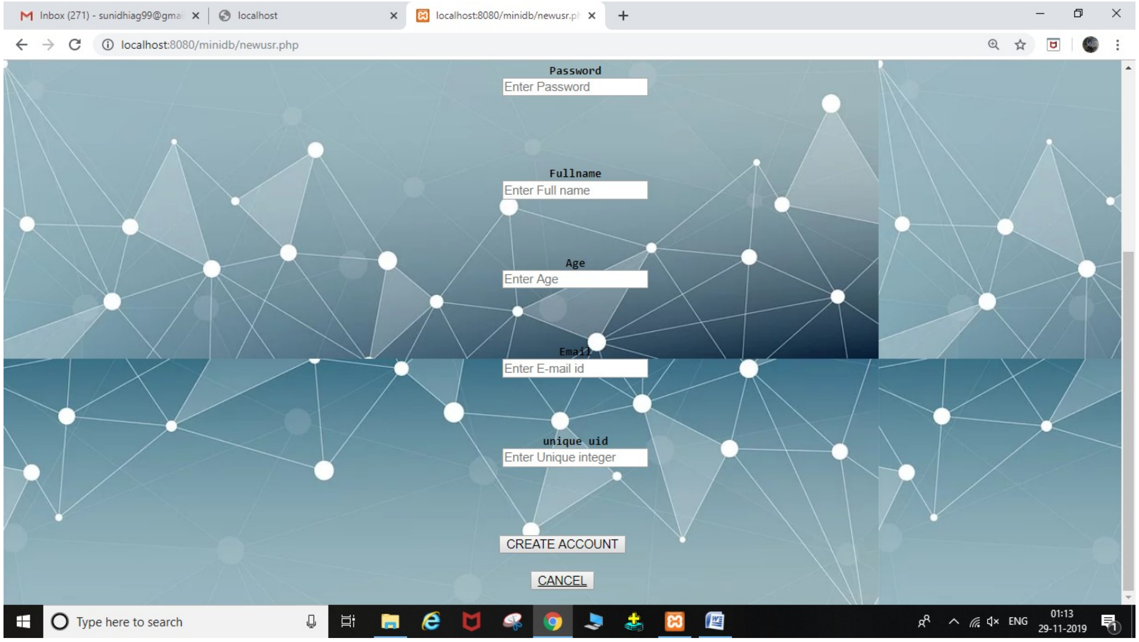
Home Apps : 8 Rs.

Between Summary: figure 5.1this page shows the users summary from within the specified range.



NEW USER

please fill out all details



New user: figure 5.2 this page shows the new account creation page for new user.

Chapter 5

Conclusion and Future Enhancements

5.1 Conclusion

The Home Management System indicates Intelligent use of various technologies and enables the ease of operations of day to day activities.

The views are customized to get proper results accouding to user entered data.

The Addition of stored procedures makes the application more dynamic and faster.

5.2 Further Enhancement

- This application acts as a prototype of a real-time large scale web application that can be implemented in client server three tier architecture .
- This right now has a minimal but useful functionalities.
- The functionalities can be extended and more features can also be added.
- It can be implemented into as an Android or IOS mobile applications.
- Many more Security protocols and measures can be taken.

References

[1] The Complete Reference J2EE, Jim Keogh, Tata McGraw Hill publications,   2002.

[2].Java The Complete Reference, Herbert Schildt, Oracle Tata McGraw Hill publications, 2011.

[3].Fundamentals of Database Management system, Ramez Elmasri, Shamkant B Navathe, Pearson publications, 7th edition.

[4].https://www.tutorialspoint.com/jsp/jsp_overview.html

[5].https://www.quickbase.com/articles/database-software-basics.html

[6].https://www.wikipedia.com .