**Chapter 1**

**Introduction**

**1.1 Business Analytics**

Business Analytics provides a set of tools for strategic planning and tactical implementation of policies, practices, guidelines, processes and procedures that are used in the development, deployment and execution of business plans and strategies and all associated management activities. Business Analytics provides a foundation for successful implementation of both strategic and tactical business decisions regarding current activities, processes, procedures and tasks for the purpose of meeting existing goals and objectives of a profit organization and satisfying customer needs and expectations.

The major idea of Business Analytics is to provide management staff with tools for planning, monitoring and controlling management activities and measuring business performance, and to implement continual improvement processes within an organization. The business management system idea identifies the principles of the successful organization’s existence and it’s closely linked to business success criteria.

Business Management System allows Business Owners to manage their business in a more efficient way. It comes handy when owners want to analyse performance of various aspects of their business.

**1.2 Introduction to Web Technology**

Web technologies is a general term referring to the many languages and multimedia packages that are used in conjunction with one another, to produce dynamic web sites such as this one. Each separate technology is fairly limited on its own, and tends to require the dual use of at least one other such technology. Therefore, we can conclude that all of the components that make up a site are interdependent on one another [1].

HTML (Hyper Text Mark-up Language) is the glue that holds together every web site. Like building a house, you always build a strong foundation first. For any site, HTML is that foundation. HTML is an open source language (i.e. not owned by anyone), which is easy to learn, and requires no fancy (or expensive!) packages to start using it. All you need is something to type with, such as Windows Notepad, and a lot of time and patience.

Although relatively limited by itself, it is the flexibility of HTML that allows web sites to grow in complexity. Like the foundation of your house, HTML is robust enough to support many kinds of languages integrated within your HTML pages.

JavaScript is a 'scripting' language. A bit like a script in a feature film, it is used to decide 'what happens next'. This may be a sequence of screen events, where one event is initiated by the end of another, or it could be a programmed response to a user interacting with the page in some way, e.g. moving their mouse over a link. JavaScript is a complex and powerful language, and may be placed directly inside a HTML page, or in a separate JavaScript file.

jQuery is a JavaScript library designed to simplify HTML DOM tree traversal and manipulation, as well as event handling, animation, and Ajax. It is free, open-source software using the permissive MIT License. Web analysis indicates that it is the most widely deployed JavaScript library by a large margin.

CSS (Cascading Style Sheets) is a relatively new language, designed to expand upon the limited style properties of HTML. Easy to learn and implement, CSS is an excellent way to control the style of your site, such as text styles like size, color and font.

CSS may also be placed inside the HTML page or in separate files. The real advantage of having all of the style properties for your entire site in one single CSS file is that you may edit that single file to effect changes on the whole site, rather than having to go through each HTML file one at a time. For this reason, it is perhaps the most useful web technology and certainly one of my favourites.CSS-P (CSS-Positioning) is a sub-set of CSSs, and is concerned mainly with the layout of your HTML pages. It allows the web designer to place any element (text, graphic etc.) exactly on the screen where they want it, to the pixel.

DOM (Document Object Model) allows the designer to access any element in a HTML page, such as an image, layer or table. Each element may be assigned a unique 'id' name to identify it. When combined with CSS and JavaScript, the DOM may be used to make changes to only "My Tag" and no other element, such as increasing text size or the position of "My Tag" on the screen. JavaScript can also be used to animated such changes to any identified element, such as gradually increasing the size of the text on screen. Each different browser has its own DOM, and this is often used to determine which browser the visitor is using. A particular action may be carried out if the person is using NS6, for example, and ignored if the person is using IE5.

**1.3 Introduction to Node.js**

**Node.js** is an open-source, cross-platform JavaScript run-time environment that executes JavaScript code outside of a browser. Typically, JavaScript is used primarily for client-side scripting, in which scripts written in JavaScript are embedded in a webpage's HTML and run client-side by a JavaScript engine in the user's web browser. Node.js lets developers use JavaScript to write Command Line tools and for server-side scripting—running scripts server-side to produce dynamic web page content before the page is sent to the user's web browser. Consequently, Node.js represents a "JavaScript everywhere" paradigm, unifying web application development around a single programming language, rather than different languages for server side and client-side scripts.

Node.js was originally written by Ryan Dahl in 2009, about thirteen years after the introduction of the first server-side JavaScript environment, Netscape's Livewire Pro Web. The initial release supported only Linux and Mac OS X. Its development and maintenance were led by Dahl and later sponsored by Joyent.

Node.js allows the creation of Web servers and networking tools using JavaScript and a collection of "modules" that handle various core functionality. Modules are provided for file system I/O, networking ( DNS, HTTP, TCP, TLS/SSL, or UDP), binary data (buffers), cryptography functions, data streams, and other core functions. Node.js's modules use an API designed to reduce the complexity of writing server applications.

Node.js brings event-driven programming to web servers, enabling development of fast web servers in JavaScript. Developers can create highly scalable servers without using threading, by using a simplified model of event-driven programming that uses callbacks to signal the completion of a task. Node.js connects the ease of a scripting language (JavaScript) with the power of Unix network programming.

**1.4 Introduction to JavaScript**

JavaScript is a programming language that started off simply as a mechanism to add logic and interactivity to an otherwise static Netscape browser. In the years since its introduction, it has not only supplanted a variety of other competing languages and technologies to become the standard for browser-based programming, but it has also expanded beyond the client space to become a dominant language on the server side, as well.

The Hyper Text Transport Protocol, or HTTP, was designed to do exactly what the name implies, which is to transport HTML text across a network for viewing. But the problem with HTML is that it is completely static, providing no capabilities to implement logic. Basic features we currently take for granted, such as input validation, autocorrect, progress bars or causing a graphic to change during a mouse rollover, would all be impossible for a browser to perform without some type of scripting capability. As a result, when the web first rose to prominence, every vendor in that space provided some mechanism to make HTML web pages more interactive.

Microsoft promoted a competitor to JavaScript named VBScript, along with plug-in technologies, such as ActiveX Data Object (ADO). Sun Microsystems promoted Java applets that were cross-platform and could be used in any browser. Many other technologies came and went as the internet matured. In the end, JavaScript won the battle for popularity, and it is the only client-side scripting technology supported by the Chrome, Firefox, Safari and Internet

JavaScript is an interpreted language, as opposed to compiled languages, such as C++ and Java. This means that the code written in JavaScript does not go through an intermediary compilation stage in which the source code is transformed into machine language that is easy for a CPU to process. Instead, JavaScript is interpreted on the fly by the computer processing it. Because JavaScript code is not packaged in a CPU-friendly form, such as Java bytecode or a binary executable, program execution can be slower than a comparable program written in Scala or F#. However, modern JavaScript engines are highly optimized, and inefficiencies are rarely noticeable when using modern hardware.

**1.5 Introduction to jQuery**

jQuery is a JavaScript library designed to simplify HTML DOM tree traversal and manipulation, as well as event handling, animation, and Ajax. It is free, open-source software using the permissive MIT License. Web analysis indicates that it is the most widely deployed JavaScript library by a large margin.

jQuery's syntax is designed to make it easier to navigate a document, select DOM elements, create animations, handle events, and develop Ajax applications. jQuery also provides capabilities for developers to create plug-ins on top of the JavaScript library. This enables developers to create abstractions for low-level interaction and animation, advanced effects and high-level, theme able widgets. The modular approach to the jQuery library allows the creation of powerful dynamic web pages and Web applications.

**1.6 Web Application Pipeline Architecture**

The figure shown gives an abstract, high-level block diagram of how a web application processes data. In the diagram, commands enter from the left and proceed through what can be thought of as a processing pipeline. Some commands specify geometric objects to be drawn, and others control how the objects are handled during the various processing stages. Let’s now take a look at the separate stages

## 

**Figure 1.1:** Pipeline Architecture of Web Applications

The concept of the web application pipeline is what really sets it apart from general CPUs, although the idea of a pipeline is the same as that used by general purpose CPUs.  
The graphics pipeline is built in stages. Every stage is specialized in precisely one element of the rendering process. The main components of pipeline architecture include web browser, CDN, cloud storage and web app servers.

As shown in the Figure 1, rather than having all commands proceeds immediately through the pipeline, you can choose to accumulate some of them in a display list for processing at a later time.In a single project scenario, separation of concerns is achieved through the use of folders. The default template includes separate folders for MVC pattern responsibilities of Models, Views, and Controllers, as well as additional folders for Data and Services. In this arrangement, presentation details should be limited as much as possible to the Views folder, and data access implementation details should be limited to classes kept in the Data folder.

**1.7 Objectives of Business Analytics**

The main purpose of Business Analytics is to provide an alternate and convenient way for a businessman to evaluates his company’s achievements of their goals and objectives in an automatic system.

The main objectives of Business Analytics are:

1. **Getting Maximum Results with Minimum Efforts -** The main objective of Business Analytics is to secure maximum outputs with minimum efforts & resources. Management is basically concerned with thinking & utilizing human, material & financial resources in such a manner that would result in best combination. This combination results in reduction of various costs.
2. **Increasing the Efficiency of factors of Production -** Through proper utilization of various factors of production, their efficiency can be increased to a great extent which can be obtained by reducing spoilage, wastages and breakage of all kinds, this in turn leads to saving of time, effort and money which is essential for the growth & prosperity of the enterprise.
3. **Maximum Prosperity for Employer & Employees -** Management ensures smooth and coordinated functioning of the enterprise. This in turn helps in providing maximum benefits to the employee in the shape of good working condition, suitable wage system, incentive plans on the one hand and higher profits to the employer on the other hand.
4. **Human betterment & Social Justice –** Business Analytics serves as a tool for the upliftment as well as betterment of the society. Through increased productivity & employment, management ensures better standards of living for the society. It provides justice through its uniform policies.

**1.8 Organization of the Report**

Chapter 1 provides the information about the web technologies used and objectives of our project. In Chapter 2, specifies the hardware and software requirements. Chapter 3, all the functions used in our program is described. Chapter 4 gives the idea of the project and its actual implementation. Chapter 5 discusses the testing done on the project. Chapter 6 discusses about the results and discussions regarding the project. Chapter 7 concludes by giving the direction for future enhancements.

**Chapter 2**

**Requirements Specification**

A computerized way of handling information about Transactions, client and employee 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.

**Specific Requirements**

The specific requirements of the Business Analytics web application are stated as follows:

**2.1 Hardware Requirement**

The section of hardware configuration is an important task related to the software development insufficient random access memory may affect adversely on the speed and efficiency of the entire system. The process should be powerful to handle the entire operations. The hard disk should have sufficient capacity to store the file and application

Processor : Intel Pentium T4200/ Intel Core Duo 2.0 GHz / more

RAM : Minimum RAM capacity

Hard disk : 80GB or more

Peripherals : Keyboard, Compatible mouse

Cache Memory : L2-1 MB

GPU : Intel HD Graphics

Monitor Resolution : 1024\*768 or 1336\*768 or 1280\*1024

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

**2.2 Software Requirement**

A major element in building a system is the section of compatible software since the software in the market is experiencing in geometric progression. Selected software should be acceptable by the firm and one user as well as it should be feasible for the system.

This document gives a detailed description of the software requirement specification. The study of requirement specification is focused specially on the functioning of the system. It allows the developer or analyst to understand the system, function to be carried out the performance level to be obtained and corresponding interfaces to be established.

Operating system : WINDOWS 10 or above

Editor : Sublime text/Notepad ++

Programming language : NodeJS, JavaScript

Web browser : Google Chrome or any web browser

Server : NodeJS

**Chapter 3**

**Function Description**

## MySQL (npm package) Introduction

A pure node.js JavaScript Client implementing the MySql protocol.

The description of all the functions used in the Business Analytics is given below:

**Table 2.1 MYSQL Functions**

|  |  |
| --- | --- |
| **Function** | **Description** |
| mysql.createConnection() | Used to setup a new connection information, like hostname, user, password, database name. |
| connector.connect() | Creates the database connection |
| connector.query(q) | Queries the database with the query q. |
| Connector.destroy() | Immediately termination of underlying socket responsible for the connection |
| mysql.createPool(config) | this module also provides built-in connection pooling using |
| Connection.release() | Closes the connection with the mysql server. |
| [mysql.query()](https://www.w3schools.com/php/func_mysqli_fetch_assoc.asp) | Helps in creating dynamic queries.  Ex:- var sql = ”SELECT \* FROM ?? WHERE ?? = ?”;  Var inserts = [ ‘users’, ’id’, userID];  Sql = mysql.format(sql,inserts); |

**Chapter 4**

**System Design and Implementation**

**4.1 Introduction**

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

This Project is implemented with languages namely, HTML, CSS, jQuery and JavaScript. Here, HTML and CSS has been used for front end designing of the project. JavaScript has been used for Client-Side Validation. JavaScript (NodeJS) has been used for server-side web development. MySQL npm package has been used to handle the administration of MySQL database.

**4.2 Overall Design Process**

The Business Analytic is designed for any Business to replace their existing manual, paper based daily record keeping. The new system is designed to store cash inflow and outflow information. These services are to be provided in an efficient, cost effective manner, with the goal of reducing the time and resources currently required for such tasks.

Financial record keeping is one of the most important operation of any Business, that plays the key role on deciding its success. The record typical involves tracking the cash flows from different entities. All of this information must be managed in an efficient and cost wise fashion so that an institution's resources may be effectively utilized. Business Analytic will make sure the record keeping of the business is automated making it more efficient and error free. It aims at standardizing data, consolidating data ensuring data integrity and reducing inconsistencies.

In this we have developed some forms. The brief description about them is as follow:-

**Overview:**

This entity shows the overall cash flow information in the form of a doughnut graph.

**Add Transaction:**

This entity used to record a transition.

**Daily Journal:**

This entity shows the daily journal entry.

**Journal Entry:**

Shows the transactions in table format.

**Bank account:**

This entity show in the cashflow occurring between account of company and customer.

**Cash account:**

This entity shows information about all the cash transactions.

All the recorded transactions are also filtered and shown based on different important attribute.

**4.3 Flowchart**

The below figure represents the schema of all relations of the Hospital Dashboard.

LOGIN

Add Transaction

HOME

View Analytics

Daily Journal

Journal Entry

Overview

**Figure 4.1**: Flowchart of the system.

**4.4 Use case Diagram**

The below figure depicts the modules for the actor ‘Admin’ using use case diagram.

**Figure 4.2**: Use case of the Admin.

**4.5 Pseudo Code for simple operations**

4.5.1. Pseudo code for login

Step 1: Begin

Step 2: Open web browser.

Step 3: Enter the URL “http://localhost:3000”.

Step 4: Enter the username and password and press the login button.

Step 5: If login credentials are correct, then home page is displayed. Else go to Step 5.

Step 6: If login credentials are incorrect, then Display ‘INVALID CREDENTIALS’.

Step 7: End

4.5.2. Algorithm for insert new transaction

Step 1: Begin.

Step 2: Make connection to Database using hostname, username and password.

Step 3: Get values of all fields like Data, Particular, Description, Name, Debit, Credit.

Step 4: From query with above fields as variables and execute using query() function of mysql node module.

Step 5: If above method returns false, Send ‘ An error occurred’ as response to client.

Step 6: Get particular2 value by calling getJournalEntryParticular().

Step 7: From query to insert into journal\_entry table and execute it.

Step 8: Call UpdateLedger() method to insert into corresponding ledger tables;

4.5.3. ALGORITHM FOR getJournalEntryParticular(data)

Step 1: Begin

Step 2: Initialize variable p as empty string.

Step 3: Check value for data.particular.

Step 4: Make suitable switch statement and update value of p accordingly.

Step 5: Return value of p.

Step 6: End

4.5.4. ALGORITHM FOR updateLedger(data,callback)

Step 1: Begin

Step 2: Get values of debitAccount and creditAccount according to value of data.particular.

Step 3: Create tables for debitAccount and creditAccount If they do not exists.

Step 4: Form queries according to the data.

Step 5: Execute queries using query() method.

Step 6: call Callback(err,success) method.

Step 7: End

4.5.5. ALGORITHM FOR displaying table

Step 1: Begin

Step 2: Connect to Database.

Step 3: Call homeData().

Step 4: Return first result of homedata() as an array.

Step 5: Inside a for loop in ejs, render different values of array as <li> or <table>.

Step 6: End

4.5.6. ALGORITHM FOR DELETING TRANSACTION

Step 1: Begin

Step 2: Get index of entry which is clicked.

Step 3: Make an AJAX call to server passing id as parameter.

Step 4: Make query to delete form journal\_entry where id = parameter.

Step 5: if err, send ‘AN ERROR OCCURRED’ as reponse to client.

Step 6: Else, send ‘DELETED SUCCESSFULLY’ as response to client.

Step 7: End

4.5.7. ALGORITHM FOR UPSATING TRANSACTION

Step 1: Begin

Step 2: Get index of entry which is clicked.

Step 3: Make an AJAX call to server passing id as parameter.

Step 4: Make query to update from journal\_entry where id = parameter.

Step 5: Execute query using query() function.

Step 6: If err, send ‘ AN ERROR OCCURRED’ as response to client.

Step 7: Else, send ‘UPDATED SUCCESSFULLY’ as reponse to client.

Step 8: End

**Chapter 5**

**TESTING**

**5.1 INTRODUCTION TO TESTING:**

Testing performs a very crucial role in quality assurance and ensuring the reliability of the software. During testing, the program to be tested is executed as a set of test cases and the output is performed as expected. The service of testing for errors in program depends critically on the test cases.

**5.2 TESTING PROCESS:**

Basic levels of testing include:

1. Unit testing.
2. Integration testing.
3. System testing.

**5.2.1 UNIT TESTING:**

Unit testing focuses verification on the smallest unit of the software design, the software components or module. The unit testing was implemented in the project to test the following:

1. Each module was tested to ensure that the information flows properly into and out of the program unit under test.

2. Each unit was tested to ensure its proper functioning.

3. Different modules are tested independently to ensure that the desired output is got each module.

**5.2.2 INTEGRATION TESTING**

Integration testing is the systematic technique for constructing the program structures. While at the same time, tests are conducted to uncover error associated with the interfacing. The objective is, to use the unit tested components and build a program structure that has been designed. During the testing phase, bottom up approach is implemented. The bottom up approach of integration testing is carried out in the following way:

1. The components at the bottom level which have no subordinates are tested.
2. After combining, different components are linked together to the main menu to ensure that all units are linked to it and the desired output is got.

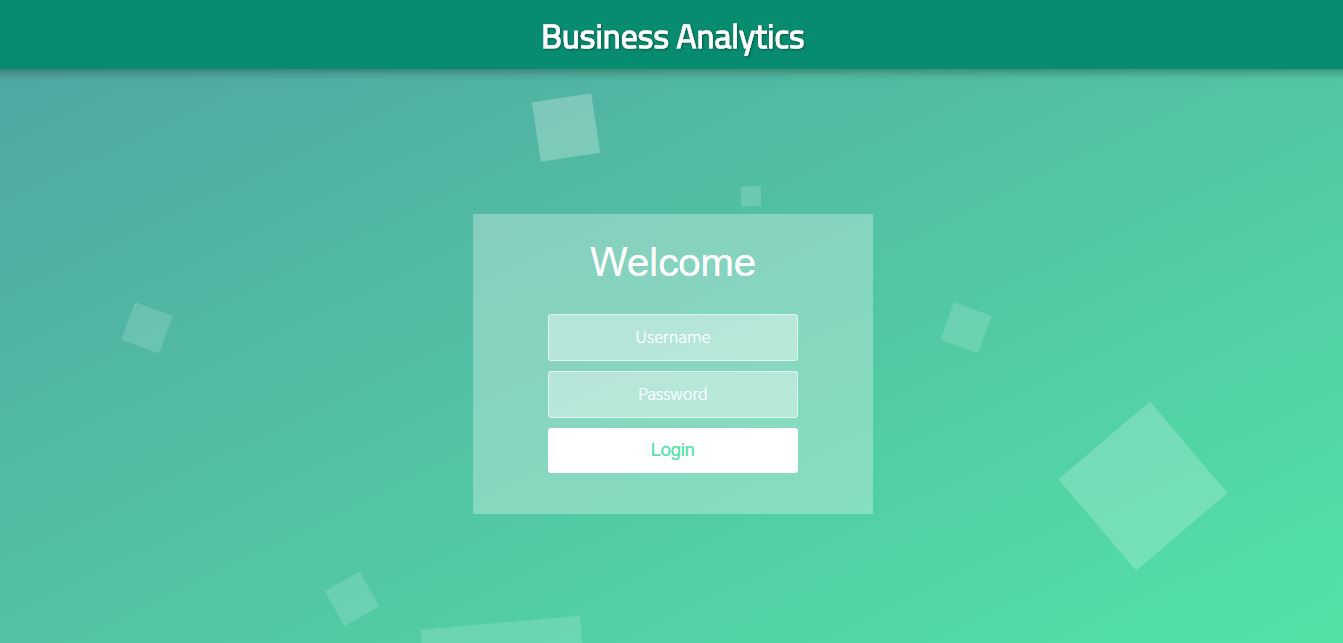
**5.2.3 SYSTEM TESTING**

Here we test the system according to the requirements that is specified in the requirements analysis. In this level, all the modules are integrated into a system and then they are tested. Thus, testing starts from the main menu and ends with end results.

**Chapter 6**

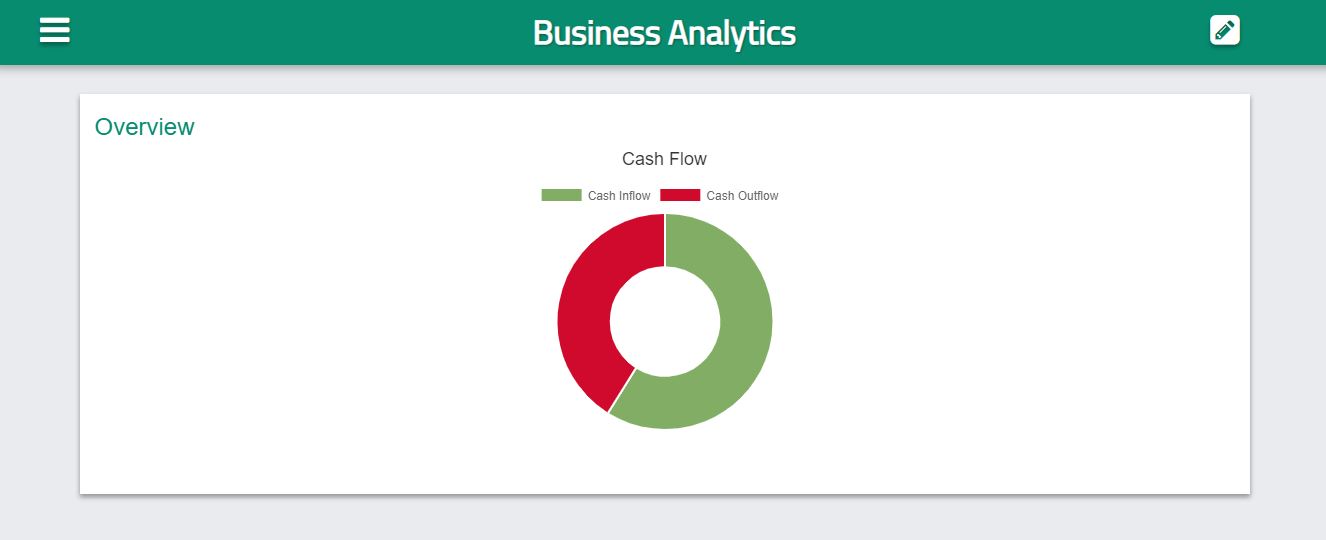
**Results and Discussion**

**Login page:** In Fig. 6.1, The user enters the username and password to login access the hospital database.

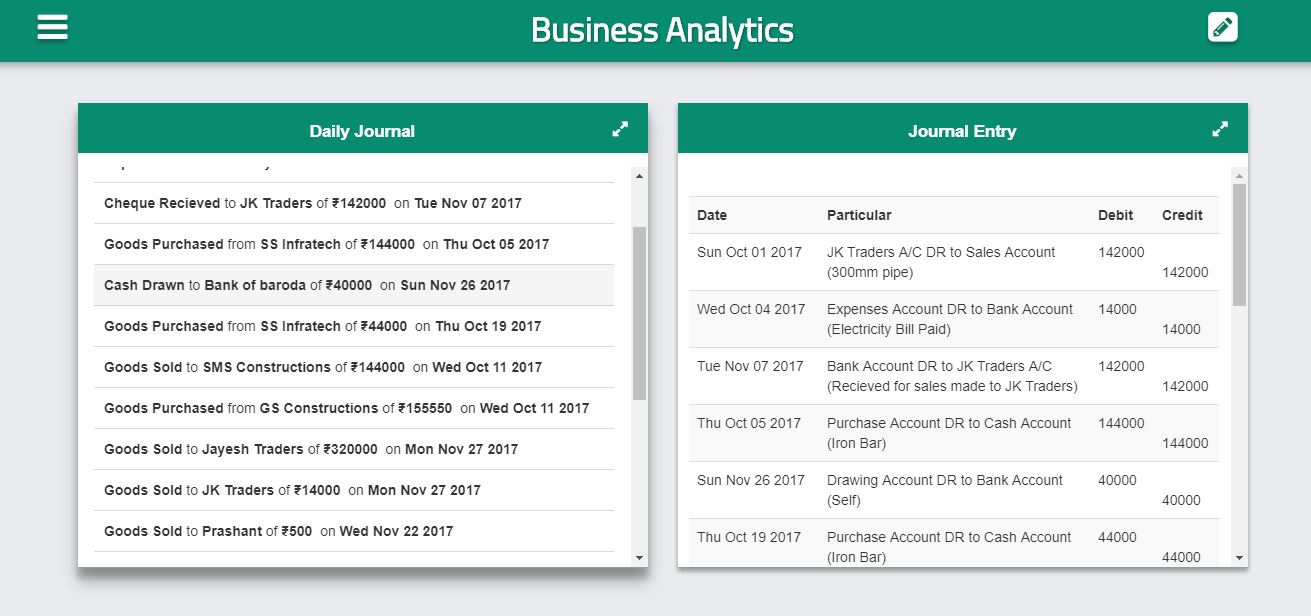
****

**Figure 6.1 Login**

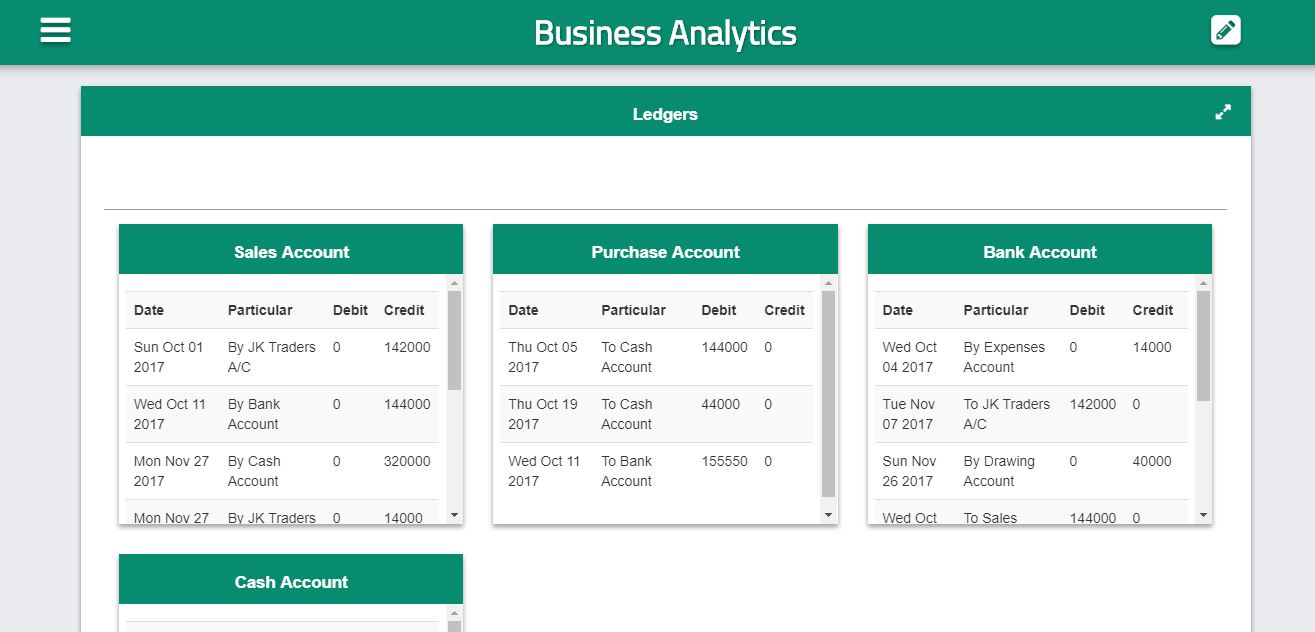
**Home Page:** In Figure 6.2, 6.3 and 6.4, the user gets a variety of options to select from.



**Figure 6.2 Home page**

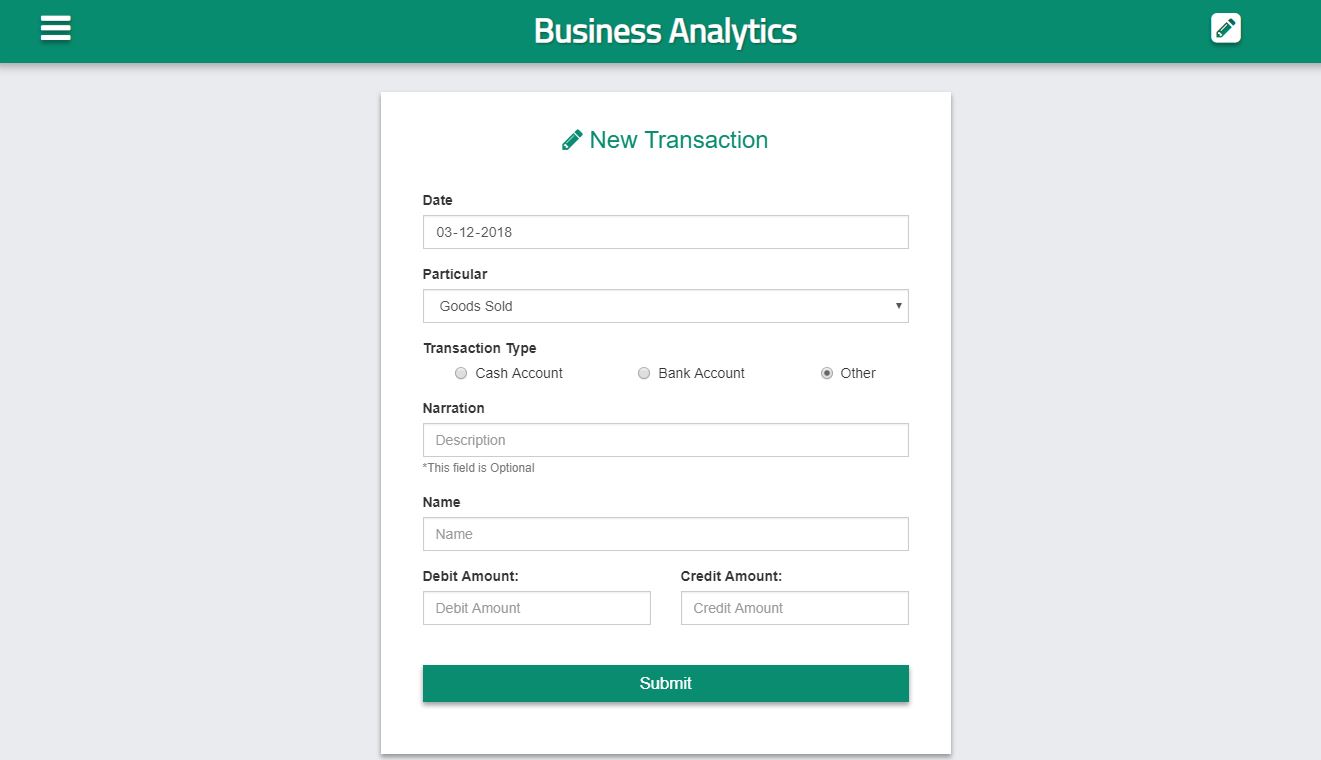
****

**Figure 6.3 Home page below\_1**

****

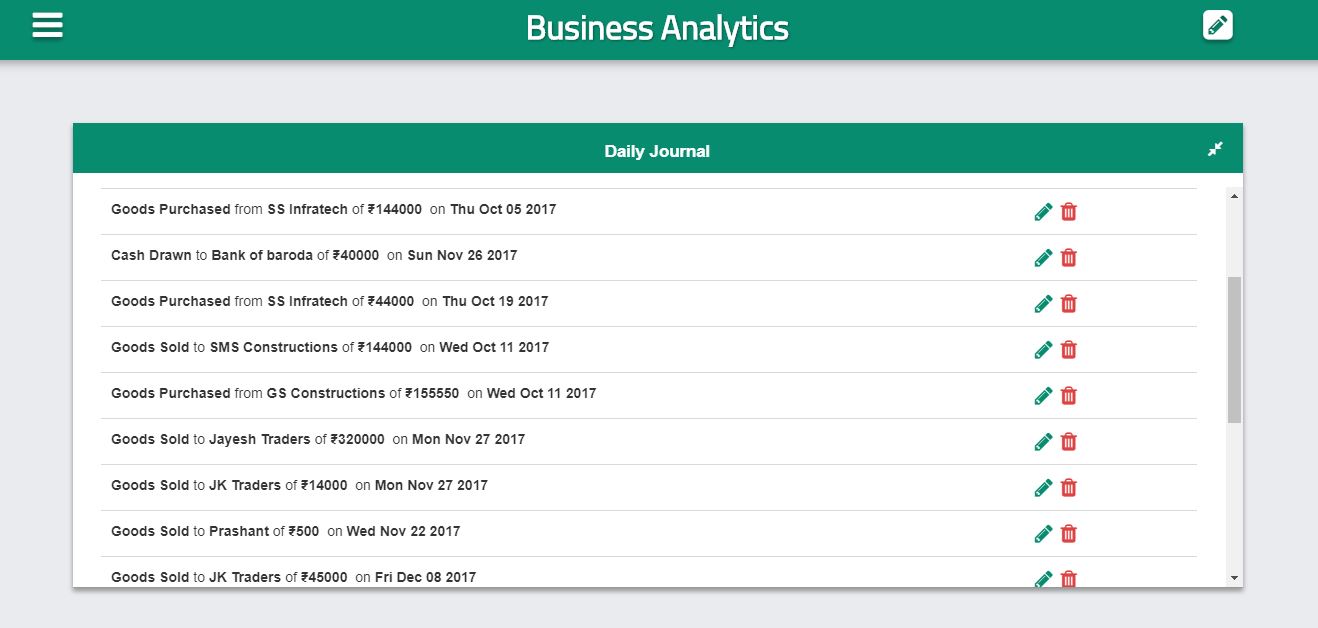
**Figure 6.4 Home page below\_2**

**Add new transaction:** In Figure 6.5, it allows the user to add transaction statements.

****

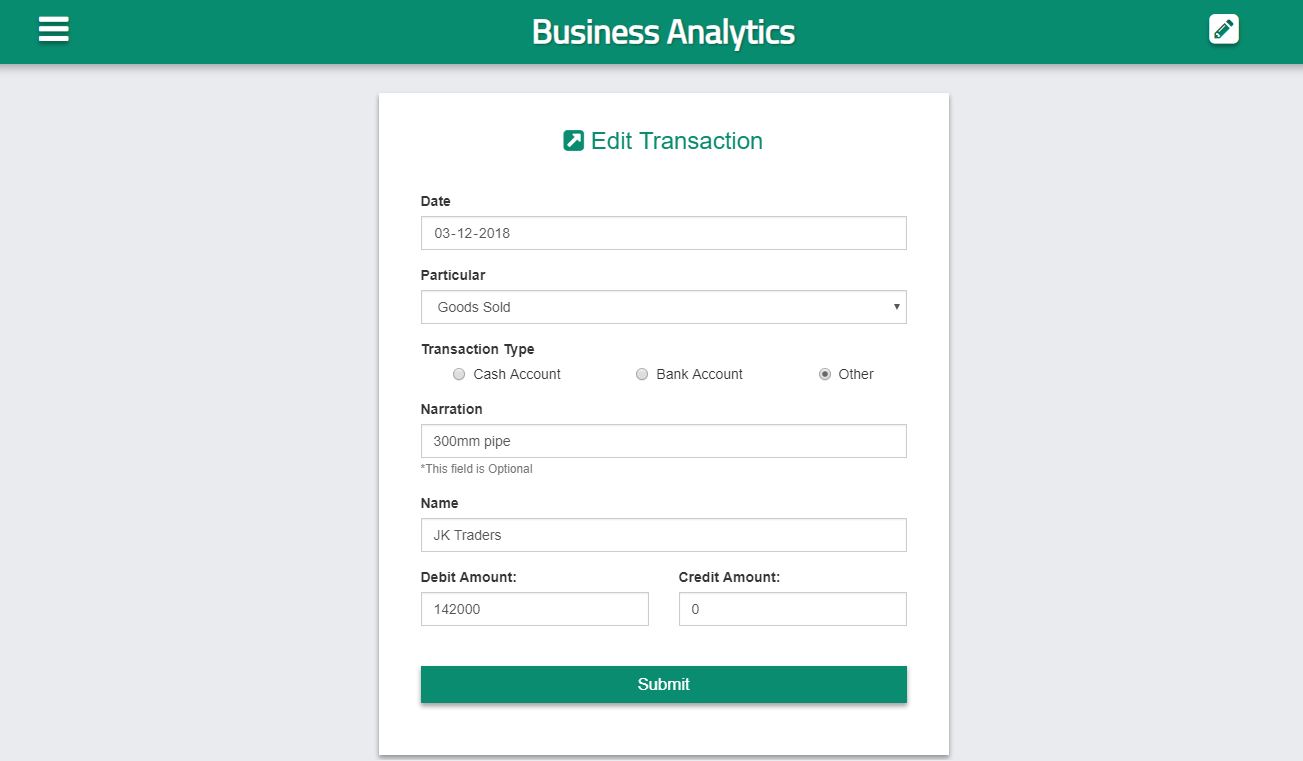
**Figure 6.5 - Add new transactions**

**Edit and Delete Transaction:** In Figure 6.6, it allows the user to modify and delete the details of the transactions.

****

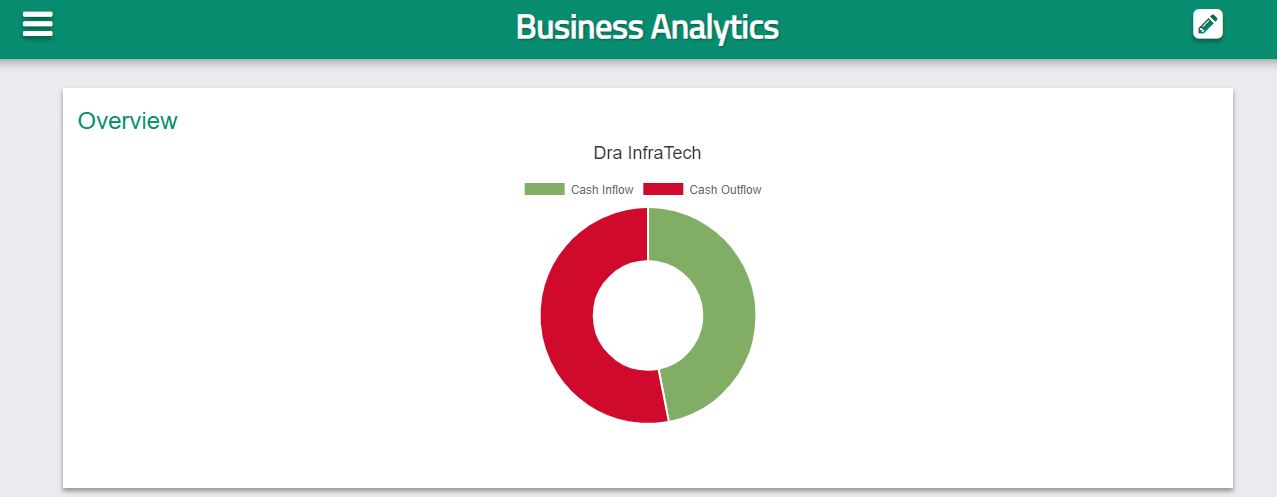
**Figure 6.6: Edit and delete transactions**

**Edit Transaction:** In Figure 6.7, it allows the user to modify the details of the transactions.

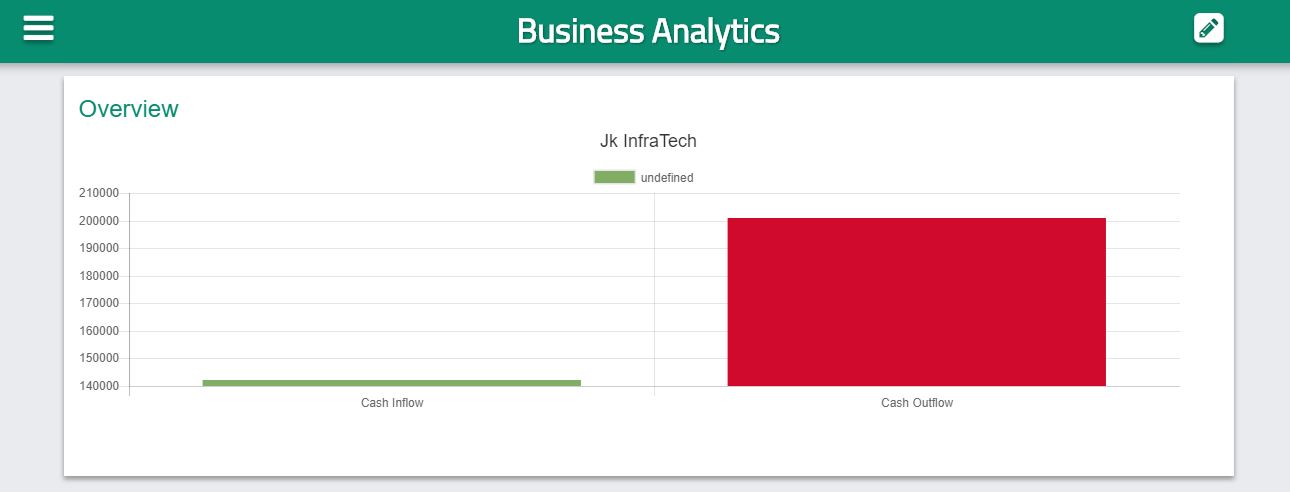
****

**Figure 6.7 Edit Transaction**

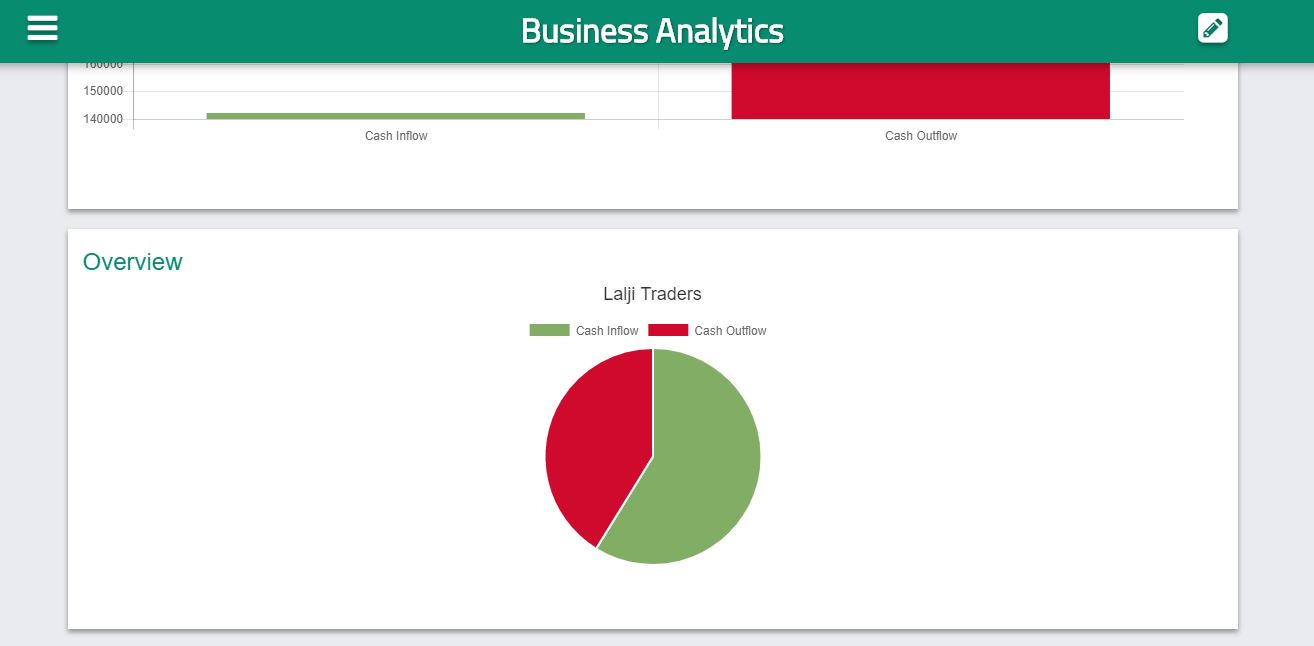
**Dashboard:** In Figure 6.8, 6.9 and 6.10, the user views the graphical representations of data stored in the database



**Figure 6.8 Dashboard**

****

**Figure 6.9 Dashboard below\_1**

****

**Figure 6.10 Dashboard below\_2**

**Chapter 7**

**Conclusion and Future Enhancements**

**7.1 Conclusion**

Small Business Owners don’t have enough insights data to manage their business. As a result, they take all financial decision based on assumptions and prior experience. This results in a big problem, if the owner or manager is an inexperienced one. This application provide owners enough insights data so that they can take financial and other decisions based on actual data which will eventually help in increased growth of business. The front end of this application is programmed in HTML, CSS, Bootstrap, JavaScript, jQuery, and the back-end by node.js and ejs and the database is to be maintained by MySQL. This will hugely impact the industry and will bring about a lot of changes in its working. One can easily maintain the stock of their company with all its details and increment and decrement based on purchase and sales. We hope to see a revolution in the industry with this project.

**7.2 Future Enhancements**

The future scope of our project is vast and can be used in extensive ways:

* It can be used to show the movements of capital in more illustrative way and also can help to clients of the businessmen to have the better view about the performance of their partner company.
* It can be connected with artificial intelligence which can predict future problems and growth of company.
* It can be designed to have more brands than the company deals with

**References**

1. <https://en.wikibooks.org/wiki/Introduction_to_Information_Technology/Web_Technologies>
2. <http://stackoverflow.com/web-application-DevOp>.
3. <http://github.com/mysql>
4. W3school (<https://www.w3schools.com/php/default.asp> , <https://www.w3schools.com/html/default.asp>, <https://www.w3schools.com/js/> , <https://www.w3schools.com/jquery/default.asp> , <https://www.w3schools.com/css/default.asp> )