CHAPTER – 1

INTRODUCTION

InvestToday is India's friendliest online Investment Platform. Here, Investors (resident Indians and INRs) get access to wide range of Mutual Funds, Equities from the Bombay Stock Exchange (BSE), Corporate Deposits from premium companies, and various other investment products in one convenient online location.

1.1 PROJECT

The Shares, Mutual Funds and Investment Management System is designed to provide a simple way for prospective businessmen (Target Audience of this Website) a platform to view the current stock market value. They can view different housing projects by well-known companies, and have access to the different mutual fund policies that they could make a choice to adhere to possibly in the near future. This website also hosts a platform for Start-ups to be recognised, so interested established businessmen can make a decision to work with them and invest shares in their Start-ups.

This platform also hosts a view of the shares currently invested by highly valued entities. The motive of this project is to provide the above mentioned idea a web platform for the prospective users to view the different idea available in the market. We encourage the businessmen to step out of their comfort zone of how they might work through the implementation of a business idea, and join hands with Start-ups and experiment the approach of implementing business ideas, given that the business idea may or may have not stemmed out of the existing ones. The online platform will also support ideas brimmed from the modern conventions.

It will allow a user to view the website and what it has to offer when visited. But only a registered user can exercise the extended features of :

- Allowing companies to advertise their Mutual Funds Policies by making details of the latter available to the web developer.

- Allowing premium housing brands to advertise their housing projects with the details of the same attached while providing the web developer the details.
- The Start-ups find a resourceful platform to showcase their business ideas to interested passer-by's or registered users of the website. Start-ups new to the website may add their details and enclose with that web URL of their upcoming brand.

1.2 DBMS

A database management system (DBMS) is system software for creating and managing databases. The DBMS provides users and programmers with a systematic way to create, retrieve, update and manage data.

A DBMS makes it possible for end users to create, read, update and delete data in a database. The DBMS essentially serves as an interface between the database and end users or application programs, ensuring that data is consistently organized and remains easily accessible.

A DBMS provides concurrency, security, data integrity, consistency, controls redundancy and data independence.

In this project the Relational DBMS (RDBMS) used is MySQL. It is an open source software which uses SQL (Structured Query Language) which is a standard language for storing, manipulating and retrieving data in databases.

1.3 JAVA CONNECTIONS

To connect the database with the front end we use a java connector JDBC (Java Database Connectivity). JDBC is an application programming interface (API) for the programming language Java, which defines how a client may access a database. It is Java based data access technology and used for Java database connectivity. It is part of the Java Standard Edition platform, from Oracle Corporation.

To achieve connectivity we use JSPs (JavaServer Pages) and Servlets in this project. JavaServer Pages (JSP) is a technology that helps software developers create dynamically generated web pages based on HTML, XML, or other document types. JSP is similar to PHP and ASP, but it uses the Java programming language.

CHAPTER – 2

REQUIREMENT SPECIFICATIONS

A computerized way of handling information about members and batch details is efficient, organized and time saving, compared to a manual way of doing so. This is done through a database driven web application whose requirements are mentioned in this section.

2.1 OVERALL DESCRIPTION

A reliable and scalable database driven web application with security features, that is easy to use and maintain is the requisite.

2.2 SPECIFIC REQUIREMENTS

The specific requirements of the Gym Database Management System are stated as follows:

2.2.1 SOFTWARE REQUIREMENTS

Technology used:

- Front end – JSP
- Controller – JSP/Servlets
- Backend – mySQL

Software:

- IDE – Netbeans 8.2
- Database support – MySQL 5.7
- Operating system – Windows 8 and above
- Server deployment – Glassfish server

Technology:

- HTML is integrated in JSP. It provides a means to structure text based information in a document. It allows users to produce web pages that include text, graphics and hyperlinks.
- Javascript is a scripting language which supports the development of both client and server applications. It is preferred at client side to write programs that can be executed by a web browser within the context of a web page.
- CSS(Cascading Style Sheets) is a style sheet language used for describing the presentation of a document written in a markup language. Although most often used to set the visual style of web pages and user interfaces written in HTML and XHTML, the language can be applied to any XML document.
- SQL is the language used to manipulate relational databases. It is tied closely with the relational model. It is issued for the purpose of data definition and data manipulation.
- Java Server pages is a simple yet powerful technology for creating and maintaining dynamic-content web pages. It is based on the Java programming language. It can be thought of as an extension to servlet because it provides more functionality than servlet A JSP page consists of HTML tags and JSP tags. The jsp pages are easier to maintain than servlet because we can separate designing and development.

We require a JDBC connection between the front end and back end components to write to the database and fetch required data.

CHAPTER – 3

DETAILED DESIGN

3.1 SYSTEM DESIGN

The web server needs a JSP engine, i.e, a container to process JSP pages. The JSP container is responsible for intercepting requests for JSP pages. A server(generally referred to as application or web server) supports the Java Server Pages. This server will act as a mediator between the client browser and a database. The following diagram shows the JSP architecture.

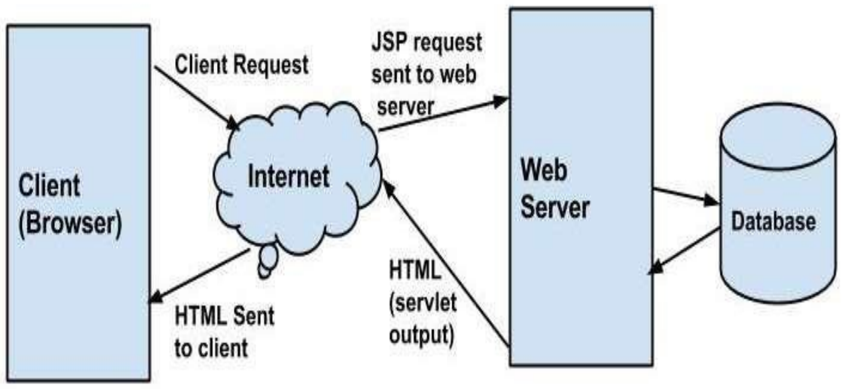


Fig 3.1: JSP architecture

Three-tier Client / Server database architecture is commonly used architecture for web applications. Intermediate layer called Application server or Web Server stores the web connectivity software and the business logic(constraints) part of application used to access the right amount of data from the database server. This layer acts like medium for sending partially processed data between the database server and the client.

Database architecture focuses on the design, development, implementation and maintenance of computer programs that store and organize information for businesses, agencies and institutions. A database architect develops and implements software to meet the needs of users. Several types of databases, including relational or multimedia, may be created. Additionally, database architects may use one of several languages to create databases, such as structured query language (SQL). SQL is a database computer language designed for the retrieval and management of data in a relational database.

3.2 ER DIAGRAMS

An entity–relationship model is usually the result of systematic analysis to define and describe what is important to processes in an area of a business.

An E-R model does not define the business processes; it only presents a business data schema in graphical form. It is usually drawn in a graphical form as boxes (entities) that are connected by lines (relationships) which express the associations and dependencies between entities.

An ER model can also be expressed in a verbal form, for example: one building may be divided into zero or more apartments, but one apartment can only be located in one building.

Entities may be characterized not only by relationships, but also by additional properties (attributes), which include identifiers called "primary keys". Diagrams created to represent attributes as well as entities and relationships may be called entity-attribute-relationship diagrams, rather than entity-relationship models.

An ER model is typically implemented as a database. In a simple relational database implementation, each row of a table represents one instance of an entity type, and each field in a table represents an attribute type. In a relational database a relationship between entities is implemented by storing the primary key of one entity as a pointer or "foreign key" in the table of another entity

There is a tradition for ER/data models to be built at two or three levels of abstraction. Note that the conceptual-logical-physical hierarchy below is used in other kinds of specification, and is different from the three schema approach to software engineering. While useful for organizing data that can be represented by a relational structure, an entity-relationship diagram can't sufficiently represent semi-structured or unstructured data, and an ER Diagram is unlikely to be helpful on its own in integrating data into a pre- existing information system.

Three main components of an ERD are the entities the relationship between those entities, and the cardinality, which defines that relationship in terms of numbers.
Cardinality notations define the attributes of the relationship between the entities. Cardinalities can denote that an entity is optional (for example, an employee rep could have no customers or could have many) or mandatory (for example, there must be at least one product listed in an order.)

The three main cardinal relationships are:

- One-to-one (1:1) – For example, if each member in a database is associated with one mailing address.
- One-to-many (1:M) – In the database, a person can invest in many projects and have shares in them.
- Many-to-many (M:N) – When N company groups invest is M projects .

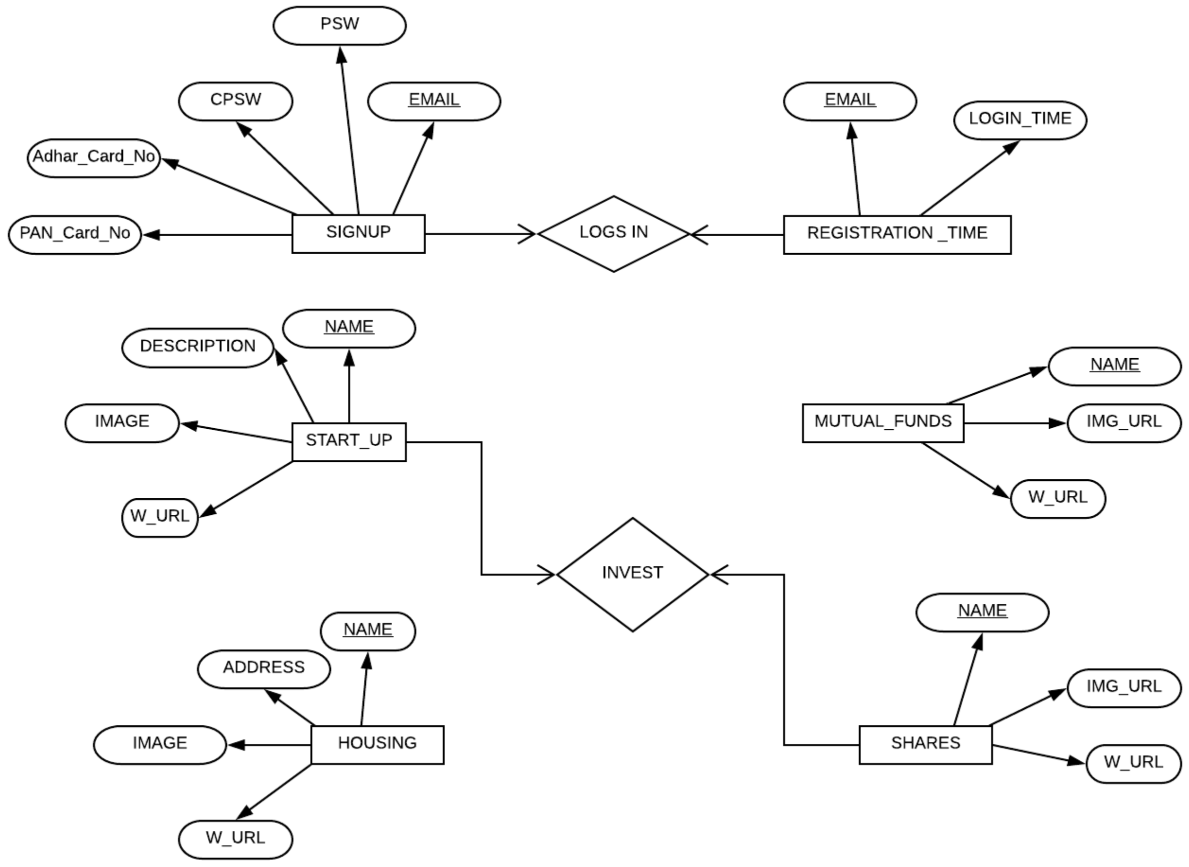


Fig 3.2: ER Diagram

3.3 RELATIONAL SCHEMA

The term "schema" refers to the organization of data as a blueprint of how the database is constructed (divided into database tables in the case of relational databases). The formal definition of a database schema is a set of formulas (sentences) called integrity constraints imposed on a database.

A relational schema shows references among fields in the database. When a primary key is referenced in another table in the database, it is called a foreign key. This is denoted by an arrow with the head pointing at the referenced key attribute.

A schema diagram helps organize values in the database. It also gives an idea of what order the tables should be created in. The following diagram shows the schema diagram for the database.

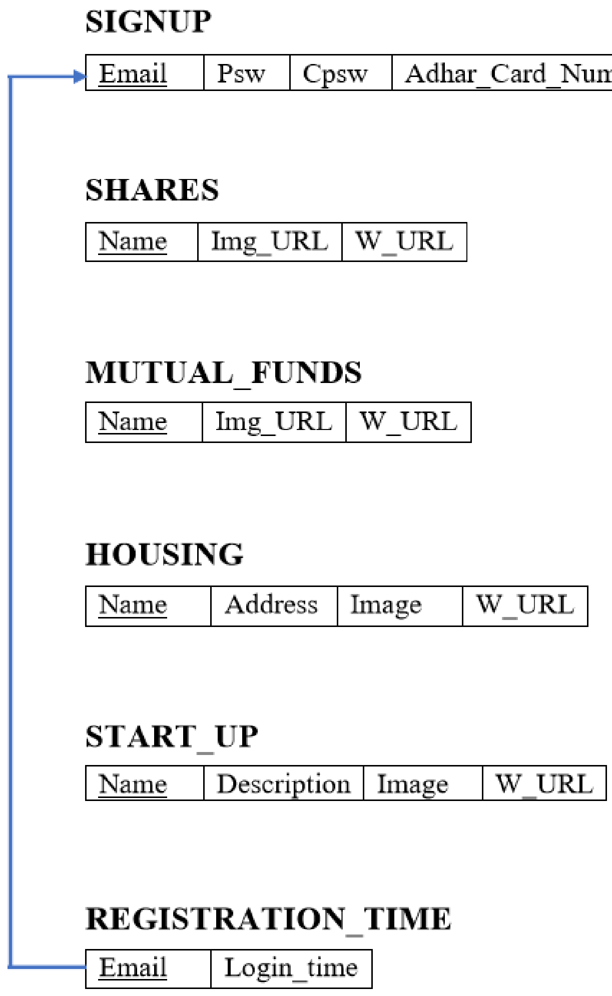


Fig 3.3: Relational Schema Diagram

3.4 DESCRIPTION OF THE TABLES

The database under consideration consists of 5 tables :

1. HOUSING
   - NAME – is the name of the housing project
   - ADDRESS – The location of the housing project currently under construction
   - IMAGE – An image of the future outlook of the housing project
   - W_URL – The website of the business hosting the particular hosting project

2. MUTUAL
   - NAME – Name of the mutual fund policy
   - IMG_URL – a picture of the logo
   - W_URL – The website of the bank advertising the mutual fund policy

3. SHARES
   - NAME – Name of the company in which they have invested shares
   - IMG_URL – Logo of the company
   - W_URL – Website of the company

4. REGISTRATION TIME
   - EMAIL – email address of the registered user
   - LOGIN_TIME – The login time of the session

5. SIGN_UP
   - EMAIL – email address of the user given during signing in

- PSW – User’s choice of password
- CPSW – Confirmation field of the password
- ACN – Adhar Card Number
- PCN – PAN Card Number

6. START_UP
    - NAME – Name of the Start-up
    - DESC – Description of the Start-up
    - IMG – Logo of the Start-up
    - W_URL – Website path that advertises the Start-up (if any)

CHAPTER 4

IMPLEMENTATION

4.1 MODULES AND THEIR REQUIREMENTS

1. User Registration: A user can register themselves with the website to be linked to the platform and exercise features such as advertising their housing projects, or advertising the mutual fund policies on behalf of the company.

2. Adding your Own Housing project details: You can add details of housing projects undertaken by you.

3. Platform for advertising Start-ups: A feature that can be exploited by budding Start-ups to advertise their business idea by making their details available on the website .

4. Add Mutual Fund Policies: Adding Mutual Fund Policies to the existing several out there, but being uniquely identified by prospective users who are registered on our website.

5. View the stock Market Statistics: This is will enable you to keep tabs on the endeavors of the current stock market.

Scope:

This system is flexible and efficient and allows easy access to member information, security, speed and accuracy should be focused on. It is a user friendly system and can overcome some user validation checks. A user can upload any feedback which will be viewed by the owner to make improvements.

All the shares information is suitably maintained on the server and can be accessed when required. It identifies various sources of information and accordingly provides access to the requested details.

The system maintains details such as type of batch, instructor, locations available, etc.

Triggers and stored procedures:

The project makes use of a trigger to compile details of the member signing up on the portal to keep track of user activity on the site. The trigger automatically logs the information of the member along with the signing up date.

A stored procedure is used to check whether the user is valid and returns their email ID when they log in. The procedure takes email as input parameter. It checks the presence of such user in the database. If present, it fetches the corresponding email ID and returns it as an output parameter.

4.2 RESULTS

The resulting system is able to:

- Authenticate user credentials during login
- Allows users to easily look at the shares screen
- The user can upload their feedback
- Stores details as uploaded by the user

CHAPTER – 5

TESTING

5.1 SOFTWARE TESTING

Testing is the process used to help identify correctness, completeness, security and quality of developed software. This includes executing a program with the intent of finding errors. It is important to distinguish between faults and failures. Software testing can provide objective, independent information about the quality of software and risk of its failure to users or sponsors. It can be conducted as soon as executable software (even if partially complete) exists. Most testing occurs after system requirements have been defined and then implemented in testable programs.

5.2 MODULE TESTING AND INTEGRATION

Module testing is a process of testing the individual subprograms, subroutines, classes, or procedures in a program. Instead of testing whole software program at once, module testing recommend testing the smaller building blocks of the program. It is largely white box oriented. The objective of doing Module testing is not to demonstrate proper functioning of the module but to demonstrate the presence of an error in the module. Module testing allows to implement parallelism into the testing process by giving the opportunity to test multiple modules simultaneously.

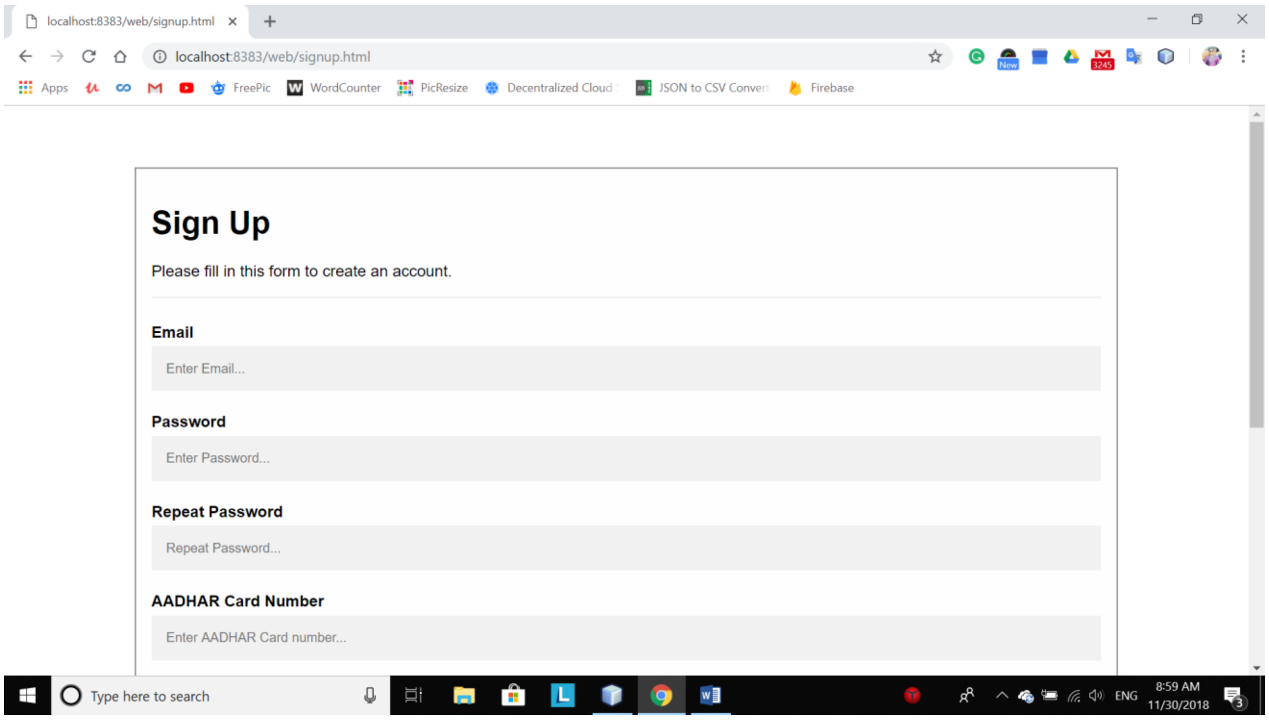
In this Database Management System, when a user logs in or registers, their email Id for the session is maintained internally. When a user wants to upload details, they are uploaded to the respective details depending on what they want to add. The system displays all shares uploaded by the owner when a user logs in. All the feedback provided by the members are stored and viewed by the owner to make the necessary improvements. Hence all the modules are linked by identifying entities that are maintained in the database.

CHAPTER – 6

SNAPSHOTS

This chapter consists of working screenshots of the project with code snippets.

6.1 SIGN UP



Sign Up

Please fill in this form to create an account.

Email

Enter Email...

Password

Enter Password...

Repeat Password

Repeat Password...

AADHAR Card Number

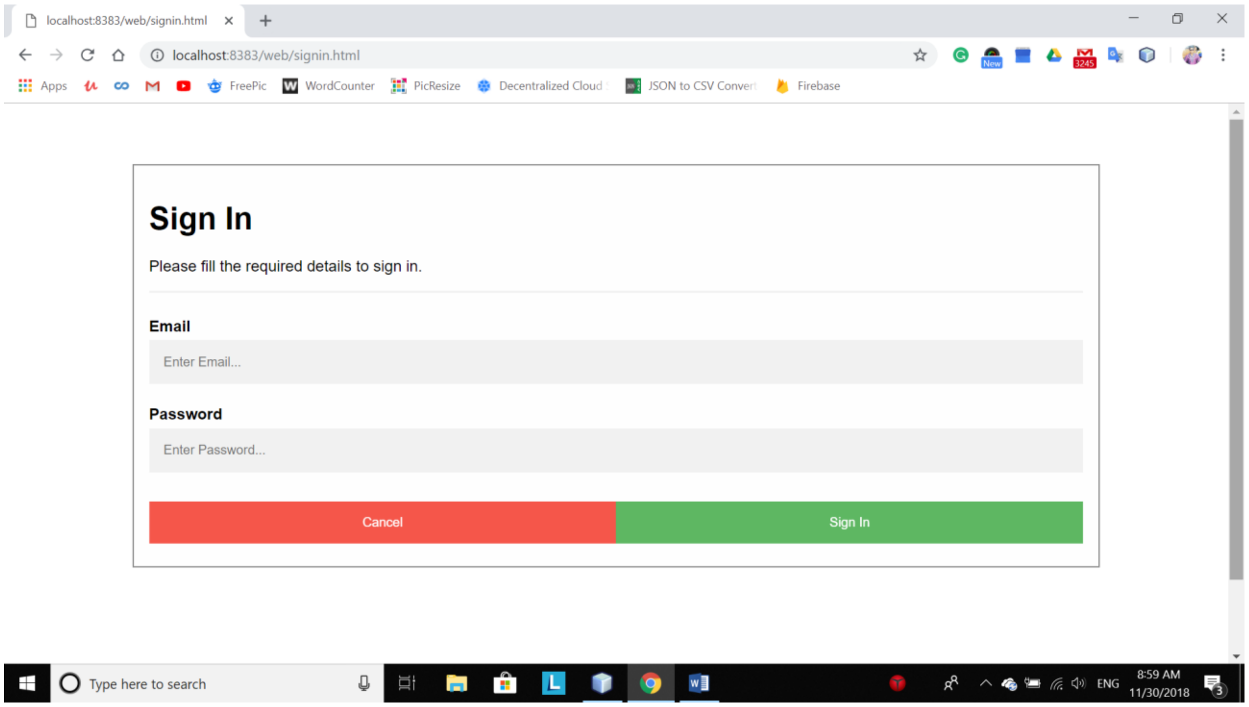
Enter AADHAR Card number...

Fig 6.1: Sign-up page

```
<body>
<form class="modal-content" action="signup.jsp">
<div class="container">
<h1>Sign Up</h1>
<p>Please fill in this form to create an account.</p>
<hr>
<label for="email"><b>Email</b></label
```

```
<input type="text" placeholder="Enter Email..." name="email" required>

<label for="psw"><b>Password</b></label>
<input type="password" placeholder="Enter Password..." name="psw" required>

<label for="psw-repeat"><b>Repeat Password</b></label>
<input type="password" placeholder="Repeat Password..." name="cpsw" required>

<label for="psw-repeat"><b>AADHAR Card Number</b></label>
<input type="text" placeholder="Enter AADHAR Card number..." name="acn" required
<label for="psw-repeat"><b>PAN Card Number</b></label>
input type="text" placeholder="Enter PAN Card number..." name="pcn" required>

<label>
<input type="checkbox" checked="checked" name="remember" style="margin-bottom:15px">
Remember me
</label>

<p>By creating an account you agree to our <a href="#" style="color:dodgerblue">Terms &
Privacy</a>.</p>

<div class="clearfix">
<button type="submit" class="signupbtn">Sign Up</button>
</div>
</div>
</form>
</body>
```

6.2 SIGN IN



Sign In

Please fill the required details to sign in.

Email

Enter Email...

Password

Enter Password...

Cancel Sign In

Fig 6.2: Sign-in page

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Sign-in</title>
</head>
<body>
<%@page import ="java.sql.*" %>
<%@page import="javax.sql.*" %>
<%
String email=request.getParameter("email");
String psw=request.getParameter("psw");
```

```
try
{
Class.forName("com.mysql.jdbc.Driver").newInstance();
java.sql.Connection
con=DriverManager.getConnection("jdbc:mysql://localhost:3306/invest_today?useSSL=false",
"root", "root");
String Query="select * from signup where email=? and psw=?";
PreparedStatement psm=con.prepareStatement(Query);
psm.setString(1, email);
psm.setString(2, psw);
ResultSet rs=psm.executeQuery();
if(rs.next())
{
response.sendRedirect("index.html");
}
else
{
response.sendError(401,"Invalid Credentials");
}
}
catch(Exception e)
{
out.println(e);
}
%>
</body>
</html>
```

6.3 NAVBAR IN HOME PAGE

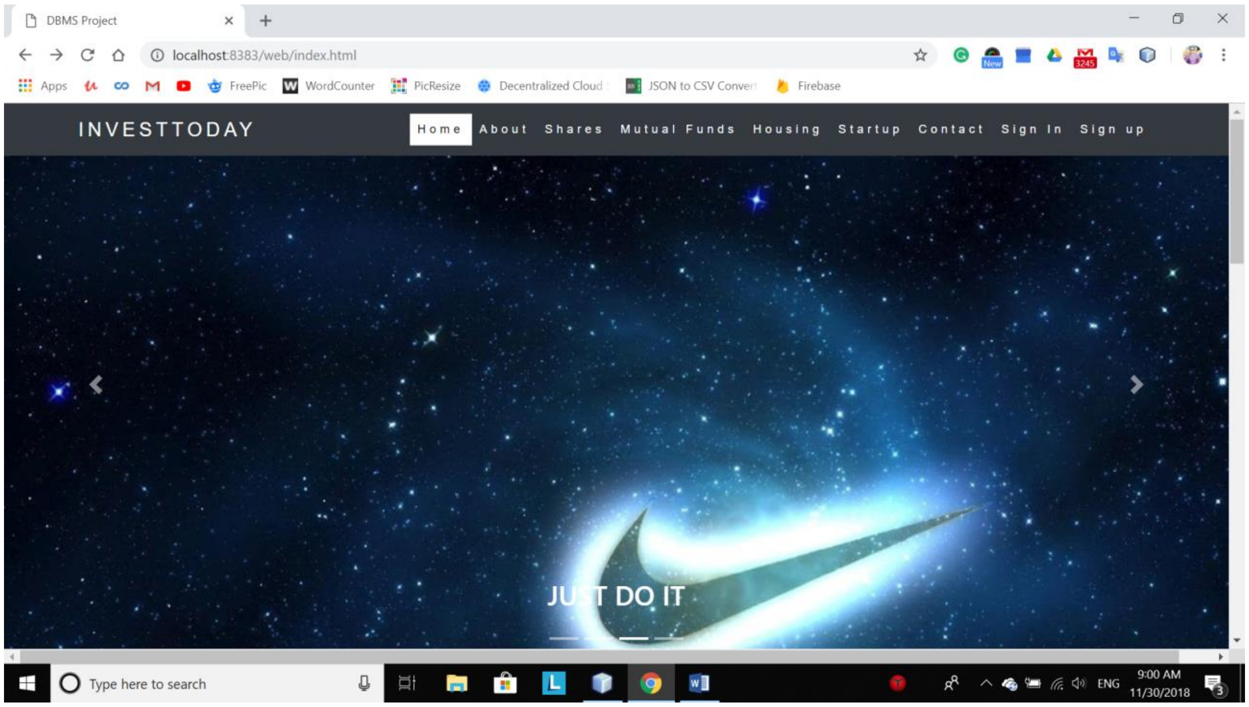
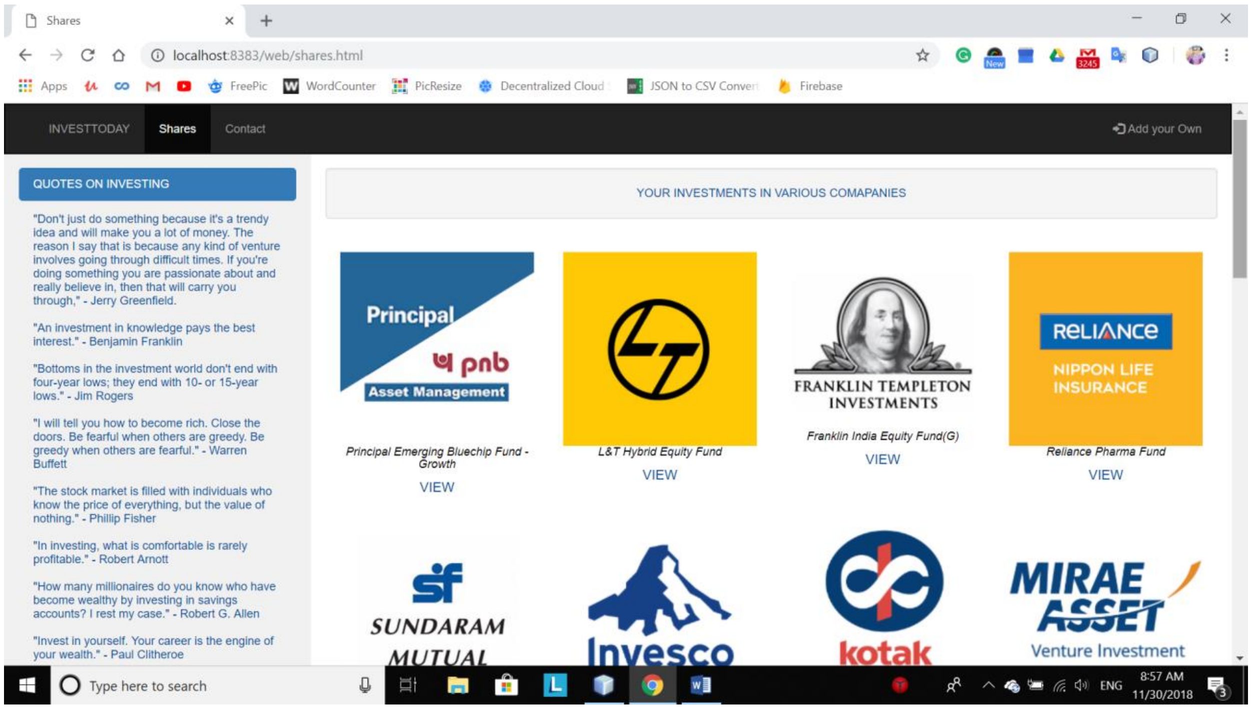


Fig 6.3: Navbar

```
<nav class="navbar navbar-expand-lg navbar-dark bg-dark fixed-top">
<div class="container">
<a class="navbar-brand" href="#">INVESTTODAY</a>
<button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarResponsive" aria-controls="navbarResponsive" aria-expanded="false" aria-
label="Toggle navigation">
<span class="navbar-toggler-icon"></span>
</button>
<div class="collapse navbar-collapse" id="navbarResponsive">
<ul class="navbar-nav ml-auto">
<li class="nav-item active">
<a class="nav-link" href="#">Home
<span class="sr-only">(current)</span>
```

```
</a>
</li>
<li class="nav-item">
<a class="nav-link" href="#">About</a>
</li>
<li class="nav-item">
<a class="nav-link" href="shares.html">Shares</a>
</li>
<li class="nav-item">
<a class="nav-link" href="mutualfunds.html">Mutual Funds</a>
</li>
<li class="nav-item">
<a class="nav-link" href="housing.html">Housing</a>
</li>
<li class="nav-item">
<a class="nav-link" href="product.html">Startup</a>
</li>
<li class="nav-item">
<a class="nav-link" href="https://www.facebook.com/ketan.keshav.007">Contact</a>
</li>
<li class="nav-item">
<a class="nav-link" href="signin.html">Sign In</a>
 </li>
<li class="nav-item">
<a class="nav-link" href="signup.html">Sign up</a>
</li>
</ul>
</div>
</div>
</nav>
```

6.4 SHARES VIEW



"Don't just do something because it's a trendy idea and will make you a lot of money. The reason I say that is because any kind of venture involves going through difficult times. If you're doing something you are passionate about and really believe in, then that will carry you through," - Jerry Greenfield.

"An investment in knowledge pays the best interest." - Benjamin Franklin

"Bottoms in the investment world don't end with four-year lows; they end with 10- or 15-year lows." - Jim Rogers

"I will tell you how to become rich. Close the doors. Be fearful when others are greedy. Be greedy when others are fearful." - Warren Buffett

"The stock market is filled with individuals who know the price of everything, but the value of nothing." - Phillip Fisher

"In investing, what is comfortable is rarely profitable." - Robert Arnott

"How many millionaires do you know who have become wealthy by investing in savings accounts? I rest my case." - Robert G. Allen

"Invest in yourself. Your career is the engine of your wealth." - Paul Clitheroe

Principal Emerging Bluechip Fund - Growth

L&T Hybrid Equity Fund

Franklin India Equity Fund(G)

Reliance Pharma Fund

Fig 6.4: Shares view in the website

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html" charset=UTF-8">
<title>JSzd P Page</title>
</head>
<body>
<%@ page import ="java.sql.*"%>
<%@ page import="javax.sql.*"%>
<%
String name=request.getParameter("name");
String imgurl=request.getParameter("imgurl");
String wurl=request.getParameter("wurl");

Class.forName("com.mysql.jdbc.Driver").newInstance();
java.sql.Connection
con=DriverManager.getConnection("jdbc:mysql://localhost:3306/invest_today?useSSL=false",
"root", "root");

String qr="insert into shares(name,imgurl,wurl) values(?,?,?)";
PreparedStatement pst=con.prepareStatement(qr);
pst.setString(1,name);
pst.setString(2,imgurl);
pst.setString(3,wurl);
pst.executeUpdate();
con.close();
pst.close();
//out.println("you data has inserted into the table.");
response.sendRedirect("shares.html");

%>
</body>
</html>
```