# “Accounting System for small scale businesses”

## J-Component PROJECT REPORT

**Submitted for the course: CSE3002 Internet and Web Programming**

*By*

**Kartikay Gupta, 18BCE2199**
**Tanishq Padwal, 18BCE2237**
**Naveen Nandakumar, 18BCE2316**

**Slot: B2**
**Name of faculty: Dr. K. Jayakumar**

**SCHOOL OF COMPUTER SCIENCE AND ENGINEERING**

VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

**November, 2020**

# CERTIFICATE

This is to certify that the project work entitled “Accounting system for small scale bussiness” that is being submitted by “Kartikay Gupta (18BCE2199), Tanishq Padwal(18BCE2237), Naveen Nandakumar (18BCE2316)” for CSE3002 Internet and Web Programming is a record of bonafide work done under my supervision. The contents of this Project work, in full or in parts, have neither been taken from any other source nor have been submitted for any other CAL course.

Place: Vellore

Date:01/11/2020

Signature of Student: KARTIKAY GUPTA, NAVEEN NANDAKUMAR, TANISHQ PADWAL

Signature of Faculty:

# ACKNOWLEDGEMENTS

I take immense pleasure in thanking Dr. G. Viswanathan, my beloved Chancellor, VIT University and respected Dean, Dr. R. Saravanan, for having permitted me to carry out the project.

I express gratitude to my guide, Dr. K. Jayakumar, for guidance and suggestions that helped me to complete the project on time. Words are inadequate to express my gratitude to the faculty and staff members who encouraged and supported me during the project. Finally, I would like to thank my ever-loving parents for their blessings and my friends for their timely help and support.

Signature of Student

Kartikay Gupta

Naveen Nandakumar

Tanishq Padwal

# TABLE OF CONTENTS

| | |
|---|---|
| **Abstract** | **5** |
| **Introduction** | **6** |
| **Existing System** | **7** |
| **Technologies learnt** | **8** |
| **Proposed System Design**<br>**Module Description (UML diagrams )** | **10** |
| **Results (dB design, Table Schemas and Implementation Screenshots)** | **21** |
| **Conclusion** | **37** |
| **References** | **39** |

# 1. Abstract

The accounting systems in place for small enterprises vary a lot. There are cases when there are no accounting requirements at all and cases where the accounting requirements are relatively strict for small enterprises. However, in practical terms, all small enterprises will need to keep some kind of financial records in order to keep financial control over their businesses.

The objective of this project is to provide views on how to improve the accounting systems of the small enterprises so that they can provide the owners/managers with appropriate financial information. This can be achieved through the identification and exchange of views in the area of accounting systems of small enterprises. As a result of the project, descriptions of accounting systems, guidance and good practices in the accounting area for small enterprises will be delivered, to accumulate theoretical knowledge with practical experience, to address and understand accounting procedures of a company, to recommend some issues regarding accounting system of the company.

# 2. Introduction

Accounting System for small scale businesses is a web enabled system designed to manage the Accounts of the seller. Accounting System for small scale businesses is a simple yet powerful one joint integrated platform that manage the entire Operations of the billing of a company to other.

It enables the users to store, modify and retrieve information of the company business with other company. The Accounting System for small scale businesses is also a utilitarian tool which enables the user to access their details/summary using their Username and Password.

## 2.1 Problem Statement

The objective of this project is to provide views on how to improve the accounting systems of the small enterprises so that they can provide the owners/managers with appropriate financial information. This can be achieved through the identification and exchange of views in the area of accounting systems of small enterprises. As a result of the project, descriptions of accounting systems, guidance and good practices in the accounting area for small enterprises will be delivered, to accumulate theoretical knowledge with practical experience, to address and understand accounting procedures of a company, to recommend some issues regarding accounting system of the company.

## 2.2 Technical details

### *2.2.1 User Interfaces*

Front-end: HTML, CSS, JavaScript and jQuery
Back-end: Nodejs
Database: mongo dB

### *2.2.2 Hardware Interfaces*

Operating System: Windows xp and above
Processor: Pentium 4 or higher
RAM: 512 MB or more
Hard Drive: 10 GB or more

# 3. Existing system

### EXISTING SYSTEM:

Accounting software is software used by businesses to track financial statements, cash flow, invoicing, bank accounts, and purchase orders. These systems occasionally come with specialized functionality for particular business sizes, the self-employed or freelancers, small businesses or enterprises. Because the accounting solution serves a critical business process, it's often a central component of an organization's enterprise resource planning (ERP) system.

### DISADVANTAGES:

✓ Needs decent computer knowledge to setup.

✓ If dB goes corrupt no chance of recovery unless backup available .

✓ Might not have advanced functionalities like Paid softwares used by MNC’s and LSB’s.

✓ Might face technical issues related to dB.

### 3.3 PROPOSED SYSTEM:

✓ We propose a novel Accounting Software that is focussed on small scale businesses

who have limited needs when it comes to accounting. Our system is failproof and DB

secured as we use it locally.

✓ Has functionalities like party creation, invoice creation and company creation that help to store details for all so they don’t have to typed everyday or on every use.

✓ Will be free for use, only requirement is a PC with a stable internet connection.

✓ Will be more easier to maintain than physical invoices, also helpful in filing taxes andreturns.

## 3.4 ADVANTAGES:

✓ Free Software for small businesses, no hidden charges.

✓ Better and accurate than invoicing physically.

✓ Can be used for official works such as filing taxes and returns.

✓ Multiple functionalities, big market.

✓ Works in absence of internet connection.

## TECHNOLOGIES LEARNT

### ➢ HTML, CSS, JavaScript and jQuery

HTML, CSS, JavaScript and jQuery are superior languages for frontend development. HTML and CSS are used for structuring the website and making basic prototype. Javascript has been used for computing calculations and for making invoice, it is most well-known as the scripting language for Web pages, many non-browser environments also use it, such as Node.js . jQuery is a JavaScript library designed to simplify HTML DOM tree traversal and manipulation, as well as event handling, CSS animation, and Ajax.

Typical uses are

- Frontend development
- Declaring functions
- Styling UI
- Authentication and Validation
- Linking to backend.

### MongoDB and NodeJS

MongoDB is a cross-platform, document oriented database that provides, high performance, high availability, and easy scalability. MongoDB works on concept of collection and document. Database is a physical container for collections. Each database gets its own set of files on the file system. A single MongoDB server typically has multiple databases. Node.js is a server-side platform built on Google Chrome's JavaScript Engine (V8 Engine). for easily building fast and scalable network applications. Node.js uses an event-driven non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices.

Node.js is an open source, cross-platform runtime environment for developing server-side and networking applications. Node.js applications are written in JavaScript, and can be run within the Node.js runtime on OS X, Microsoft Windows, and Linux.

**Typical Uses :**

**NodeJS**

- I/O bound Applications
- Data Streaming Applications
- Data Intensive Real-time Applications (DIRT)
- JSON APIs based Applications
- Single Page Applications

**MongoDB :**

- Schema less − MongoDB is a document database in which one collection holds

different documents. Number of fields, content and size of the document can differ

from one document to another.

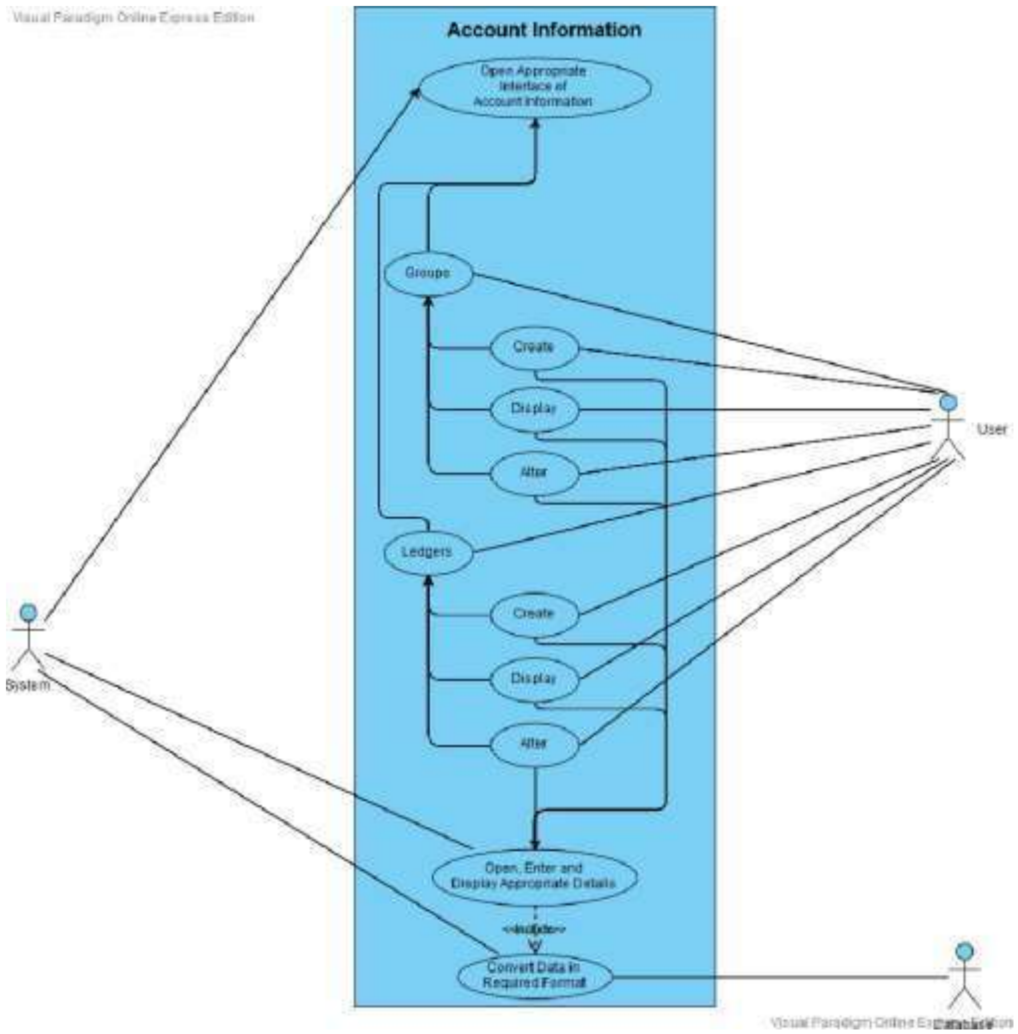
- Structure of a single object is clear.
- No complex joins.
- Deep query-ability. MongoDB supports dynamic queries on documents using a document-based query language that's nearly as powerful as SQL.
- Tuning.
- Ease of scale-out − MongoDB is easy to scale.
- Conversion/mapping of application objects to database objects not needed.
- Uses internal memory for storing the (windowed) working set, enabling faster

access of data

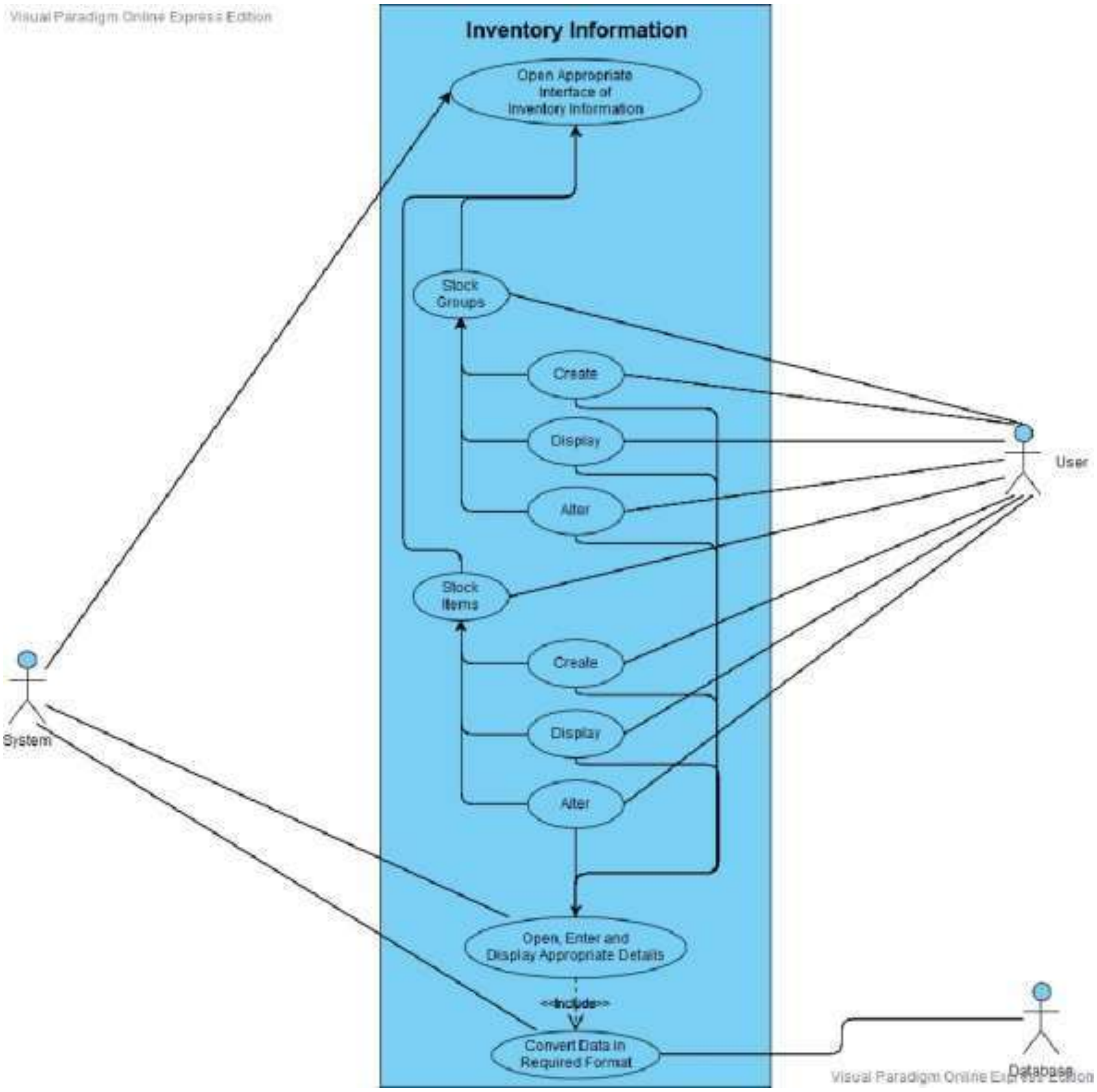
# 4. Proposed System Design

## 4.1 Use Case Diagram

### 4.1.1 For Account Information

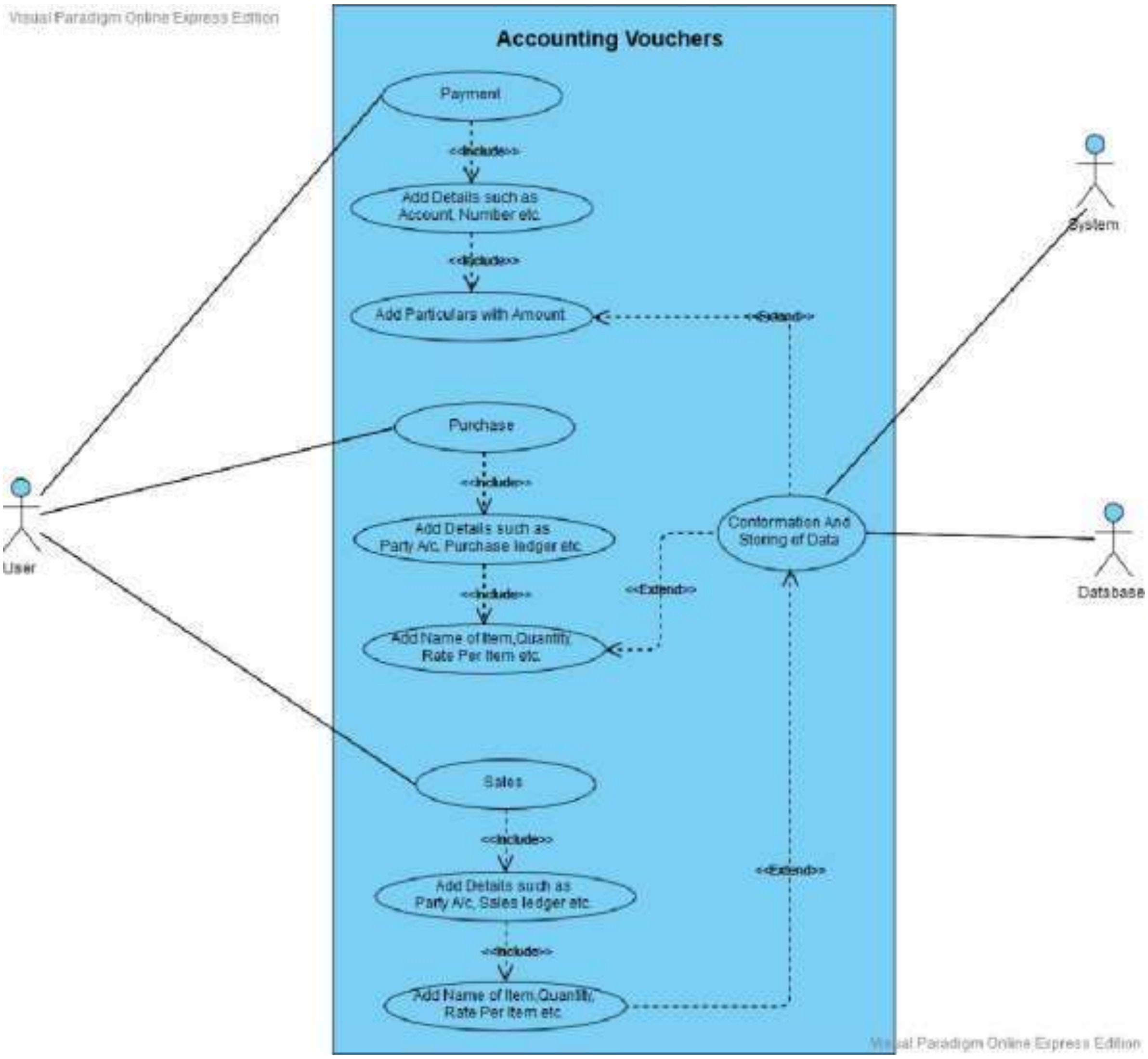

Visual Paradigm Online Express Edition
Account Information
Open Appropriate Interface of Account Information
Groups
Create
Display
Alter
Ledgers
Create
Display
Alter
Open, Enter and Display Appropriate Details
<<Include>>
Convert Data in Required Format
User
System
Database
Visual Paradigm Online Express Edition

### 4.1.2 For Inventory Information

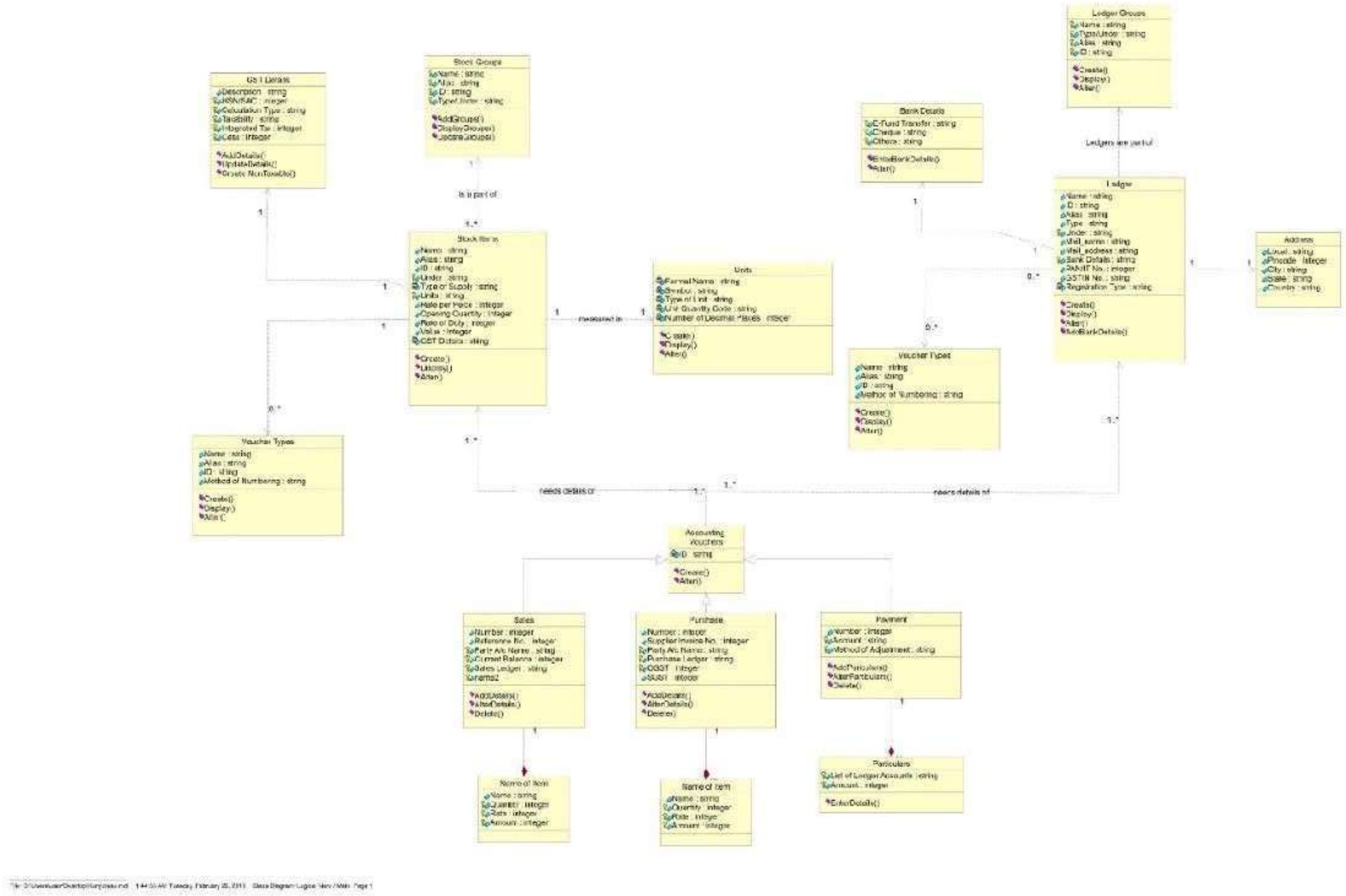


Visual Paradigm Online Express Edition
Inventory Information
Open Appropriate Interface of Inventory Information
Stock Groups
Create
Display
Alter
Stock Items
Create
Display
Alter
Open, Enter and Display Appropriate Details
<<Include>>
Convert Data in Required Format
System
User
Database
Visual Paradigm Online Express Edition

### 4.1.3 For Accounting Vouchers

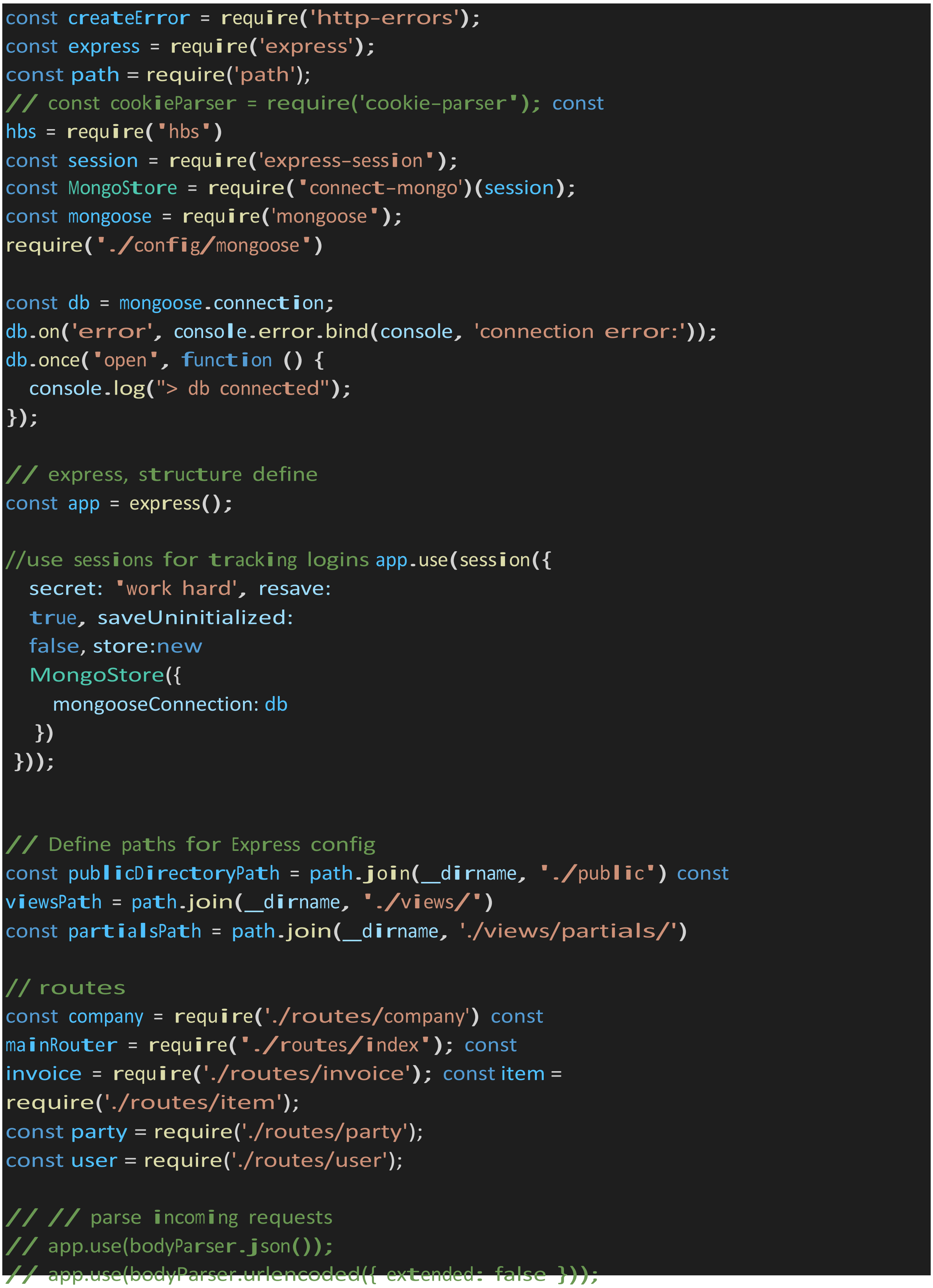

Visual Paradigm Online Express Edition
Accounting Vouchers
Payment
<<Include>>
Add Details such as Account, Number etc.
<<Include>>
Add Particulars with Amount
<<Extend>>
Purchase
<<Include>>
Add Details such as Party A/c, Purchase ledger etc.
<<Include>>
Add Name of Item,Quantity, Rate Per Item etc.
<<Extend>>
Conformation And Storing of Data
Sales
<<Include>>
Add Details such as Party A/c, Sales ledger etc.
<<Include>>
Add Name of Item,Quantity, Rate Per Item etc.
<<Extend>>
User
System
Database
Visual Paradigm Online Express Edition

## 4.2 Class Diagram:

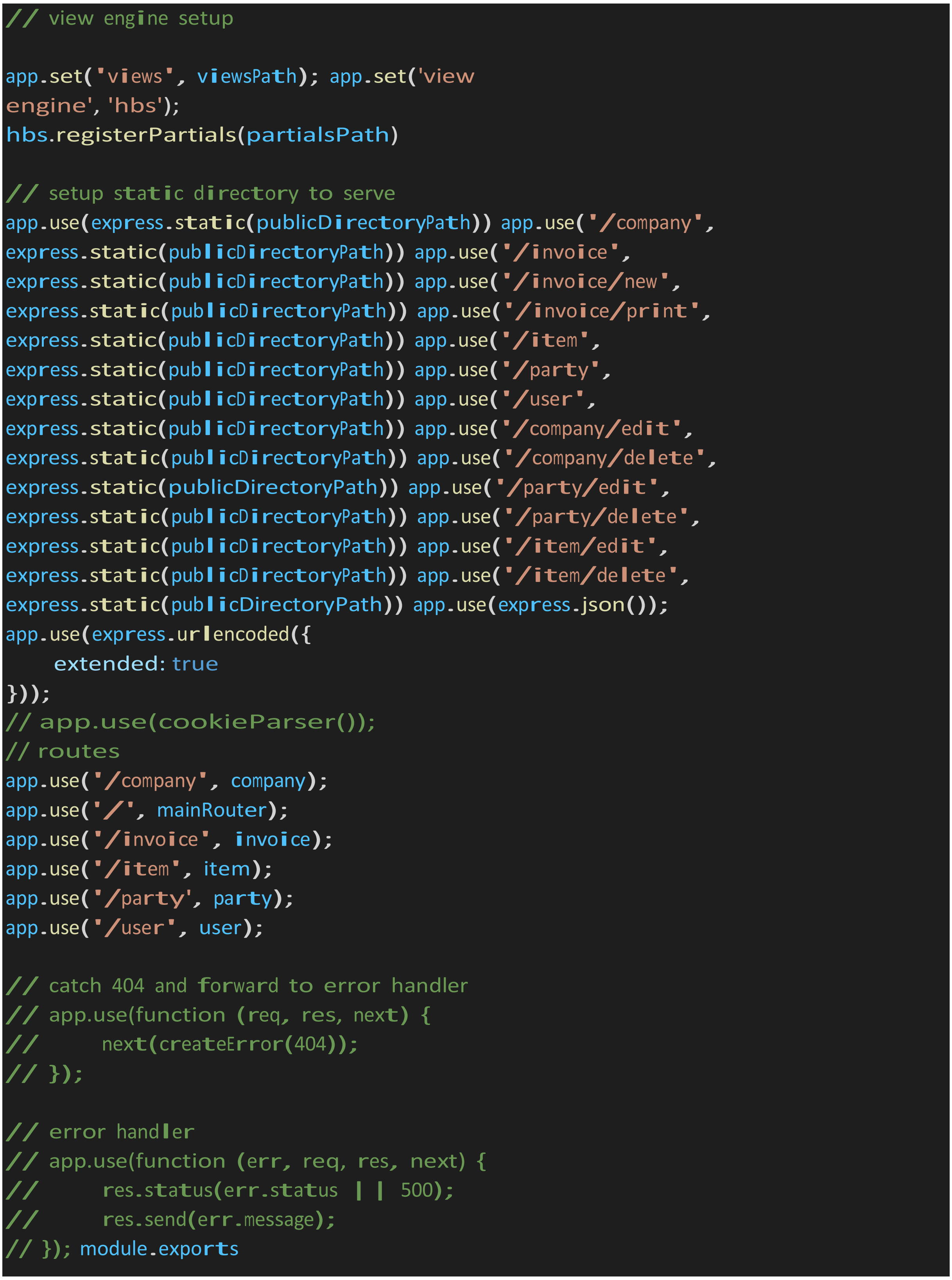


## 4.3 Sample Code’s

### App.js



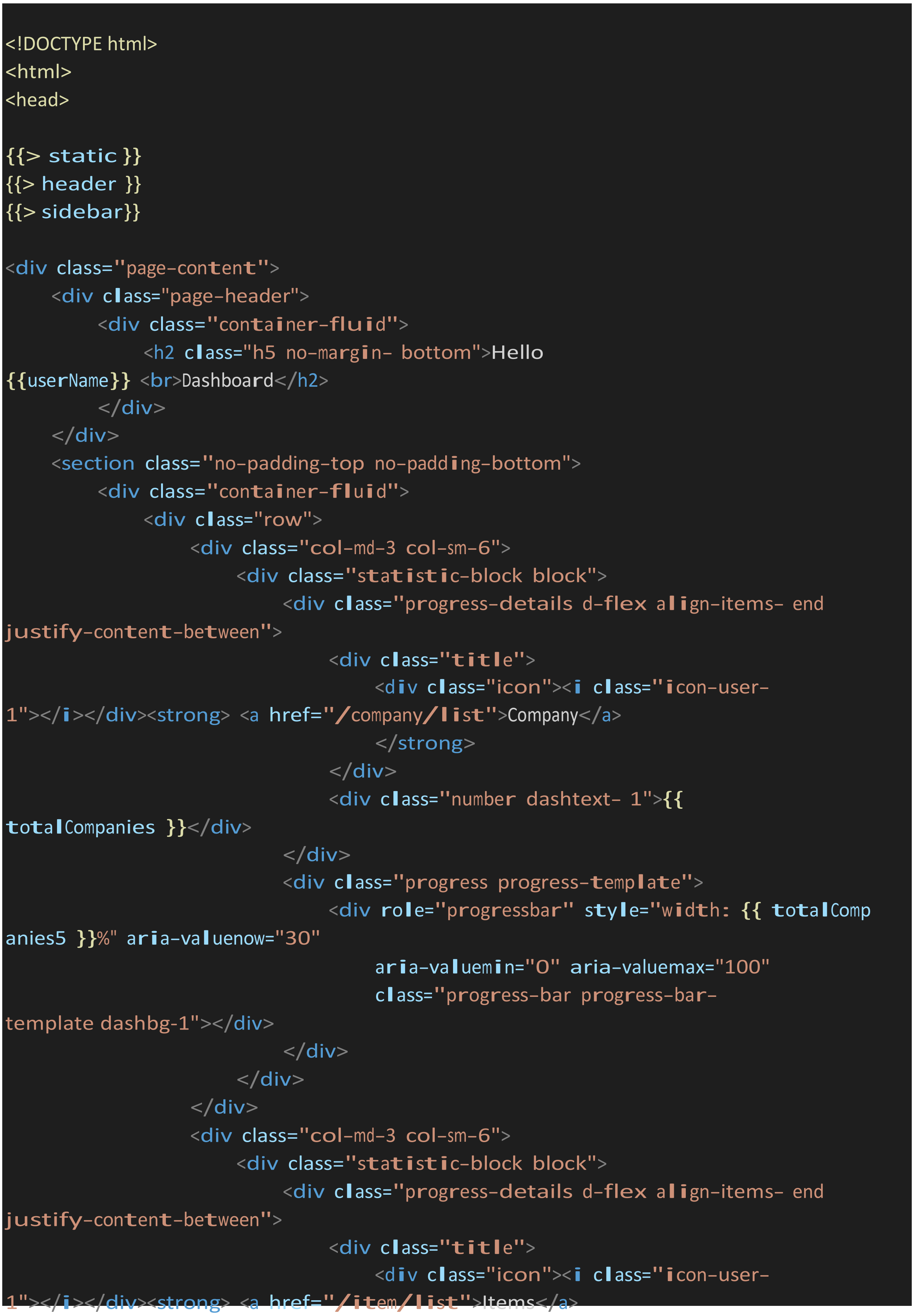
```
const createError = require('http-errors');
const express = require('express');
const path = require('path');
// const cookieParser = require('cookie-parser'); const
hbs = require('hbs')
const session = require('express-session');
const MongoStore = require('connect-mongo')(session);
const mongoose = require('mongoose');
require('./config/mongoose')

const db = mongoose.connection;
db.on('error', console.error.bind(console, 'connection error:'));
db.once('open', function () {
  console.log("> db connected");
});

// express, structure define
const app = express();

//use sessions for tracking logins app.use(session({
  secret: 'work hard', resave:
  true, saveUninitialized:
  false, store:new
  MongoStore({
    mongooseConnection: db
  })
 }));


// Define paths for Express config
const publicDirectoryPath = path.join(__dirname, './public') const
viewsPath = path.join(__dirname, './views/')
const partialsPath = path.join(__dirname, './views/partials/')

// routes
const company = require('./routes/company') const
mainRouter = require('./routes/index'); const
invoice = require('./routes/invoice'); const item =
require('./routes/item');
const party = require('./routes/party');
const user = require('./routes/user');

// // parse incoming requests
// app.use(bodyParser.json());
// app.use(bodyParser.urlencoded({ extended: false }));
```



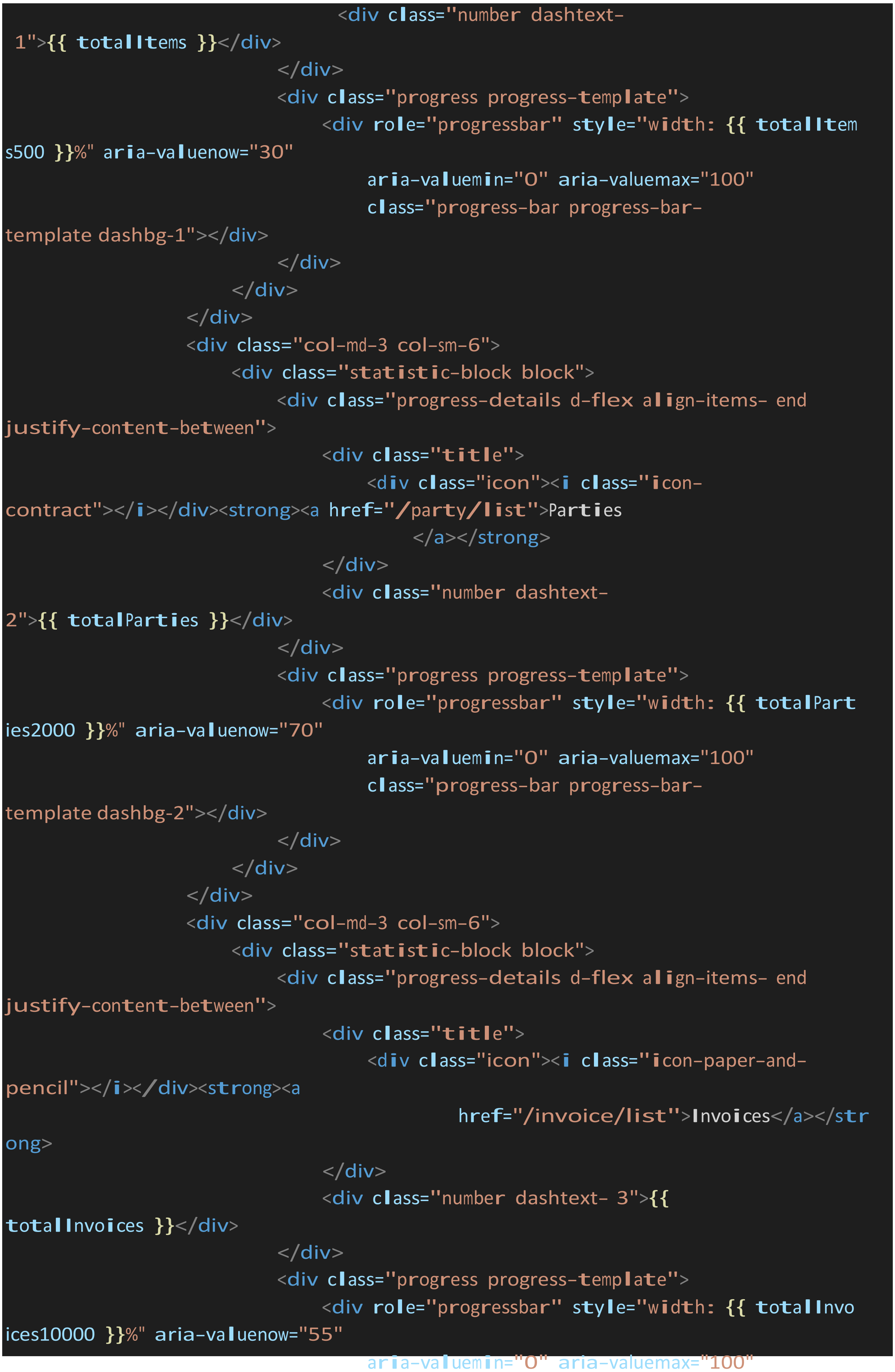
```
// view engine setup

app.set('views', viewsPath); app.set('view
engine', 'hbs');
hbs.registerPartials(partialsPath)

// setup static directory to serve
app.use(express.static(publicDirectoryPath)) app.use('/company',
express.static(publicDirectoryPath)) app.use('/invoice',
express.static(publicDirectoryPath)) app.use('/invoice/new',
express.static(publicDirectoryPath)) app.use('/invoice/print',
express.static(publicDirectoryPath)) app.use('/item',
express.static(publicDirectoryPath)) app.use('/party',
express.static(publicDirectoryPath)) app.use('/user',
express.static(publicDirectoryPath)) app.use('/company/edit',
express.static(publicDirectoryPath)) app.use('/company/delete',
express.static(publicDirectoryPath)) app.use('/party/edit',
express.static(publicDirectoryPath)) app.use('/party/delete',
express.static(publicDirectoryPath)) app.use('/item/edit',
express.static(publicDirectoryPath)) app.use('/item/delete',
express.static(publicDirectoryPath)) app.use(express.json());
app.use(express.urlencoded({
    extended: true
}));
// app.use(cookieParser());
// routes
app.use('/company', company);
app.use('/', mainRouter);
app.use('/invoice', invoice);
app.use('/item', item);
app.use('/party', party);
app.use('/user', user);

// catch 404 and forward to error handler
// app.use(function (req, res, next) {
//      next(createError(404));
// });

// error handler
// app.use(function (err, req, res, next) {
//      res.status(err.status || 500);
//      res.send(err.message);
// }); module.exports
```

```
= app;
```

## Login page: home.hbs

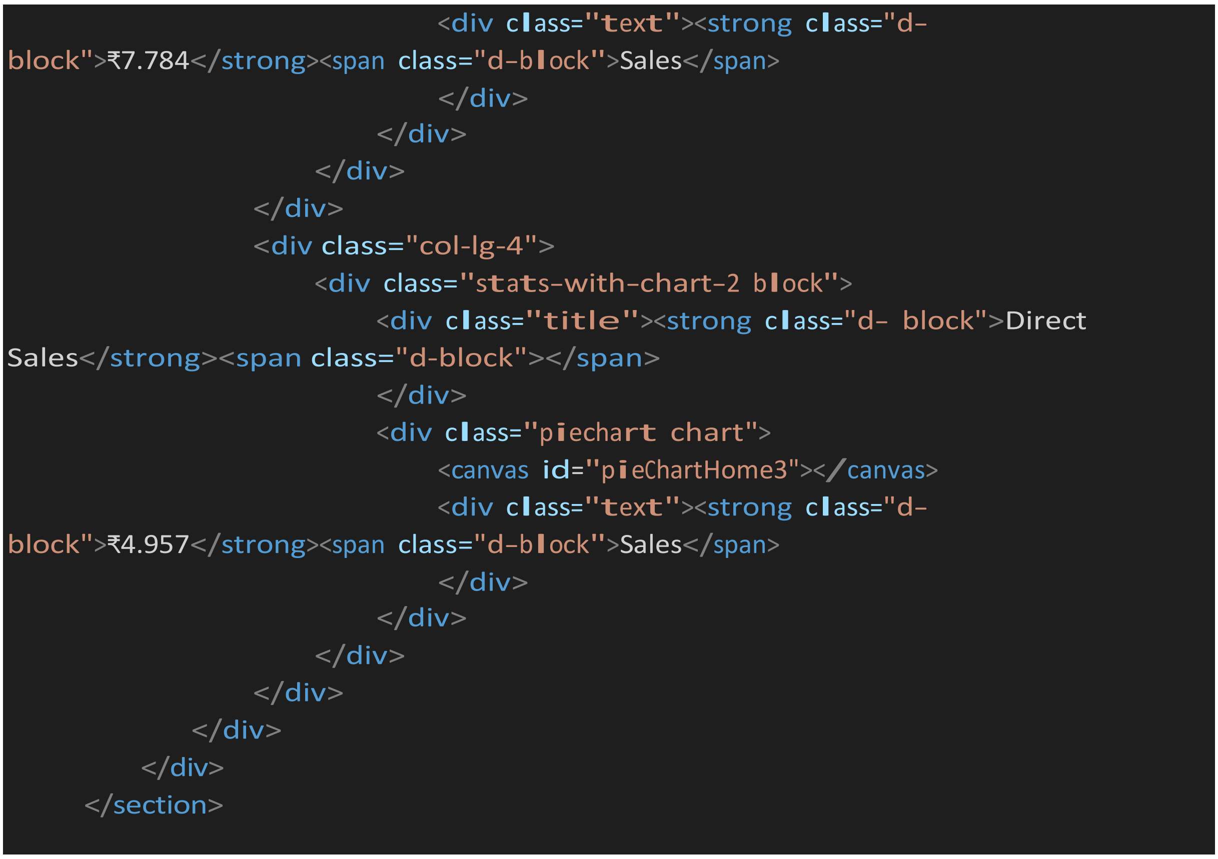


```
<!DOCTYPE html>
<html>
<head>

{{> static }}
{{> header }}
{{> sidebar}}

<div class="page-content">
     <div class="page-header">
          <div class="container-fluid">
               <h2 class="h5 no-margin- bottom">Hello
{{userName}} <br>Dashboard</h2>
          </div>
     </div>
     <section class="no-padding-top no-padding-bottom">
          <div class="container-fluid">
               <div class="row">
                    <div class="col-md-3 col-sm-6">
                         <div class="statistic-block block">
                              <div class="progress-details d-flex align-items- end
justify-content-between">
                                   <div class="title">
                                        <div class="icon"><i class="icon-user-
1"></i></div><strong> <a href="/company/list">Company</a>
                                        </strong>
                                   </div>
                                   <div class="number dashtext- 1">{{
totalCompanies }}</div>
                              </div>
                              <div class="progress progress-template">
                                   <div role="progressbar" style="width: {{ totalComp
anies5 }}%" aria-valuenow="30"
                                        aria-valuemin="0" aria-valuemax="100"
                                        class="progress-bar progress-bar-
template dashbg-1"></div>
                              </div>
                         </div>
                    </div>
                    <div class="col-md-3 col-sm-6">
                         <div class="statistic-block block">
                              <div class="progress-details d-flex align-items- end
justify-content-between">
                                   <div class="title">
                                        <div class="icon"><i class="icon-user-
1"></i></div><strong> <a href="/item/list">Items</a>
                                        </strong>
                                   </div>
```



```
                             <div class="number dashtext-
 1">{{ totalItems }}</div>
                        </div>
                        <div class="progress progress-template">
                            <div role="progressbar" style="width: {{ totalItem
s500 }}%" aria-valuenow="30"
                                aria-valuemin="0" aria-valuemax="100"
                                class="progress-bar progress-bar-
template dashbg-1"></div>
                        </div>
                    </div>
                </div>
                <div class="col-md-3 col-sm-6">
                    <div class="statistic-block block">
                        <div class="progress-details d-flex align-items- end
justify-content-between">
                            <div class="title">
                                <div class="icon"><i class="icon-
contract"></i></div><strong><a href="/party/list">Parties
                                    </a></strong>
                            </div>
                            <div class="number dashtext-
2">{{ totalParties }}</div>
                        </div>
                        <div class="progress progress-template">
                            <div role="progressbar" style="width: {{ totalPart
ies2000 }}%" aria-valuenow="70"
                                aria-valuemin="0" aria-valuemax="100"
                                class="progress-bar progress-bar-
template dashbg-2"></div>
                        </div>
                    </div>
                </div>
                <div class="col-md-3 col-sm-6">
                    <div class="statistic-block block">
                        <div class="progress-details d-flex align-items- end
justify-content-between">
                            <div class="title">
                                <div class="icon"><i class="icon-paper-and-
pencil"></i></div><strong><a
                                        href="/invoice/list">Invoices</a></str
ong>
                            </div>
                            <div class="number dashtext- 3">{{
totalInvoices }}</div>
                        </div>
                        <div class="progress progress-template">
                            <div role="progressbar" style="width: {{ totalInvo
ices10000 }}%" aria-valuenow="55"
                                aria-valuemin="0" aria-valuemax="100"
                                class="progress-bar progress-bar-
template dashbg-3"></div>
```

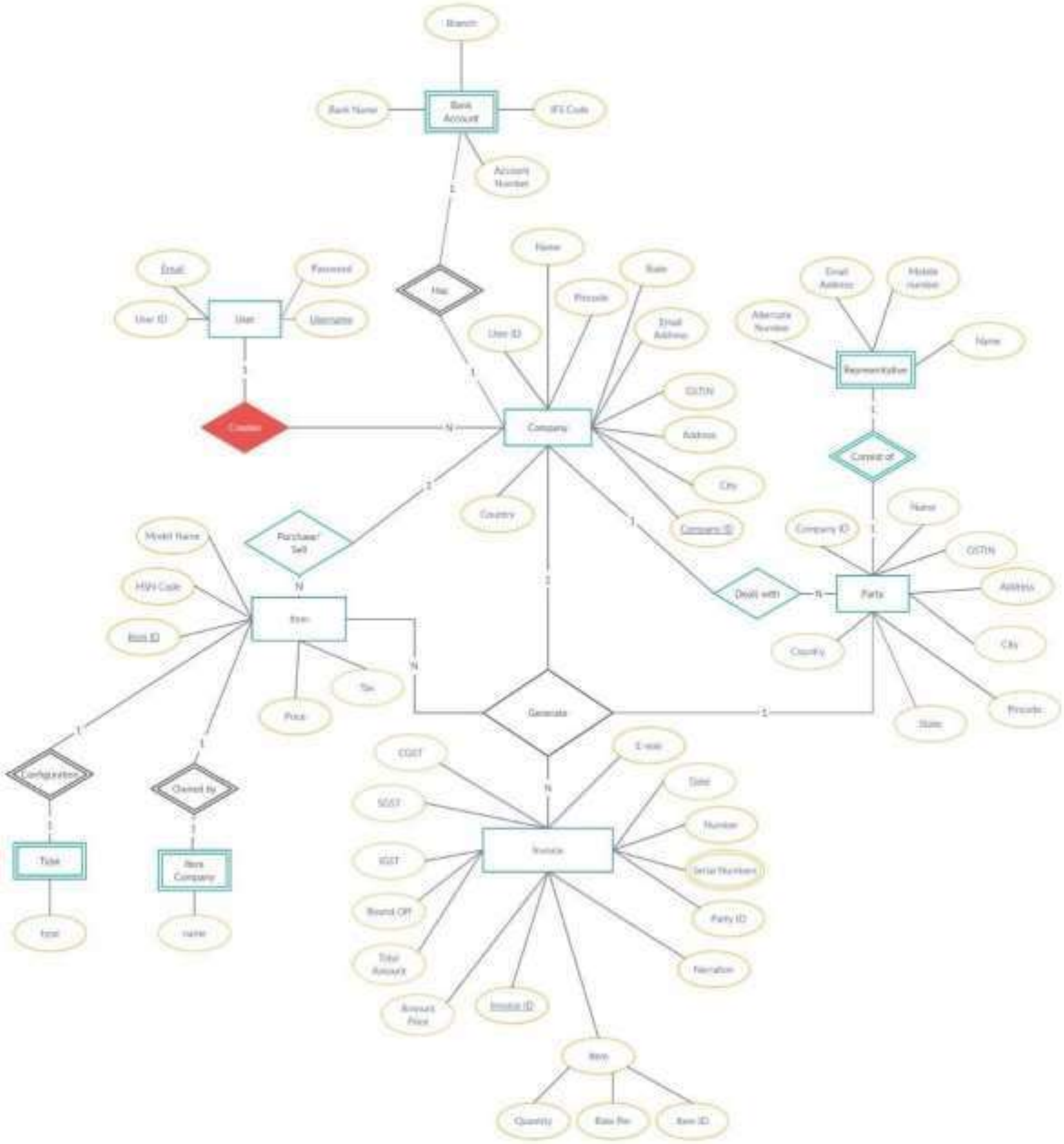
```
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </section>


    <section class="margin-bottom-sm">
        <div class="container-fluid">
            <div class="row d-flex align-items-stretch">
                <div class="col-lg-4">
                    <div class="stats-with-chart-1 block">
                        <div class="title"> <strong class="d-
block">Sales Difference</strong><span class="d-block">Lorem
                                ipsum dolor sit</span></div>
                        <div class="row d-flex align-items-end justify-
content-between">
                            <div class="col-5">
                                <div class="text"><strong class="d-
  block dashtext-3">₹740</strong><span
                                        class="d-
block">May 2017</span><small class="d-block">320 Sales</small></div>
                            </div>
                            <div class="col-7">
                                <div class="bar-chart chart">
                                    <canvas id="salesBarChart1"></canvas>
                                </div>
                            </div>
                        </div>
                    </div>
                </div>
                <div class="col-lg-4">
                    <div class="stats-with-chart-1 block">
                        <div class="title"> <strong class="d-
block">Visit Statistics</strong><span class="d-block">Lorem
                                ipsum dolor sit</span></div>
                        <div class="row d-flex align-items-end justify-
content-between">
                            <div class="col-4">
                                <div class="text"><strong class="d-
  block dashtext-1">₹457</strong><span
                                        class="d-
block">May 2017</span><small class="d-block">210 Sales</small></div>
                            </div>
                            <div class="col-8">
                                <div class="bar-chart chart">
                                    <canvas id="visitPieChart"></canvas>
                                </div>
                            </div>
                        </div>
                    </div>
                </div>
                <div class="col-lg-4">
                    <div class="stats-with-chart-1 block">
                        <div class="title"> <strong class="d-
block">Sales Activities</strong><span class="d-block">Lorem
                                ipsum dolor sit</span></div>
                        <div class="row d-flex align-items-end justify-
content-between">
                            <div class="col-5">
                                <div class="text"><strong class="d-
 block dashtext-2">80%</strong><span
                                          class="d-
block">May 2017</span><small class="d-block">+35 Sales</small></div>
                            </div>
                            <div class="col-7">
                                <div class="bar-chart chart">
                                    <canvas id="salesBarChart2"></canvas>
                                </div>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </section>



    <section>
        <div class="container-fluid">
            <div class="row">
                <div class="col-lg-4">
                    <div class="stats-with-chart-2 block">
                        <div class="title"><strong class="d-
block">Credit Sales</strong><span class="d-block"></span>
                        </div>
                        <div class="piechart chart">
                            <canvas id="pieChartHome1"></canvas>
                            <div class="text"><strong class="d-
block">₹2.145</strong><span class="d-block">Sales</span>
                            </div>
                        </div>
                    </div>
                </div>
                <div class="col-lg-4">
                    <div class="stats-with-chart-2 block">
                        <div class="title"><strong class="d-
block">Channel Sales</strong><span class="d-block"></span>
                        </div>
                        <div class="piechart chart">
                            <canvas id="pieChartHome2"></canvas>
```



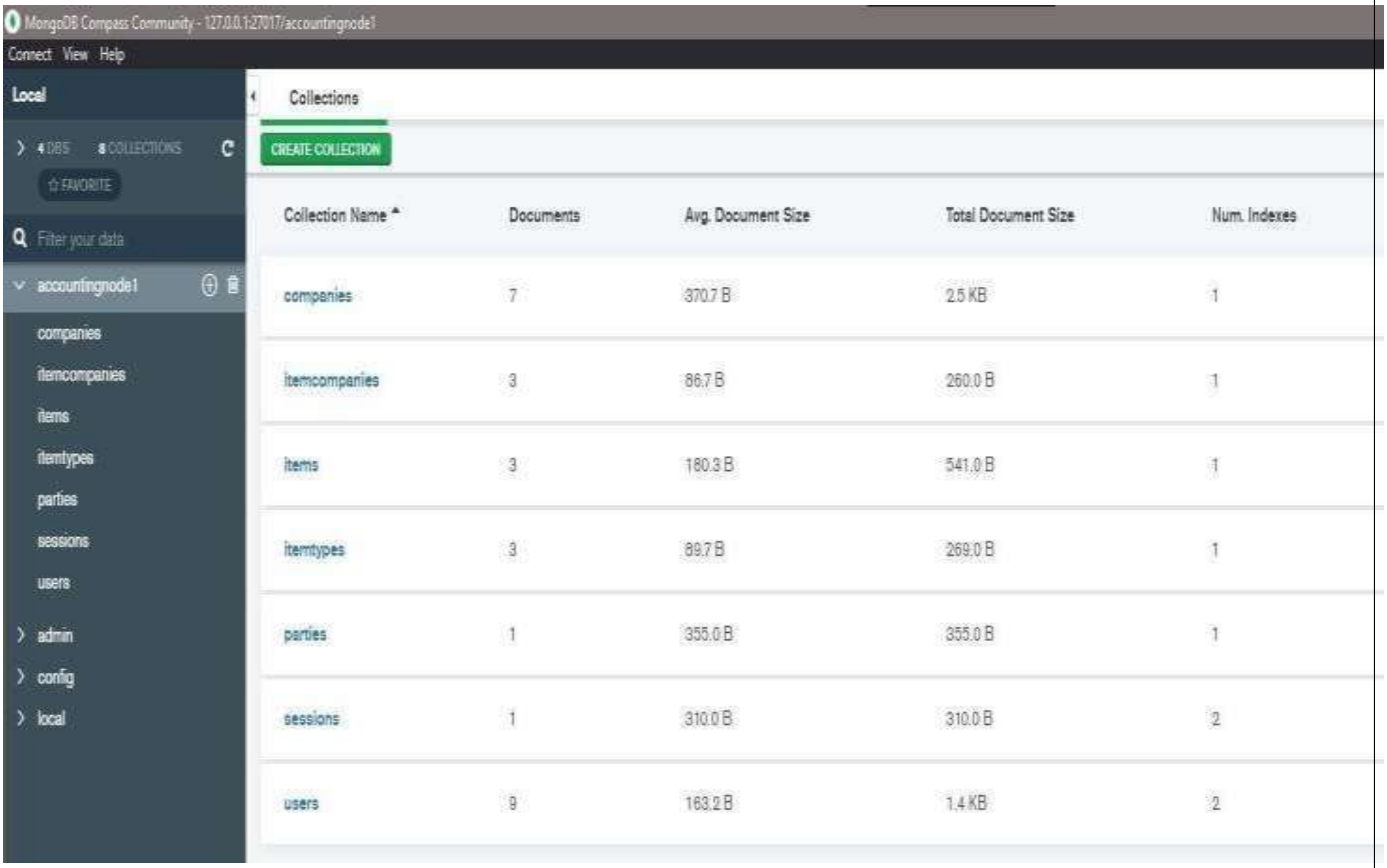
```
                            <div class="text"><strong class="d-
block">₹7.784</strong><span class="d-block">Sales</span>
                            </div>
                        </div>
                    </div>
                </div>
                <div class="col-lg-4">
                    <div class="stats-with-chart-2 block">
                        <div class="title"><strong class="d- block">Direct
Sales</strong><span class="d-block"></span>
                        </div>
                        <div class="piechart chart">
                            <canvas id="pieChartHome3"></canvas>
                            <div class="text"><strong class="d-
block">₹4.957</strong><span class="d-block">Sales</span>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </section>
```

# 5. Result

## 5.1 Database design



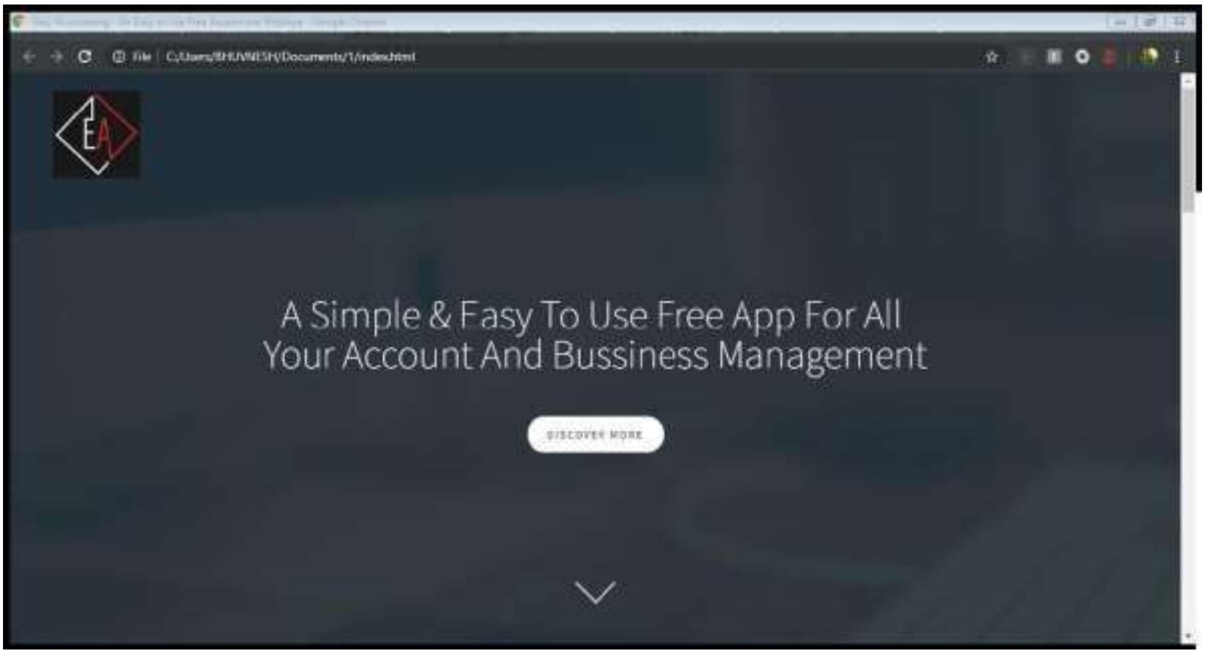
## 5.2 Table Schema



| Collection Name ▴ | Documents | Avg. Document Size | Total Document Size | Num. Indexes |
|---|---|---|---|---|
| companies | 7 | 370.7 B | 2.5 KB | 1 |
| itemcompanies | 3 | 86.7 B | 260.0 B | 1 |
| items | 3 | 180.3 B | 541.0 B | 1 |
| itemtypes | 3 | 89.7 B | 269.0 B | 1 |
| parties | 1 | 355.0 B | 355.0 B | 1 |
| sessions | 1 | 310.0 B | 310.0 B | 2 |
| users | 9 | 163.2 B | 1.4 KB | 2 |

## 5.3 Screenshot with explanation

### 5.3.1 Landing Page



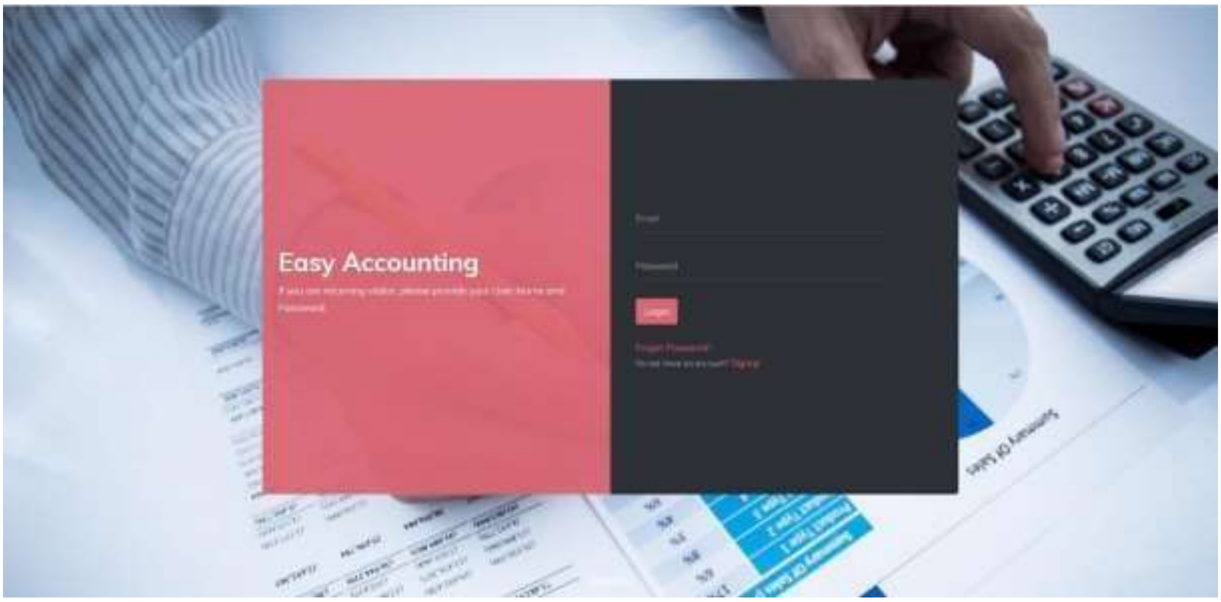
A Simple & Easy To Use Free App For All Your Account And Bussiness Management

Used to access main home page and make use of all functionalities, validation used where required.

Registration page with page will validation in required fields, registration compulsory for further signin.

All input and login details will be saved in json type on MongoDB, which will be encrypted and thus cannot be copied or stolen.

### 5.3.2 Login Page:



Easy Accounting

If you are returning visitor, please provide your User Name and Password.

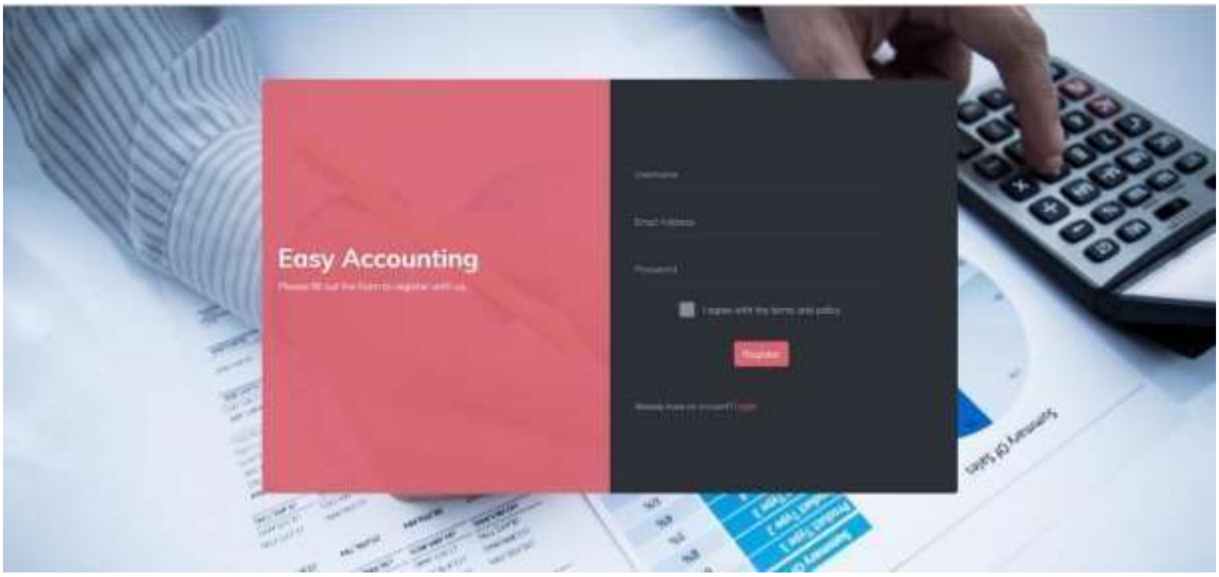
Used to access main home page and make use of all functionalities, validation used where required.

Used to access main home page and make use of all functionalities, validation used where required.

Registration page with page will validation in required fields, registration compulsory for further signin.

All input and login details will be saved in json type on MongoDB, which will be encrypted and thus cannot be copied or stolen.

### 5.3.3 Registration Page:


Easy Accounting
Please fill out the form to register with us.
Username
Email Address
Password
□ I agree with the terms and policy
Register
Already have an account? Login

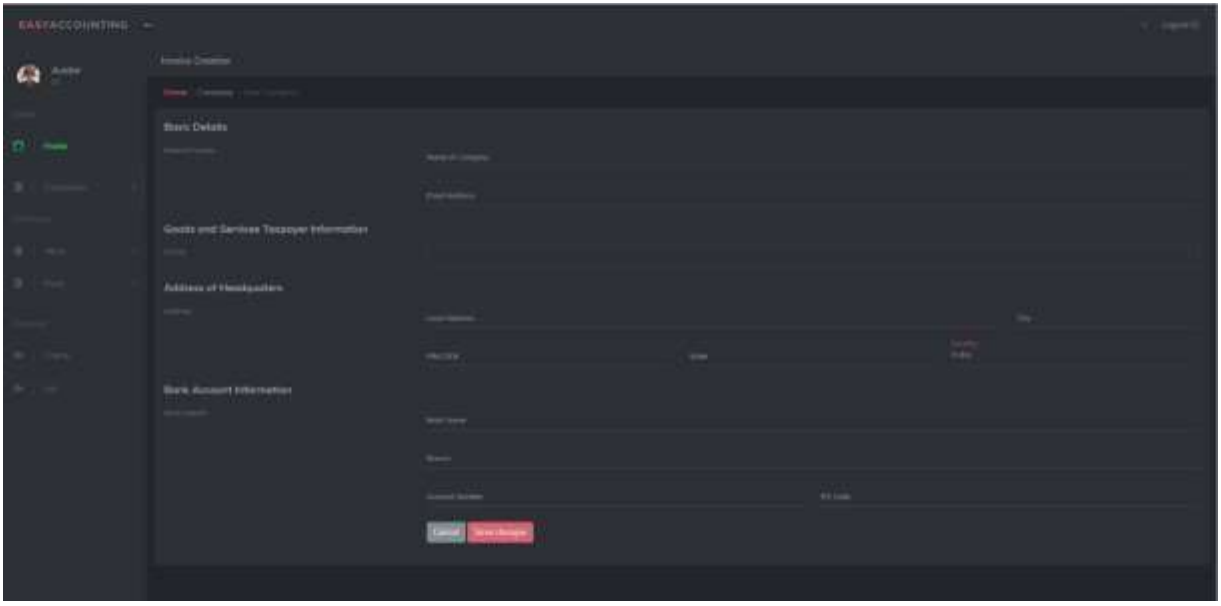
Registration page with page will validation in required fields, registration compulsory for further sign in.

Used to access main home page and make use of all functionalities, validation used where required.

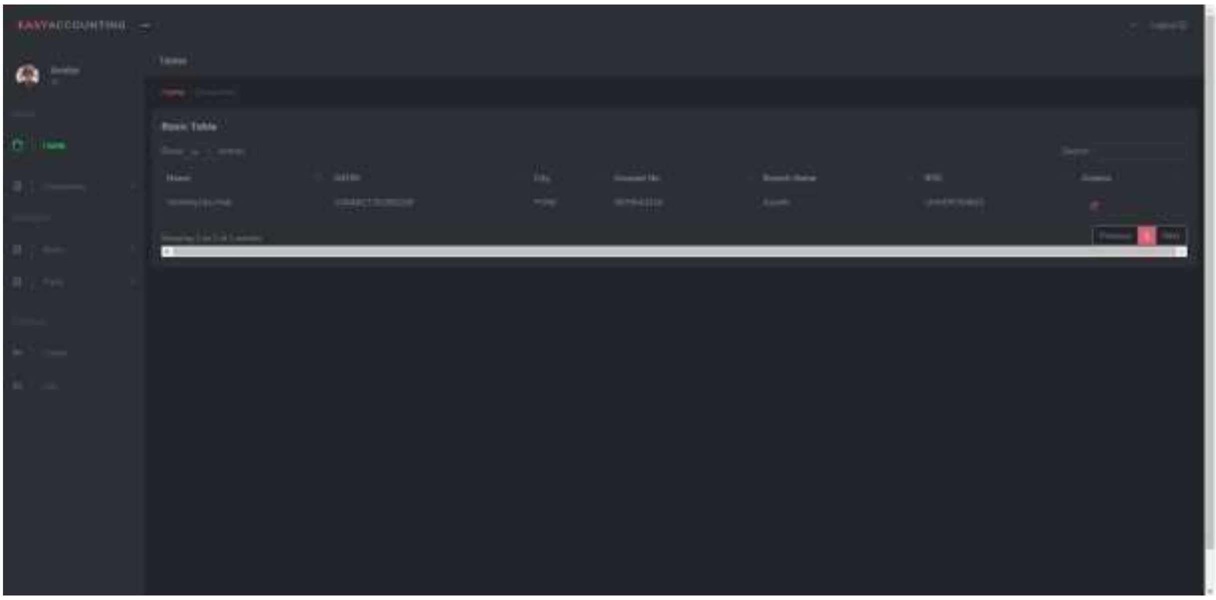
Registration page with page will validation in required fields, registration compulsory for further signin.

All input and login details will be saved in json type on MongoDB, which will be encrypted and thus cannot be copied or stolen.

### 5.3.4 Create Company:



Company lists :



Used to create a prebuilt profile for each company that keeps track of all information so that it doesnt

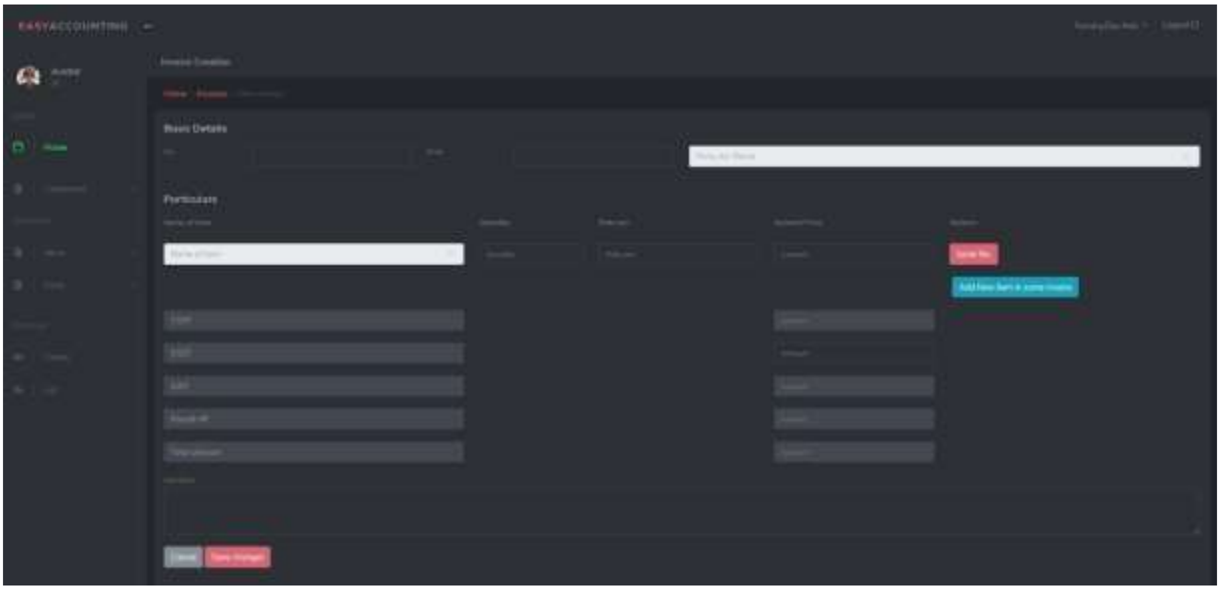
have to be entered on every order.

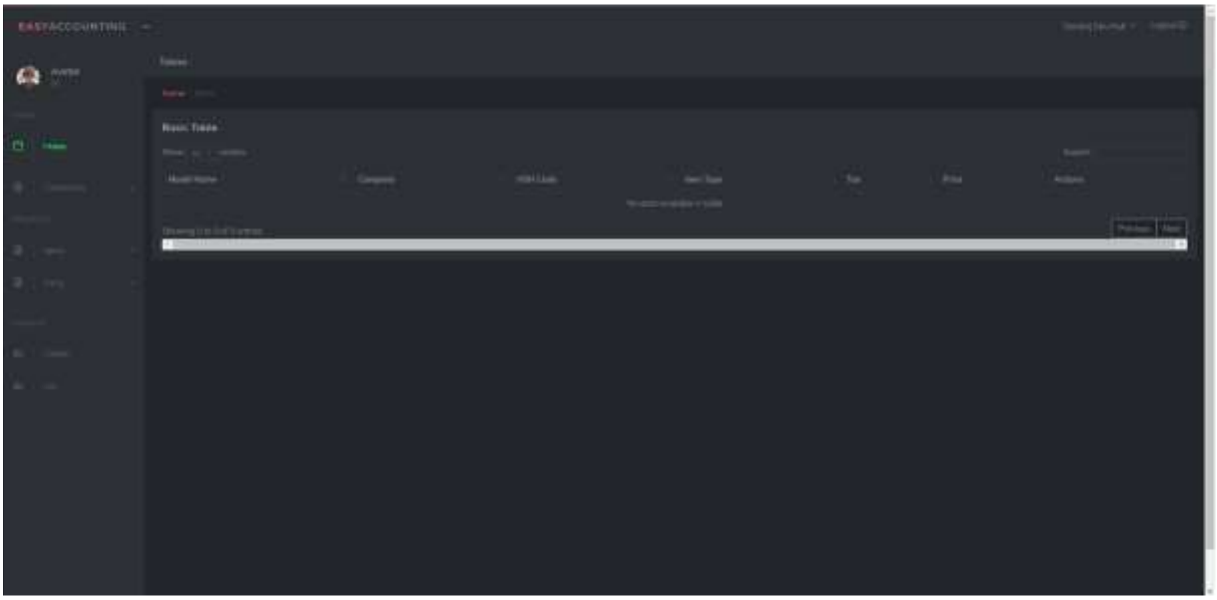
Modules used :

- sales and accounts receivable sub-system
- management information sub-system

All field inputs are stored and fetched from cluster as needed.

### 5.3.5 Invoice Creation:





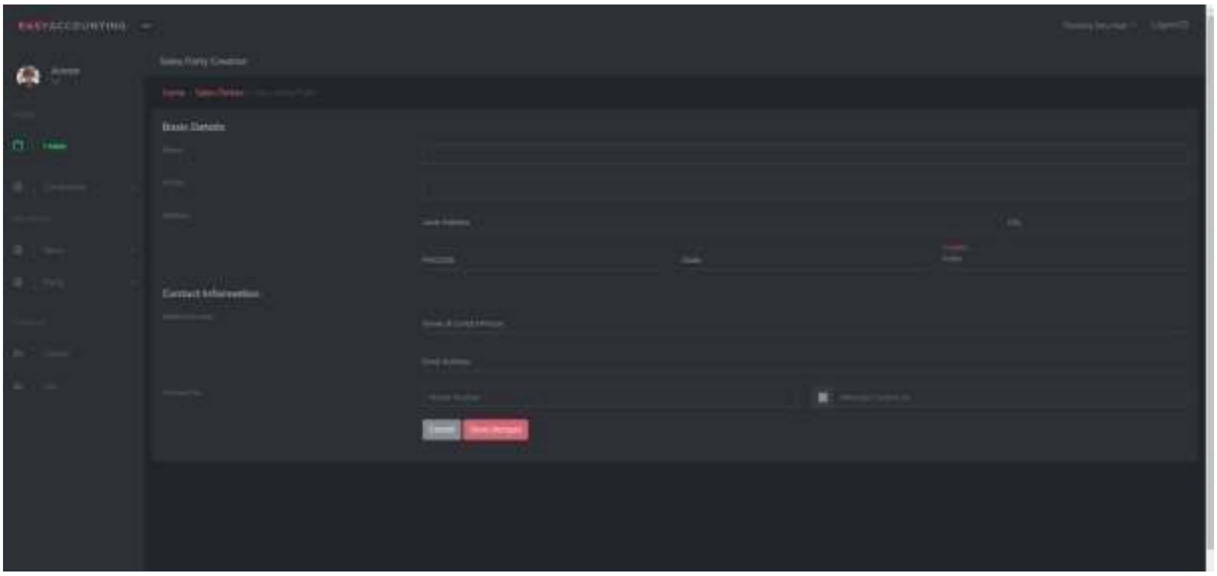
Used for creating invoice used for filing taxes or maintaining records of all debit and credit in account.

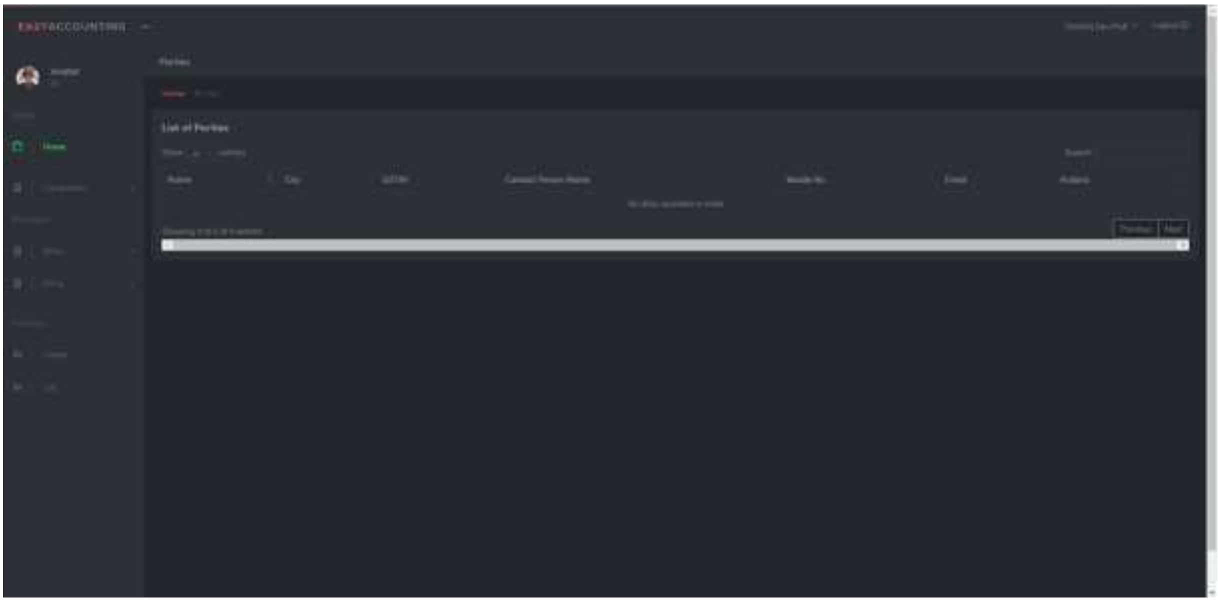
Modules :

- purchase and accounts payable sub-system,
- expense accounting sub-system,
- tax accounting sub- system,

All invoices dB will be fetched from companies,parties and items and then put together while generating invoice.

### 5.3.6 Sales Party:





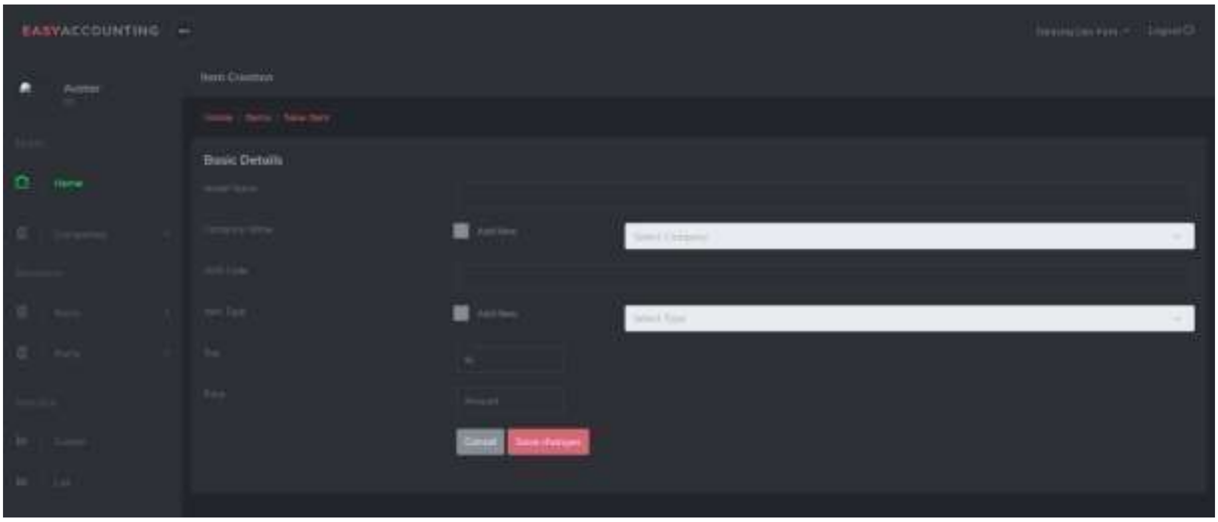
Used to store data of individuals to whom goods sold, includes data such as name, contact, email and

other financial info

Modules used :

- purchase and accounts payable sub-system
- management information sub-system

### 5.3.7 Inventory Management :



Used to keep track of all goods available, can be added or deleted as needed along with all pricing and

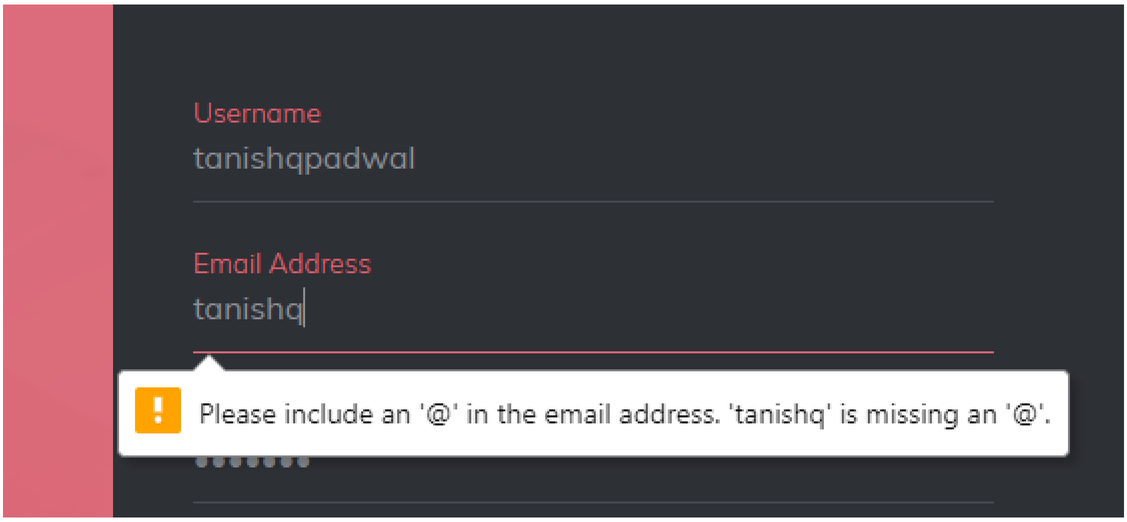
taxation.

Modules used :

- inventory sub-system
- expense accounting sub-system
- budget sub-system

All items are stored in a json sorted according to each input field.

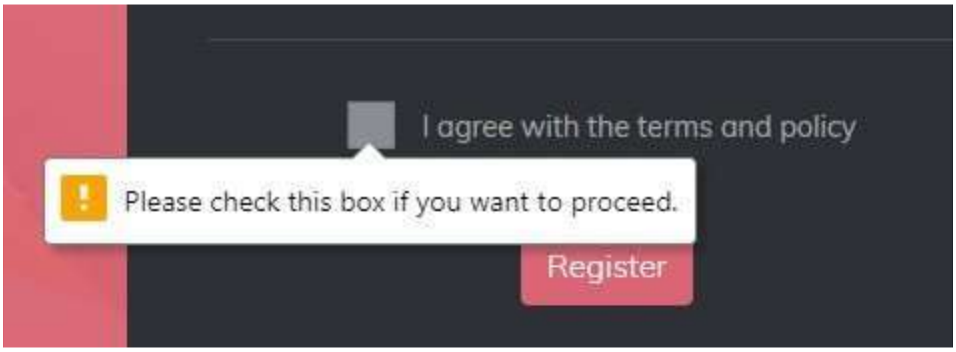
### 5.3.8 Validation:



Username
tanishqpadwal

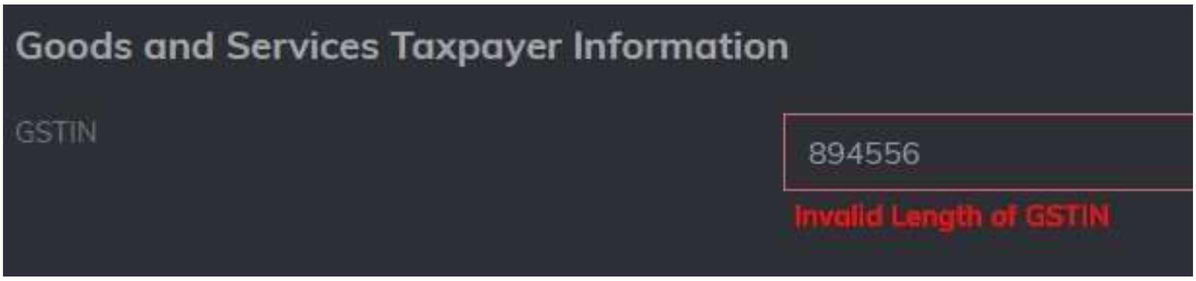
Email Address
tanishq

Please include an '@' in the email address. 'tanishq' is missing an '@'.



□ I agree with the terms and policy

Please check this box if you want to proceed.

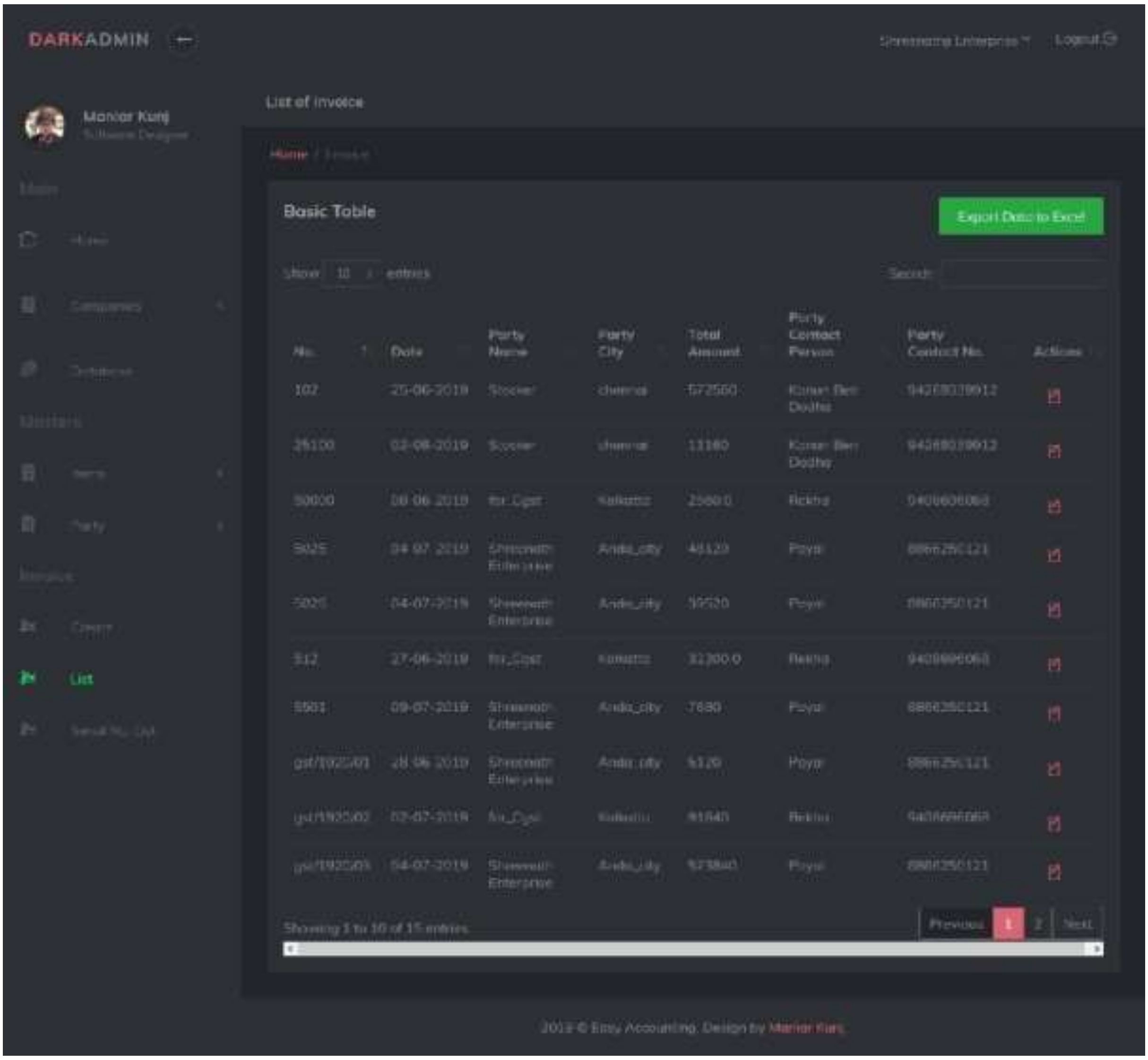


Goods and Services Taxpayer Information

GSTIN 894556

Invalid Length of GSTIN

### 5.3.9 List Creation



List of invoice

Basic Table

Show 10 entries

| No. | Date | Party Name | Party City | Total Amount | Party Contact Person | Party Contact No. | Actions |
|---|---|---|---|---|---|---|---|
| 102 | 25-06-2019 | Stocker | chennai | 572560 | Kiran Ben Dodhia | 94268039912 | ✎ |
| 25100 | 02-08-2019 | Stocker | chennai | 13180 | Kiran Ben Dodhia | 94268039912 | ✎ |
| 50000 | 08-06-2019 | for_Cost | Kolkatta | 25600.0 | Rekha | 9408606060 | ✎ |
| 5025 | 04-07-2019 | Shreenath Enterprise | Anda_city | 46120 | Payal | 8866250121 | ✎ |
| 5026 | 04-07-2019 | Shreenath Enterprise | Anda_city | 10520 | Payal | 8866250121 | ✎ |
| 512 | 27-06-2019 | for_Cost | Kolkatta | 32300.0 | Rekha | 9408606060 | ✎ |
| 5501 | 09-07-2019 | Shreenath Enterprise | Anda_city | 7680 | Payal | 8866250121 | ✎ |
| gst/1920/01 | 28-06-2019 | Shreenath Enterprise | Anda_city | 5120 | Payal | 8866250121 | ✎ |
| gst/1920/02 | 02-07-2019 | for_Cost | Kolkatta | 81640 | Rekha | 9408606060 | ✎ |
| gst/1920/03 | 04-07-2019 | Shreenath Enterprise | Anda_city | 573840 | Payal | 8866250121 | ✎ |

Showing 1 to 10 of 15 entries

2019 © Easy Accounting. Design by Manar Kunj

Used to create a prebuilt profile for each company that keeps track of all information so that it doesnt

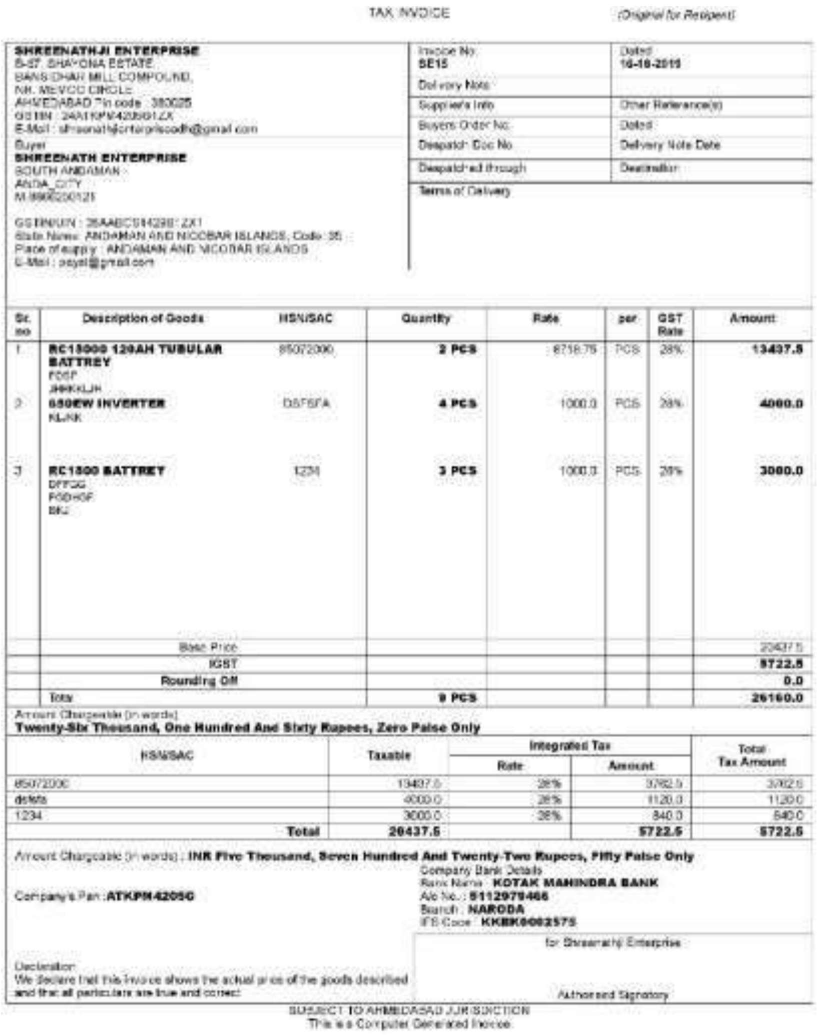
have to be entered on every order.

Modules used :

- sales and accounts receivable sub-system
- management information sub-system

All field inputs are stored and fetched from cluster as needed.

### 5.3.10 Invoice Print



TAX INVOICE (Original for Recipient)

**SHREENATHJI ENTERPRISE**
B-57, SHAYONA ESTATE,
BANSIDHAR MILL COMPOUND,
NR. MEVCO CIRCLE
AHMEDABAD Pin code : 380025
GSTIN : 24ATKPM4205G1ZX
E-Mail : shreenathjienterprisedh@gmail.com

Buyer
**SHREENATH ENTERPRISE**
SOUTH ANDAMAN -
ANDA_CITY
M 9998250121

GSTIN/UIN : 35AABCS1429B1ZX1
State Name: ANDAMAN AND NICOBAR ISLANDS, Code: 35
Place of supply : ANDAMAN AND NICOBAR ISLANDS
E-Mail : payal@gmail.com

| | |
|---|---|
| Invoice No<br>**SE15** | Dated<br>**16-10-2019** |
| Delivery Note | |
| Supplier's Info | Other Reference(s) |
| Buyers Order No | Dated |
| Despatch Doc No | Delivery Note Date |
| Despatched through | Destination |
| Terms of Delivery | |

| Sr. no | Description of Goods | HSN/SAC | Quantity | Rate | per | GST Rate | Amount |
|---|---|---|---|---|---|---|---|
| 1 | **RC15000 120AH TUBULAR BATTREY**<br>FDSF<br>JHHKKLJH | 85072000 | **2 PCS** | 6718.75 | PCS | 28% | **13437.5** |
| 2 | **650EW INVERTER**<br>KLJKK | DSFSFA | **4 PCS** | 1000.0 | PCS | 28% | **4000.0** |
| 3 | **RC1500 BATTREY**<br>DFFGG<br>FGDHGF<br>BKJ | 1234 | **3 PCS** | 1000.0 | PCS | 28% | **3000.0** |
| | Base Price | | | | | | 20437.5 |
| | **IGST** | | | | | | **5722.5** |
| | **Rounding Off** | | | | | | **0.0** |
| | Total | | **9 PCS** | | | | **26160.0** |

Amount Chargeable (in words)
**Twenty-Six Thousand, One Hundred And Sixty Rupees, Zero Paise Only**

| HSN/SAC | Taxable | Integrated Tax | | Total Tax Amount |
|---|---|---|---|---|
| | | Rate | Amount | |
| 85072000 | 13437.5 | 28% | 3762.5 | 3762.5 |
| dsfsfa | 4000.0 | 28% | 1120.0 | 1120.0 |
| 1234 | 3000.0 | 28% | 840.0 | 840.0 |
| **Total** | **20437.5** | | **5722.5** | **5722.5** |

Amount Chargeable (in words) : **INR Five Thousand, Seven Hundred And Twenty-Two Rupees, Fifty Paise Only**

Company's Pan :**ATKPM4205G**

Company Bank Details
Bank Name : **KOTAK MAHINDRA BANK**
A/c No. : **6112979466**
Branch : **NARODA**
IFS Code : **KKBK0002575**

for Shreenathji Enterprise

Authorised Signatory

Declaration
We declare that this Invoice shows the actual price of the goods described and that all particulars are true and correct

SUBJECT TO AHMEDABAD JURISDICTION
This is a Computer Generated Invoice

Used for creating invoice used for filing taxes or maintaining records of all debit and credit in account.

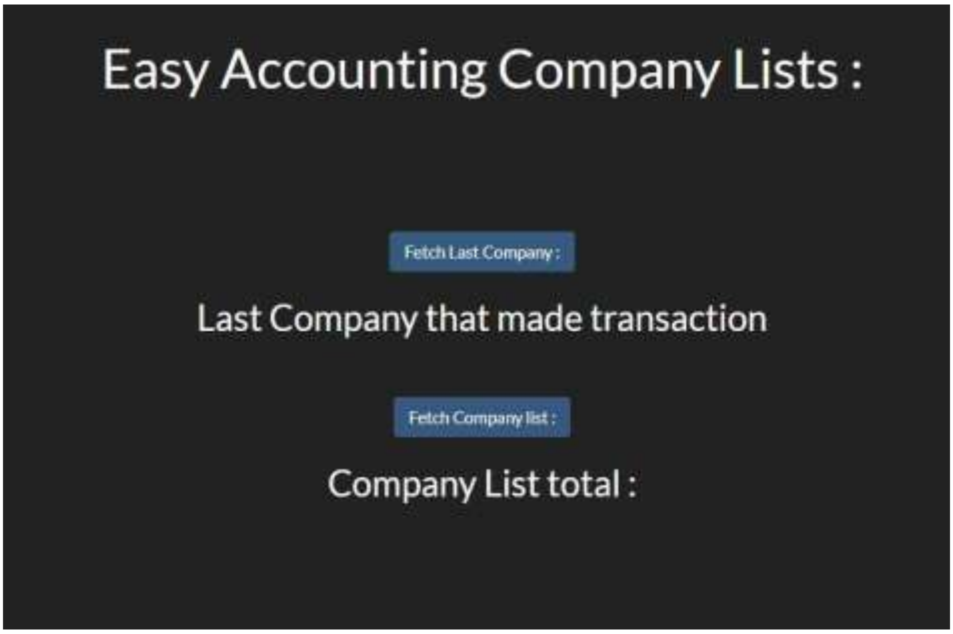
Modules :

- purchase and accounts payable sub-system,
- expense accounting sub-system,
- tax accounting sub- system,

All invoices dB will be fetched from companies,parties and items and then put together while generating invoice.

### 5.3.11 AJAX Implementation :

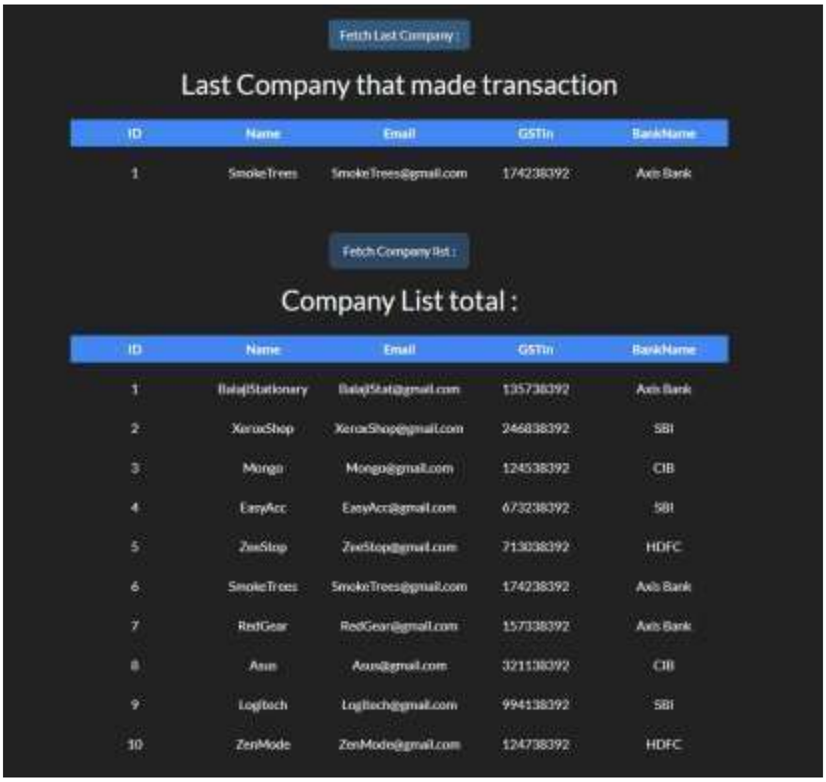
Companies : Used AJAX to fetch dB info and make it easy for the user to check company details in one click. Over here we have fetched dB to display info of all companies as required. It is hosted on a webpage thus making it easier for admin to access data whenever needed.



Easy Accounting Company Lists :

Last Company that made transaction

Company List total :



Last Company that made transaction

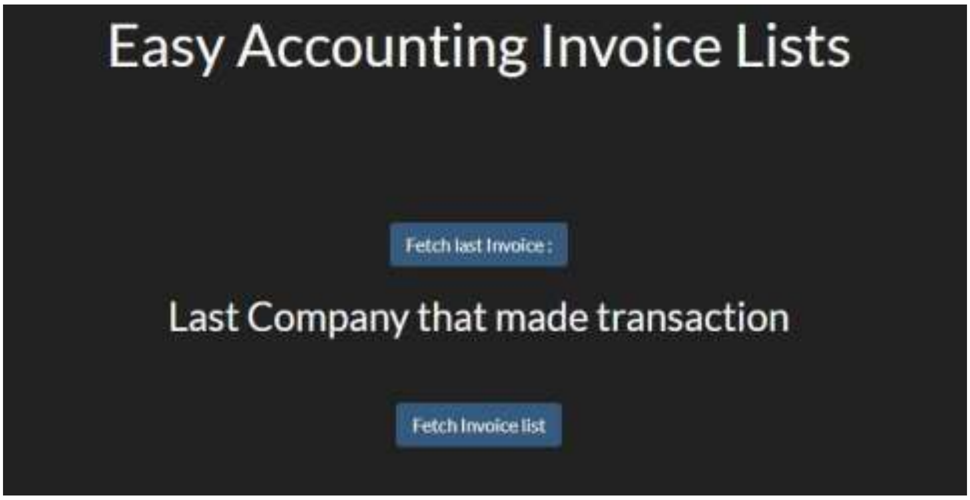
| ID | Name | Email | GSTin | BankName |
|---|---|---|---|---|
| 1 | SmokeTrees | SmokeTrees@gmail.com | 174238392 | Axis Bank |

Company List total :

| ID | Name | Email | GSTin | BankName |
|---|---|---|---|---|
| 1 | BalajiStationary | BalajiStat@gmail.com | 135738392 | Axis Bank |
| 2 | XeroxShop | XeroxShop@gmail.com | 246838392 | SBI |
| 3 | Mongo | Mongo@gmail.com | 124538392 | CIB |
| 4 | EasyAcc | EasyAcc@gmail.com | 673238392 | SBI |
| 5 | ZenStop | ZenStop@gmail.com | 713038392 | HDFC |
| 6 | SmokeTrees | SmokeTrees@gmail.com | 174238392 | Axis Bank |
| 7 | RedGear | RedGear@gmail.com | 157338392 | Axis Bank |
| 8 | Asus | Asus@gmail.com | 321138392 | CIB |
| 9 | Logitech | Logitech@gmail.com | 994138392 | SBI |
| 10 | ZenMode | ZenMode@gmail.com | 124738392 | HDFC |

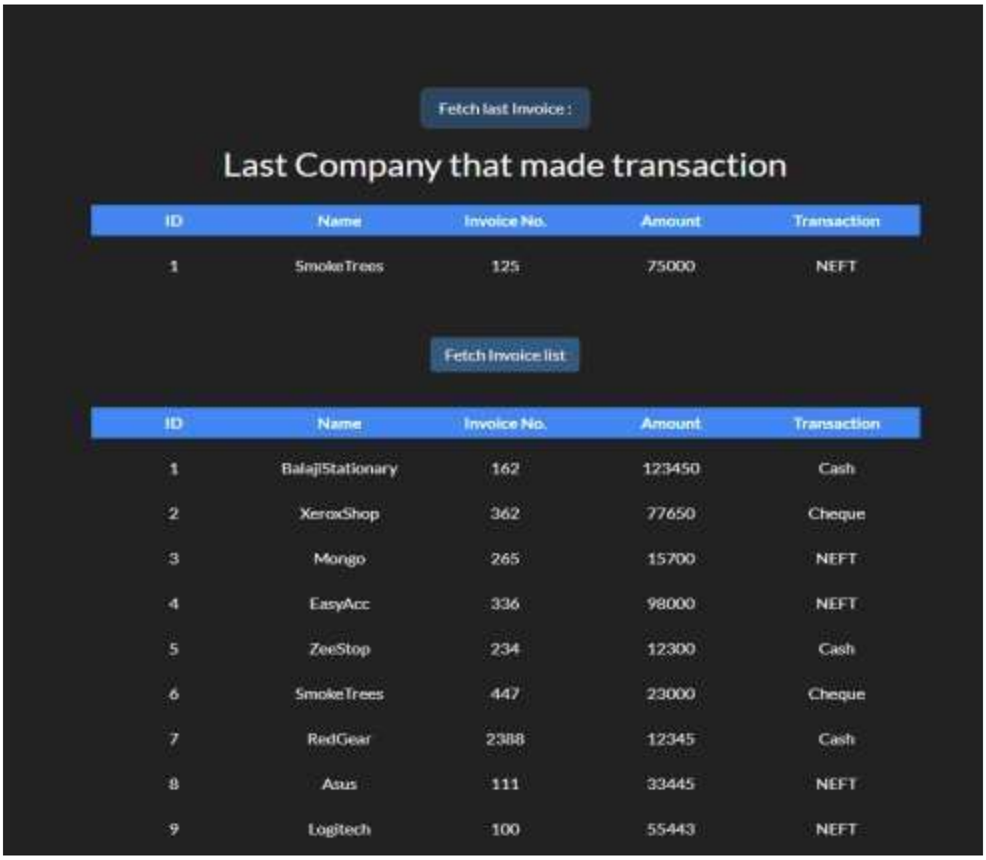
**Invoices :**

Invoices : Used AJAX to fetch dB info and make it easy for the user to check invoice details in one click. Over here we have fetched dB to display info of all invoices as required. It is hosted on a webpage thus making it easier for admin to access data whenever needed.



Easy Accounting Invoice Lists

Last Company that made transaction



Last Company that made transaction

| ID | Name | Invoice No. | Amount | Transaction |
|---|---|---|---|---|
| 1 | SmokeTrees | 125 | 75000 | NEFT |

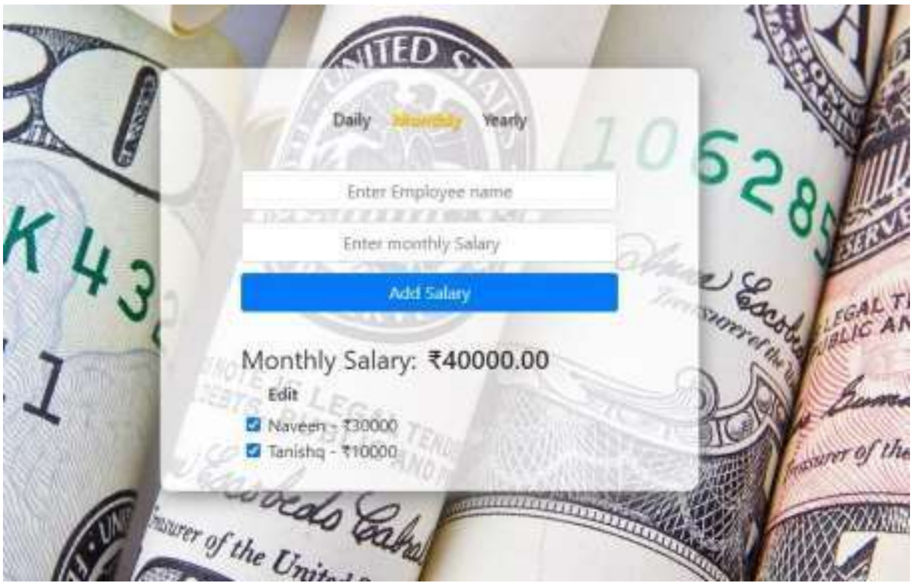
| ID | Name | Invoice No. | Amount | Transaction |
|---|---|---|---|---|
| 1 | BalajiStationary | 162 | 123450 | Cash |
| 2 | XeroxShop | 362 | 77650 | Cheque |
| 3 | Mongo | 265 | 15700 | NEFT |
| 4 | EasyAcc | 336 | 98000 | NEFT |
| 5 | ZeeStop | 234 | 12300 | Cash |
| 6 | SmokeTrees | 447 | 23000 | Cheque |
| 7 | RedGear | 2388 | 12345 | Cash |
| 8 | Asus | 111 | 33445 | NEFT |
| 9 | Logitech | 100 | 55443 | NEFT |

## React :

We have implemented an salary management module for our main accounting system, where we can log salary out for all our employees. That will make it easier for us to see how much money going out in salary and also daily, monthly and yearly views.

- Using this it will be easier to manage outgoing salaries of employees.
- We can enter daily salaries and it will show how much we have paid monthly or yearly.
- Thus giving us a proper estimate of the moneyflow

## Monthly view :

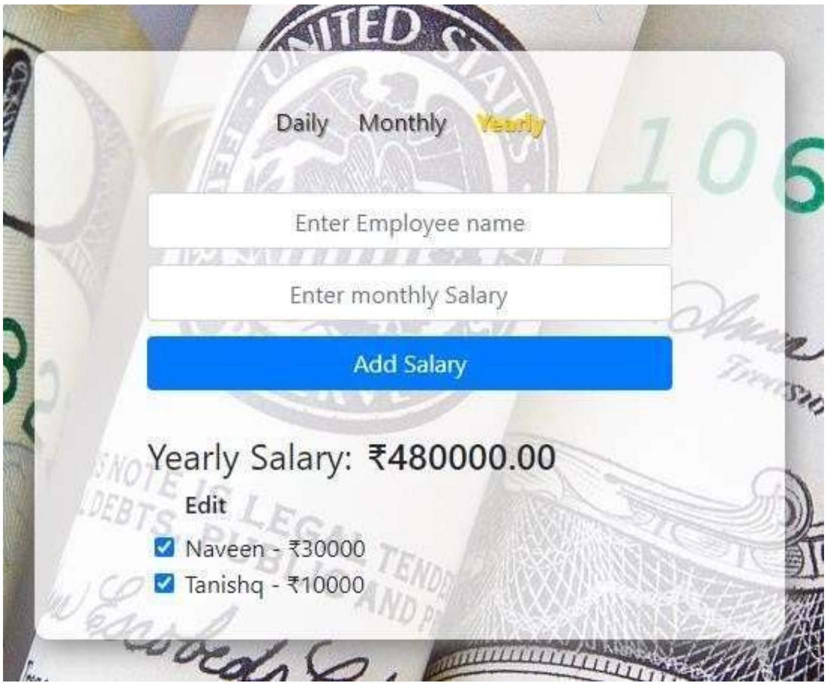


Monthly Salary: ₹40000.00

☑ Naveen - ₹30000

☑ Tanishq - ₹10000

## Yearly view :

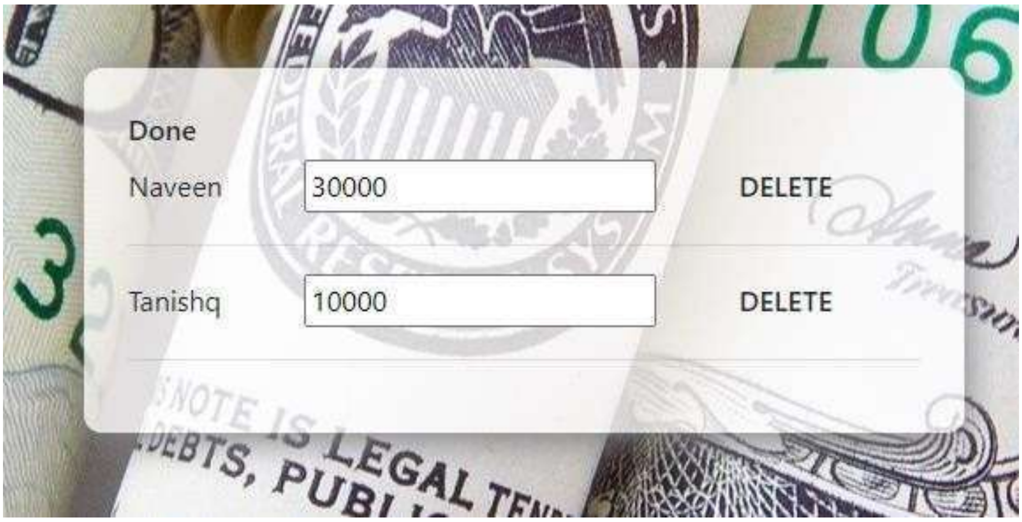


Yearly Salary: ₹480000.00

☑ Naveen - ₹30000

☑ Tanishq - ₹10000

## Edit details :



| | | |
|---|---|---|
| Naveen | 30000 | DELETE |
| Tanishq | 10000 | DELETE |

## 6. Conclusion

The accounting systems in place for small enterprises vary a lot. There are cases when there are no accounting requirements at all and cases where the accounting requirements are relatively strict for small enterprises. However, in practical terms, all small enterprises will need to keep some kind of financial records in order to keep financial control over their businesses. This report summarizes the likely accounting systems from the point of view of small enterprises and identifies some good practices on how to improve the accounting systems for small enterprises. The objective of this report is to provide views on how to improve the accounting systems so that they can provide the owners/managers of the small enterprises with appropriate financial information. The aim is not to add to regulation but to identify good practices which small enterprises may consider before deciding on an appropriate accounting system. However, these recommendations are in no way intended to encroach upon the sovereignty accounting matters.

Accounting Information System is an integration of various sub-systems such as:
(i) cash sub-system
(ii) sales and accounts receivable sub-system
(iii) inventory sub-system,
(iv) purchase and accounts payable sub-system
(v) payroll accounting sub-system
(vii) expense accounting sub-system
(viii) tax accounting subsystem,
(ix) final accounts sub-system
(x) costing sub-system

## 7. References

d
https://nodejs.org/en/docs/
https://devdocs.io/html/
https://reactjs.org/docs/getting-started.html
https://developer.mozilla.org/en-US/docs/Web/JavaScript
https://api.jquery.com/category/ajax/
https://api.jquery.com/

## 8. Code :

**Easy Accounting management system**
https://github.com/tanishqpadwal/Easy-Accounting

**AJAX implementation**
https://tanishqpadwal.github.io/CompanyDetails-AJAX/
https://tanishqpadwal.github.io/Invoice-AJAX/

**React Implementation**
https://github.com/tanishqpadwal/Salary-Management-React/