

# **BREAKING THE BRICKS**

**A Project Report Submitted**

**In Partial Fulfilment of the Requirements For**

**the Degree of**

**MASTER OF COMPUTER APPLICATIONS**

**By**

**BHARTI KHANNA**

**University Roll No 1900290149035**

**Under the Supervision of**

**Mr.Ankit Verma**

**Assistant Professor**

**KIET Group of Institutions**



**Submitted to**

**DEPARTMENT OF COMPUTER APPLICATION**

**Affiliated to**

**DR. A. P. J ABDUL KALAM TECHNICAL UNIVERSITY**

**LUCKNOW**

**JULY,2021**

## **DECLARATION**

I hereby declare that the work presented in this report entitled "**Breaking the Bricks**". was carried out by US. I have not submitted the matter embodied in this report for the award of any other degree or diploma of any other University or Institute. I have given due credit to the original authors/sources for all the words, ideas, diagrams, graphics, computer programs, experiments, results that are not my original contribution.

I have used quotation marks to identify verbatim sentences and given credit to the original authors/sources.

I affirm that no portion of my work is plagiarized, and the experiments and the results reported on the report are not manipulated. In the event of a complaint of plagiarism and the manipulation of the experiments and result, We shall be fully responsible and answerable.

**BHARTI KHANNA**

**UNIV.ROLLNO 1900290149035**

## **CERTIFICATE**

Cerified that **BHARTI KHANNA(Univ.Roll No-1900290149035)** have carried out the project work having the “**Breaking the bricks**” for Masters of Computer Applications from Dr.A.P.J Abdul Kalam Technical University (AKTU) ,Technical University,Lucknow under my supervision.The project report embodies original work ,and studies are carried out by the student himself/herself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

**Date:**

**BHARTI KHANNA(Univ.RollNo-1900290149035)**

This is to certify that the above statement made by the candidate is correct to the best of our knowledge.

**Date:**

**Mr. Ankit Verma**  
**Assistant Professor**  
**Department of Computer Application**  
**KIET Group of Institutions, Ghaziabad**

**Signature of External Examiner**

**Signature of Internal Examiner**

**Dr. Ajay Kumar Shrivastava**  
**Head, Department of Computer Application**  
**KIET Group of Institutions, Ghaziabad**

## ABSTRACT

This system reduces the work load and easy to maintain for a long time than normal game. The user can check his/her records results/score. The user can check score by entering his/her username only no need to search to search all the records.

So the maintenance and Presidency of game became very easy

- Easy accessibility
- It make searching records easier and faster
- User is no longer require to check all the records as now it can be searched over the software by choosing some options
- The user need not to type in most of the game
- On the whole it liberates the user from keeping lengthy manual records.
- Every one wants his/her work to be done by computer automatically and displaying the results for further manipulations.
- So this project is about providing convenience.

## ACKNOWLEDGEMENTS

Success in life is never attained single handedly. My deepest gratitude goes to my thesis supervisor **Mr. Ankit Verma**, for his guidance, help and encouragement throughout our project work. Their enlightening ideas, comments, and suggestions.

Words are not enough to express my gratitude to **Dr. Ajay Kumar Shrivastava**, Professor and Head, Department of Computer Application, for his insightful comments and administrative help at various occasions.

Fortunately, I have many understanding friends, who have helped me a lot on many critical conditions.

Finally, my sincere thanks go to my family members and all those who have directly and indirectly provided me moral support and other kind of help. Without their support, completion of this work would not have been possible in time. They keep my life filled with enjoyment and happiness.

**BHARTI KHANNA**

**Univ.Roll No.1900290149035**

## TABLE OF CONTENTS

Declaration	ii
Certificate	iii
Abstract	iv
Acknowledge	v
Table of content	vi
List of tables	viii
List of figures	viii
Chapter 1: Introduction	9-20
1.1 Project description	9
1.2 Project Scope	11
1.3 Identification of need	12
1.4 Problem Statement	13
1.5 Hardware/software used in project	14
1.6 Project schedule	18
1.6.1 Pert Chart	
1.6.2 Gantt Chart	
Chapter 2: Literature Review	21-26
Chapter 3: Feasibility Study	27-32
3.1 Introduction	27
3.2 Main Aspects	29
3.2.1 Technical feasibility	
3.2.2 Economical Feasibility	
3.2.3 Operational Feasibility	
3.3 Benefits	31
3.4 SRS	32

Chapter 4: Design	34-44
4.1 Introduction	34
4.2 Analysis	35
4.3 SDLC	37
4.4 Soft. Engg. Paradigm	39
4.4.1 Prototype model	
4.5 DFD	42
4.7 ER Diagram	44
 Chapter 5: Report	 45-51
5.1 Gist	45
5.2 Some Screenshots	46
 Chapter 6: Coding	 52-59
 Chapter 7: Testing	 60-62
7.1 Introduction	60
7.2 Level of testing	61
7.3 Some Important observations	62
7.4 test case result summary	62
 Chapter 8: Conclusion & Future scope	 64
8.1 Conclusion	64
8.2 Future Scope	64
 Bibliography	 65

## LIST OF FIGURES

Figure 1.1 Pert chart.....	18
Figure 1.2:Gannt chart for project .....	19
Figure 2.1:Design .....	23
Figure 2.2:Flow of website .....	23
Figure 2.3:Some screens.....	24
Figure 4.1:Above image depicting the planning step.....	377
Figure 4.2:Prototype model .....	40
Figure 4.3:Apex Work .....	42
Figure 4.4 Context level.....	42
Figure 4.5 Dfd I level .....	43
Figure 4.6:Uml use case diagram.....	43
Figure 4.7:ER diagram of system .....	444
Figure 5.1 System .....	455
Figure 7.1:Testing pyramid.....	61



# **CHAPTER 1**

## **INTRODUCTION**

### **1.1 PROJECT DESCRIPTION**

The objective of this game is to show that how a normal can easily be played without even get knowledge of the game. This is simply a game in which user can be easily be logged in with their username and password by entering this it easily be logged in .It is easily flexible like data can be deleted enter or updated easily.

“BREAKING THE BRICKS” is a javascript and node.js application .These days we are using so many games in which we can log in with their gmail account facebook account in this we have simply login with their username .BUT this time we have an application which can do our work more simpler and easier i.e javascript. This game is used to store huge amount of data properly and consistently.

Using Javascript platform provides different features in it:

#### **1.1.1 Validating User's Input**

Javascript is very useful while using forms .It has the capability to validate user Input for errors and also saves times.If the user leaves a required field empty or the information is incorrect,Javascript checks for them before sending the data over to the server.

### **1. 1.1.2 Simple Client -side Calculations**

Since Javascript is a client-side technology, it can perform basic calculations on the browser. The browser does not need to ask server time for every task. This is especially helpful when a user needs to perform these calculations repeatedly. In these cases, connecting to the server would take a lot more time than performing the actual calculations.

### **1.1.3 Greater Control**

JavaScript provides greater control to the browser rather than being completely dependent on the web servers. JavaScript provides various browsers with additional functionalities that help reduce server load and network traffic.

### **1.1.4 Platform Independent**

Since browser interprets Javascript, it solves the problems of compilation and compatibility. Thus it can run on windows, Macintosh, and other Netscape-supported systems. Also, it is possible to embed them in any other script like HTML that keeps Javascript into use.

### **1.1.5 Handling Dates and Time**

Unlike other programming languages, Javascript has built-in functions to determine the date and time. Thus it is very easy to code only by using methods like `getDate()`.

### **1.1.6 Generating HTML Content**

Javascript has very handy features to dynamically generate HTML content for the web. It allows us to add text, links, images, tables etc after an event occurrence.

### **1.1.7 Detecting the User's Browser and OS**

Javascript is very capable in the detection of the user's browser and OS information. Through Javascript runs on every platform, there may occur a situation where we need the user's browser before processing. This can be helpful for writing code that results in different outputs in different browsers.

### 1. 1.1.8 Arrow Functions

These Functions are very useful in simplifying the syntax and tamp down the lines of codes for the web page or web application. Since these are light-weight in syntax , they can be very easily used in anonymous.

### 1.1.9 New Array Functions

Through array functions are not necessary for any programming language ,they do simplify things for the developer . This also compacts the code and makes it much easier to understand .A regular array and an associative array,JavaScript supports them both.While a regular array contains integer values for its index,indexes can be string for an associative array.

### 1.1.10 Default Parameters

This JavaScript features helps to avoid collapsing the whole code for a simple mistake.It is very useful when the developer needs to check the working of a function without any parameters.

### 1.1.11 Property Shorthand

Built-in methods like .get() are available for the developers's use.These methods help avoid writing the same code every time and cut back on various lines of code.These inborn methods are really supportive of cutting back developing time and cost.

### 1.1.12 Let/Const

Javascript has introduced the keywords **let** and **const** that are available to replace **var** .Unlike 'var', they are important due to their blocked scope i.e we can only access them in the block we defined them in. Where 'var',even if we initialize it inside a function,we can access it outside of the function...

### 1.1.13 Territory Presidency

This feature is used to create multiple territory models.

#### 1. 1.1.14 Lead Presidency

This features initiates and tracks the leads that are in progress.

## 1.2 PROJECT SCOPE

The following documentation is a project the “Breaking the bricks “.It describe the drawback of the old system and how the new proposed system overcomes these shortcomings.

The new system takes into account the various factors while designing a new system.It keeps into the account the economical bandwidth available for the new system.The foremost thing that is taken care of its need and requirements of the user.

It is basically for the user with the new trend of managing the stuffs.

- It is time saving as it doesn't involve manual process for facing difficulties due to heavy rush and safe from infectious place.
- Accuracy in work.
- It is very user friendly.
- Easy to update information.
- Access to any information individually.
- Easy availability.
- Easy & fast retrieval of information.

1.

### **3 IDENTIFICATION OF NEED**

User need identification and analysis are concerned with what user needs rather than what he/she wants. Not until the problem has been identified, defined, and evaluated should the analyst think about solutions and whether the problem is worth working. This step intended to help the user and analyst understand the real problem rather than its symptoms. The user or the analyst may identify the need for a candidate system or for enhancement in the existing system.

An analyst is responsible for performing following tasks:

- Studied strength and weakness of the current system.
- Determined “what” must be done to solve the problem.
- Prepared a functional specifications document.

These modules are developed with the aim of reducing time, reducing manpower so that everything can be easily maintained and. The volume of work and complexity are increasing year by year. This system reduces complexity and time. Also provide availability 24\*7.

## 1. 4 PROBLEM STATEMENT

In the existing system all the work is done manually. This is chance of committing errors and it will take more time to perform or checkout any information. There are so many limitations in the existing system. So the existing system should be automatized. If the system is carried over manually, for everything it take more time. So it is difficult to take immediate decisions.

- In the traditional system, if you wish to analyze any record you have to turn pages many time.
- Existing systems are time consuming as it requires too much planning and so much human involvement.
- As it involves much human involvement, the cost of the system automatically gets increased.
- Existing systems require paper use, which isn't good for the environment.
- With too much human involvement, there are high chances of risk as well.
- There is too much of paper work too, which makes the tasks in the existing system, very tedious.

## 1.5

### HARDWARE / SOFTWARE USED IN PROJECT

#### 1.5.1 HARDWARE REQUIREMENT

Hardware	Configuration
Processor	Intel(R)core(TM)i5-7200UCPU @2.50GHz
Ram	4GB
Monitor	Normal

#### 1.5.2 SOFTWARE REQUIREMENT

Software	Configuration
Operating System	Windows10
Language	Javascript

## 1.5

### 3 SOME REQUIREMENTS

Performance Requirements:

To achieve good performance the following requirements must be satisfied

- Scalability: The ease with which a system or component can be modified to fit the problem area.
- Portability: The ease with which a system or component can be transferred from one hardware or software environment to another.
- Security: It is the ideal state where all information can be communicated across the internet / company secure from unauthorized persons being able to read it and/or manipulate it..
- Maintainability: The ease with which a software system or component can be modified to correct faults, improve performance, or other attributes, or adapt to a changed environment.
- Reliability: The ability of a system or component to perform its required functions under stated conditions for a specified period of time.
- Reusability: The degree to which a software module or other work product can be used in more than one computing program or software system.

Safety Requirements:

In case scenarios where data integrity can be compromised, measures should be taken to ensure that all changes are made before system is shutdown. The user must have a registered account to use all facility of the web application.



## 1.5

### 1.5.4 OTHER REQUIREMENTS

#### 1.5.4.1 JAVASCRIPT

JavaScript is the world's most popular programming language.

JavaScript is the programming language of the Web.

JavaScript is easy to learn.

This tutorial will teach you JavaScript from basic to advanced.

JavaScript is a text-based programming language used both on the client-side **and server-side that allows you to make web pages interactive**. Where HTML and CSS are languages that give structure and style to web pages, JavaScript gives web pages interactive elements that engage a user

JavaScript is **a simple and easy-to-learn programming language** as compared to other languages such as C++, Ruby, and Python. It is a high-level, interpreted language that can easily be embedded with languages like HTML

Key differences between Java and JavaScript: **Java is** an OOP programming language while Java Script is an OOP scripting language. Java creates applications that run in a virtual machine or browser while JavaScript code is run on a browser only.

JavaScript is **used in both Back End and Front End Development**. JavaScript is used across the web development stack. That's right: it's both front end and backend.

Javascript is easier to start with, but **it gets way way difficult than Java**. At edge cases, finding bugs in Javascript becomes very difficult. Java has a very powerful type system which makes complex projects easy. But, there's nothing in Javascript to regulate its powers.

## 1.5

While JavaScript is a step up from the most fundamental web development skills (languages like HTML and CSS, which can be learned in under a month), you can still expect to **learn JS basics in a matter of months**, not years—and that's whether you learn through online classes or teach yourself through book study.

**We can enable JavaScript in Google Chrome on your Windows 10 device** to fully experience websites and apps. JavaScript is a programming language used to create any content that moves, changes, or updates on your screen without having to refresh the page.

For those want to learn to program, one of the biggest advantages of JavaScript is that **it is all free. JavaScript code runs entirely within the browser.** This means that all you need to run JavaScript elements is a modern browser that has been updated to the newest available version.

**Back end** developers are focused on data, modeling, and the back end of a website. Front end developers help build what users interact with and see. A full stack developer does some or all of the above.

### 1.5.4.2 NODE .JS

Node. js is **a platform built on Chrome's JavaScript runtime 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 primarily used for **non-blocking, event-driven servers**, due to its single-threaded nature. It's used for traditional web sites and back-end API services, but was designed with real-time, push-based architectures in mind.

Node. js (Node) is an **open source development platform for executing JavaScript code server-side.** ... js is intended to run on a dedicated HTTP server and to employ a single thread with one process at a time. Node. js applications are event-based and run asynchronously.

## 1.6 PROJECT SCHEDULE

The objective of software project planning is to provide a framework that enables the manager to make reasonable estimates of resources, costs and schedule. These estimates are made within a limited time frame at the beginning of a software project and should be updated regularly as the project progresses. In addition, estimates should attempt to define "best case" and "worst case" scenarios so that project outcomes can be bounded.

The first activity in software project planning is the determination of software scope. Function and performance allocated to software during system engineering should be assessed to establish a project scope that is ambiguous and understandable at Presidency and technical levels. Software scope describes function, performance, constraints, interfaces and reliability.

During early stages of project planning, a microscopic schedule is developed. This type of schedule identifies all major software engineering activities and the product functions to which they are applied. As the project gets under way, each entry on the macroscopic schedule is refined into detailed schedule. Here specific software tasks are identified and scheduled.

Scheduling has following principles:

1. Compartmentalization: the project must be compartmentalized into a number of manageable activities and tasks.
2. Interdependency: the interdependencies of each compartmentalized activity or tasks must be determined.
3. Time allocation: each task to be scheduled must be allocated some number of work units.
4. Effort validation: every project has a defined number of staff members.
5. Defined responsibilities: every task that is scheduled should be assigned to a specific team member.
6. Defined outcomes: every task that is scheduled should have a defined outcome.

### 1.6.1 Pert chart

Program evaluation and review technique (pert) is a project scheduling method that is applied to software development.

Pert provide quantitative tool that allow the software planner to-Determine the critical path-the chain of tasks that determines the duration of the project; Establish “most likely” time estimates for individual tasks by applying statistical models; and

Calculate “boundary times” that defines a time “window” for a particular task.

Pert chart(program evolution review technique) for project-

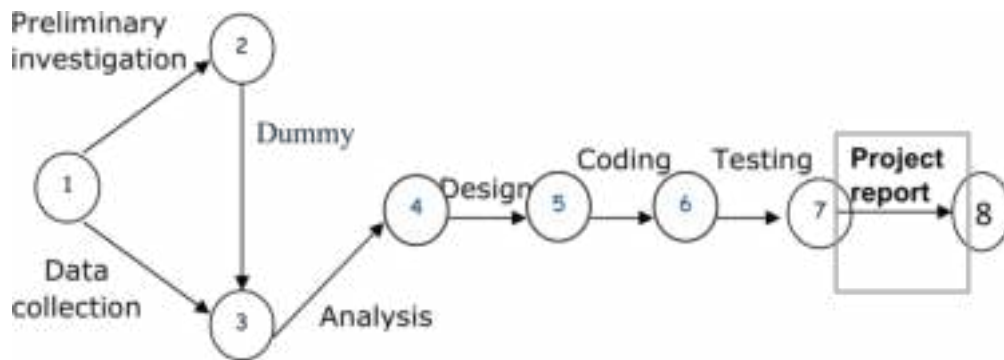


Figure 1.1 Pert chart

### 1.6.2 Gantt Chart

When creating a project schedule, the planner begins with a set of tasks (the work breakdown structure). If automated tools are used, the work breakdown is input as a task network. Effort, duration and start dates are input for each task network. As a consequence of this input, a timeline chart also called a Gantt chart is generated. A timeline chart is developed for entire project.

Gantt chart for project:

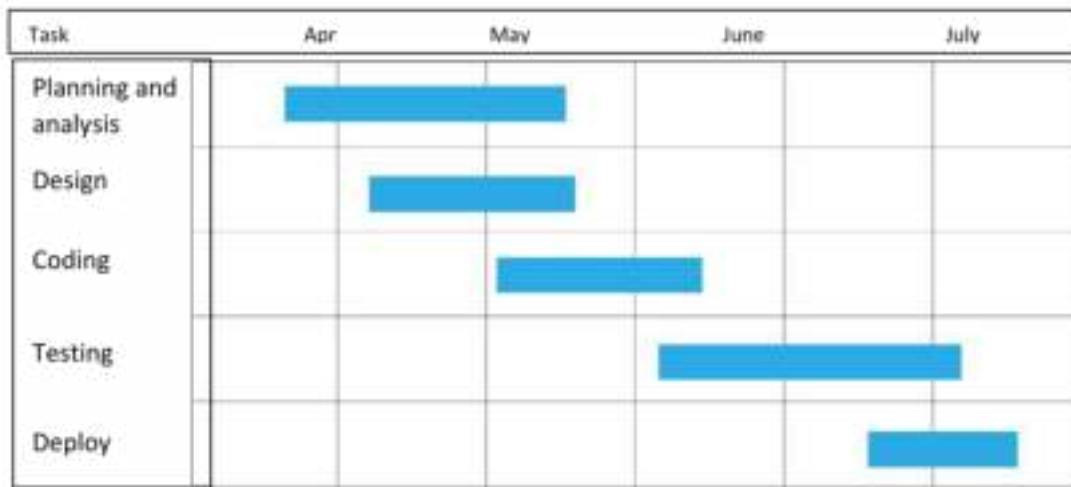


Figure 1.2:Gannt chart for project

Here horizontal bars indicate the duration of each task.

## CHAPTER 2

### LITERATURE REVIEW

#### ABSTRACT

Our Software is planned for the developing up an Breaking the bricks.that is very useful for the user that will help to manage all data of the user on the node.js with the help node.js and javascript .javascript is a text based programming language used both on the client-side and server-side that allows you to make web pages interactive.HTML and CSS are used to create web pages CSS(Cascading style sheets) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML.CSS is a cornerstone technology of the world wide web,alongside HTML and Javascript.

CSS stands for Cascading Style Sheets and it is used to style web documents .It is used to provide the background color and is also used for styling.It is responsible for the

designing or style of the website ,including the layout,visual effects and background color.Our game is available 24\*7.No need to install any extra software.We can do our work on node.js with ease and flexible as well as per availability.

Our project is for people which is HTML and CSS based with javascript flexibility.Our project is simple to access the information .For accessing this the user Should be logged with the system,after that they could change or manipulate data by the authorizations provided to the user.Breaking the bricks is based on javascript that provide giving information to the user.

#### KEYWORDS:-

Javascript,node.js,HTML,CSS

## 2.1 INTRODUCTION

The Aim of this system is to making an application which allow a normal user who is even familiar with the “

BREAKING THE BRICKS” is based on javascript ,a node.js application.

With the traditional database we use to store data in database respectively with any software,coding or

JavaScript is a lightweight , interpreted , or **just in time** compiled programming language with **first - class function**.While it is most well-known as the scripting language for web pages, **many non- browser environment** also use it, such as **Node.js, Apache CouchDB** and **Adobe Acrobat**.JavaScript is a prototype - based multi -paradigm,single-threaded,dynamic language,supporting object-

### Asynchronous JavaScript

It is most important and how it can be used to effectively handle potential

programming concept can use our system easily.[10]

manually,it’s quite hectic job.[15] that managing the database separately.but this time we have node-js storage mechanism which is easier and simple to manage the database,[13].

oriented,imperative and declarative styles.

### JavaScript building blocks

Continues our coverages of Javascript ‘s key fundamental features,turning our attention to commonly-encountered types of code blocks such as conditional statements ,loops,functions and events.

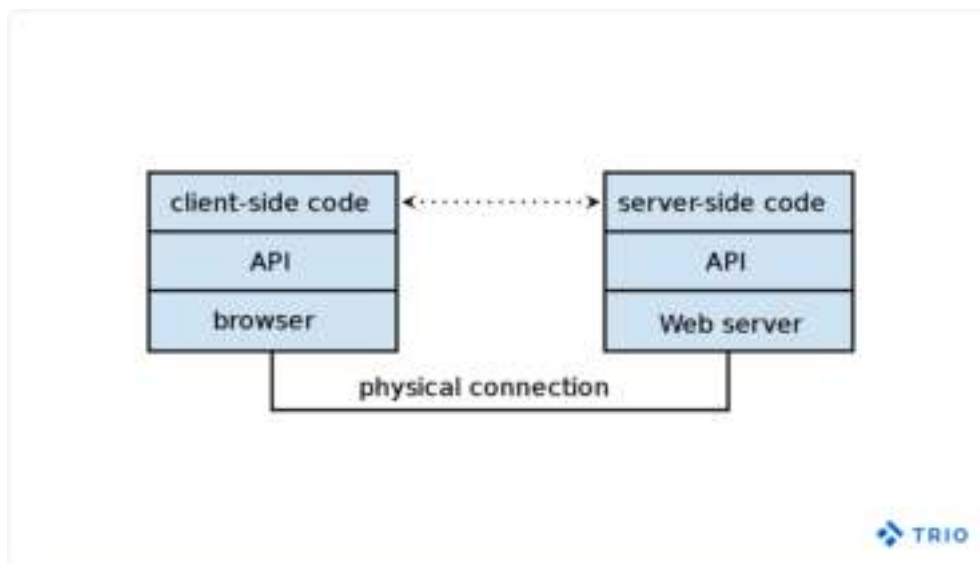
blocking operations such as fetching resources from a server

## 2.2 RELATED WORK

### 2.2.1 Speed and efficiency

It was previously established that JavaScript is a client-side language. Customers, of course, do not have the patience or

will to deal with lag. They come to your business to get their impulses assuaged and quickly at that.



### 2.2.2 Simplicity

Most developers consider JavaScript to be easy to learn and use. Keeping this in

mind, it's fairly easy to find software developers that code in JavaScript.



### 2.2.3 POPULARITY

A software framework is an abstraction in which software providing generic functionality can be selectively changed by additional user-written code.

JavaScript framework is an application framework written in JavaScript where the programmers can manipulate the functions and use them for their convenience.

Frameworks are more adaptable for the designing of websites, and hence, most website developers prefer them.

JavaScript frameworks are a type of tool that makes working with JavaScript easier and smoother. These frameworks also make it possible for the programmer to code the application as a device responsive. This responsiveness is yet another reason why the JavaScript frameworks are quite popular when it comes to the question of using a high-level machine language. Let's have a look at the best JS Frameworks in 2021.

load quickly, delivering automatic code-splitting. So users load the code required to render the view they request.

### 2.2.4 ANGULAR

#### Cross-Platform

- **Progressive Web Apps:** Modern web platform capabilities to deliver app-like experiences that are high performance, offline, and zero-step installation.
- **Native:** Strategies from Cordova, Ionic, or NativeScript are used to build the native app.
- **Desktop:** Desktop-installed apps across Mac, Windows, and Linux can be created using the same Angular methods like the web plus the ability to access native OS APIs.

#### Speed and Performance

- **Code Generation:** Angular turns templates into code that's highly optimized for JavaScript virtual machines, giving hand-written code benefits.
- **Universal:** Serve the first view of Node.js, .NET, PHP, and other servers for near-instant rendering in just HTML and CSS.
- **Code Splitting:** With the new Component Router, angular apps

#### Productivity

- **Templates:** Create UI views with simple and powerful template syntax.
- **Angular CLI:** Command line tools: start building fast, add components and tests, then instantly deploy.
- **IDEs:** Get intelligent code completion, instant errors, and other feedback in popular editors and IDEs.

#### Full Development Story

- **Testing:** Protractor makes your scenario tests run faster and in a stable manner.
- **Animation:** Create high-performance, complex choreographies, and animation timelines with very little code through Angular's intuitive API.
- **Accessibility:** Create accessible applications with ARIA-enabled components, developer guides, and built-in a11y test infrastructure.

### 2.3.2.1

JavaScript is among the most powerful and flexible programming languages of the web. It powers the dynamic behavior on most websites, including this one.

The standards for JavaScript are the [ECMAScript Language Specification](#) (ECMA-262) and the [ECMAScript Internationalization API specification](#) (ECMA-402). The JavaScript documentation throughout MDN is based on the latest draft versions of ECMA-262 and ECMA-402. And in cases where some [proposals for new ECMAScript features](#) have already been implemented in browsers,

documentation and examples in MDN articles may use some of those new features.

The object-oriented nature of JavaScript is important to understand if you want to go further with your knowledge of the language and write more efficient code, therefore we've provided this module to help you.

A much more detailed guide to the JavaScript language, aimed at those with previous programming experience either in JavaScript or another language

### 2.3.2.2 NODE.JS

**Node.js®** is a JavaScript runtime built on Chrome's V8 JavaScript engine.

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,<sup>[6]</sup> unifying [web-application](#) development around a single programming language, rather than different languages for server-side and client-side scripts.

Though `.js` is the standard [filename extension](#) for JavaScript code, the name "Node.js" doesn't refer to a particular file in this context and is merely the name of the product. Node.js has an [event-driven architecture](#) capable of [asynchronous I/O](#). These design choices aim to optimize [throughput](#) and [scalability](#) in web applications with many input/output operations, as well as for [real-time Web](#) applications (e.g., [real-time communication](#) programs and [browser games](#)).<sup>1</sup>

Node.js allows the creation of [Web servers](#) and networking tools using [JavaScript](#) and a collection of "modules" that handle various core functionalities.<sup>[29][32][46][47][48]</sup> 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.<sup>[32][47][49]</sup> Node.js's modules use an API designed to reduce the complexity of writing server applications.<sup>[32][47]</sup>

JavaScript is the only language that Node.js supports natively, but many [compile-to-](#)

[JS](#) languages are available.<sup>[50]</sup> As a result, Node.js applications can be written in [CoffeeScript](#),<sup>[51]</sup> [Dart](#), [TypeScript](#), [ClojureScript](#) and others.

Node.js is primarily used to build network programs such as Web servers.<sup>[46]</sup> The most significant difference between Node.js and [PHP](#) is that most functions in PHP [block](#) until completion (commands execute only after previous commands finish), while Node.js functions are [non-blocking](#) (commands execute [concurrently](#) or even in [parallel](#),<sup>[52][53]</sup> and use [callbacks](#) to signal completion or failure).<sup>[46]</sup>

## Platform architecture

Node.js brings [event-driven programming](#) to [web servers](#), enabling development of fast web servers in JavaScript.<sup>[32]</sup> Developers can create scalable servers without using [threading](#), by using a simplified model of [event-driven programming](#) that uses callbacks to signal the completion of a task.<sup>[32]</sup> Node.js connects the ease of a scripting language (JavaScript) with the power of Unix network programming.<sup>[32]</sup>

Node.js was built on top of [Google's V8 JavaScript engine](#) since it was open-sourced under the [BSD license](#). It is proficient with internet fundamentals such as [HTTP](#), [DNS](#), and [TCP](#).<sup>[29]</sup> [JavaScript](#) was also a well-known language, making Node.js accessible to the [web development community](#).<sup>[29]</sup>

There are thousands of open-source libraries for Node.js, most of them hosted on the [npm](#) website. There are multiple developer conferences and events that support the Node.js community, including NodeConf, Node Interactive, and Node Summit as well as a number of regional events.

The open-source community has developed [web frameworks](#) to accelerate the development of applications. Such frameworks include

Connect, [Express.js](#), [Socket.IO](#), [Feathers.js](#), [Koa.js](#), [Hapi.js](#), [Sails.js](#), [Meteor](#), [Derby](#), and many others.<sup>[32][56]</sup> Various packages have also been created for interfacing with other languages or runtime environments such as [Microsoft .NET](#).<sup>[57]</sup>

Modern desktop [IDEs](#) provide editing and debugging features specifically for Node.js applications. Such IDEs include [Atom](#), [Brackets](#), [JetBrains WebStorm](#),<sup>[58][59]</sup> [Microsoft Visual Studio](#) (with Node.js Tools for Visual Studio,<sup>[60]</sup> or [TypeScript](#) with Node definitions,<sup>[61][62][63][64]</sup> [NetBeans](#),<sup>[65]</sup> [Nodeclipse](#) [Eclipse Studio](#)<sup>[66]</sup> ([Eclipse](#)-based), and [Visual Studio Code](#).<sup>[67][68]</sup> Certain online web-based IDEs also support Node.js, such as [Codeanywhere](#), [Codenvy](#), [Cloud9 IDE](#), [Koding](#), and the visual flow editor in [Node-RED](#).



## 2.4 RESULT

We have come on result that our system working fine in each case.It is user friendly as

well as efficient to use. We have done different things to verify the performance

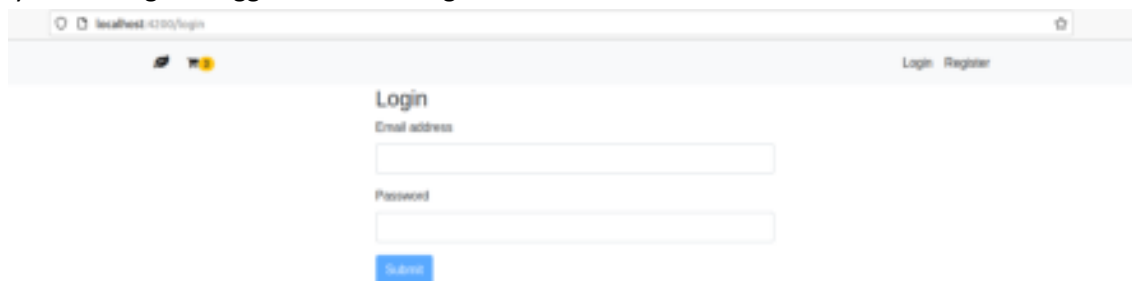
### TEST CASE RESULT

TEST CASE	DESCRIPTION	RESULT
TC#1	Loading the Login	Passed
TC#2	Readme	Passed
TC#3	Script.js	Passed
TC#4	Style.js	Passed
TC#5	index.html	Passed
TC#6	images	Passed
TC#7	Sound	Passed
TC#7	Logout	Passed

## 2.5 DISCUSSION

System was completely done after was duly coded. each module of the project were checked to ensure they are fully functional unit. This was done by checking each unit to give assurance that it functions as required and that it performed exactly as defined. The success of each individual gave us go ahead to carryout testing properly[9]. The defined system was validated by the using a series of short questionnaire that was completely filled by representatives users who have used the system and give suggestion according to the

need. That was completely filled by representatives users who have used the system and give suggestions according to the need. This was done to the access if the system and give suggestions according to the need. This was done to the assess if the system met their respective needs and their requirements. It was also find that it is easy to access the data as well as available when needed. With the flexibility of the javascript it is quite useful and better version of management system



2.6

## CONCLUSION & FUTURE SCOPE

### 2.6.1 CONCLUSION

A system means a lot of experience. I learned a lot of things in this project development. This

project has also sharpened my concept about javascript, node.js, html and css

### 2.6.2 FUTURE SCOPE

To make our      3.      system more user friendly.Iwill add helping BOT in the system .

## 2.7 REFERENCES

[1] HOW TO BUILD A GAME WITH HTML,CSS AND JAVASCRIPT(<https://blog.logrocket.com/build-a-game-with-html-css-javascript/>)

[2]WRITING A MULTIPLAYER TEXT ADVENTURE ENGINE IN NODE.JS:GAME ENGINE SERVER DESIGN(<https://www.smashingmagazine.com//>)

[3]VISUAL -js- game engine([https://developer.mozilla.org/en-US/docs/Game/Visual-js\\_game\\_engine//](https://developer.mozilla.org/en-US/docs/Game/Visual-js_game_engine//))

[4]HOW TO MAKE YOUR OWN ONLINE GAME WITH NODE.js(<https://www.skysilk.com/blog/2018/online-javascript-game-tutorial//>)

[5]USE OF JAVASCRIPT IN ONLINE GAMES(<https://dzone.com/articles/use-of-javascript-in-online-games//>)

[6]LEARN THE JAVASCRIPT LANGUAGE IF YOU WANT TO ADVANCE YOUR SKILLS IN GAME DEVELOPMENT(<https://www.gamedesigning.org/learn/javascript//>)

[7]2D BREAKOUT GAME USING PURE  
JAVASCRIPT([https://developer.mozilla.org/en-  
US/docs/tutorial/2D Breakout game pure Javascript//](https://developer.mozilla.org/en-US/docs/tutorial/2D_Breakout_game_pure_Javascript//))

[8]MODELING PLAYER EXPERIENCE IN SUPER MARIO BROS(SOURCE:IEEE Xplore)

[9]MARIO GAME IN PYTHON WITH SOURCE CODE(itsourcecode.com)

[10]Adding Page Break Inside PDF Document in Javascript full  
example([https://codingshiksha.com/javascript/html-full-example-with-source-  
code-2021//](https://codingshiksha.com/javascript/html-full-example-with-source-code-2021//))

[11]MODELLING PLAYER EXPERIENCE IN SUPER MARIO BROS(by Georgious  
Yannakakis)



## CHAPTER 3

### FEASIBILITY STUDY

#### 3.1 INTRODUCTION

Feasibility of the system is an important aspect, which is to be considered. The system needs to satisfy the law of economic, which states that the maximum output should be yielded in minimum available resources.

A feasibility analysis evaluates the project's potential for success; therefore, perceived objectivity is an essential factor in the credibility of the study for potential investors and lending institutions. There are five types of feasibility study—separate areas that a feasibility study examines, described below.

##### 1. Technical Feasibility

This assessment focuses on the technical resources available to the organization. It helps organizations determine whether the technical resources meet capacity and whether the technical team is capable of converting the ideas into working systems. Technical feasibility also involves the evaluation of the hardware, software, and other technical requirements of the proposed system. As an exaggerated example, an organization wouldn't want to try to put Star Trek's transporters in their building—currently, this project is not technically feasible.

##### 2. Economic Feasibility

This assessment typically involves a cost/ benefits analysis of the project, helping organizations determine the viability, cost, and benefits associated with a project before financial resources are allocated. It also serves as an independent project assessment and enhances project credibility—helping decision-makers determine the positive economic benefits to the organization that the proposed project will provide.

### 3. Legal Feasibility

This assessment investigates whether any aspect of the proposed project conflicts with legal requirements like zoning laws, data protection acts or social media laws. Let's say an organization wants to construct a new office building in a specific location. A feasibility study might reveal the organization's ideal location isn't zoned for that type of business. That organization has just saved considerable time and effort by learning that their project was not feasible right from the beginning.

### 4. Operational Feasibility

This assessment involves undertaking a study to analyze and determine whether—and how well—the organization's needs can be met by completing the project. Operational feasibility

studies also examine how a project plan satisfies the requirements identified in the requirements analysis phase of system development.

### 5. Scheduling Feasibility

This assessment is the most important for project success; after all, a project will fail if not completed on time. In scheduling feasibility, an organization estimates how much time the project will take to complete.

When these areas have all been examined, the feasibility analysis helps identify any constraints the proposed project may face, including:

- Internal Project Constraints: Technical, Technology, Budget, Resource, etc.
- Internal Corporate Constraints: Financial, Marketing, Export, etc.
- External Constraints: Logistics, Environment, Laws, and Regulations, et

### 3. 2 MAIN ASPECTS

There are three aspects of feasibility to be considered namely.

1. Technical
2. Operational
3. Economical

#### TECHNICAL:-

In the technical aspects one may consider the hardware equipment for the installation of the software. The system being centralized will required very little hardware appliances. Hence this helps the system to work smoothly with limited amount of working capitals.

#### OPERATIONAL:-

In the operational aspects may think of the benefits of the workload that many a personal may have to share. This is eased out and the required output may be retrieved in a very short time. Thus there is accuracy in the work on time is also saved there will be very little work that needs to be performed.

#### ECONOMICAL:-

Economical system is definitely feasible because the hardware requirement is less and the operational working for the system requires less number of recruits. This help introduction over-staffing and wastage funds.

We studied on the position to evaluate solution. Most important factors in this study were tending to overlook the confusion inherent in system Development the constraints and the assumed studies. It can be started that it the feasibility study is to serve as a decision document it must answer three key questions.

1. Is there a new and better way to do the job that will benefit the user?
2. What are the costs and savings of the alternatives?
3. What is recommended?

On these questions it can be explained that feasibility study of the system includes following different angles.

3.

#### **2.1 Technical feasibility:**

This centers on the existing computer system (hardware, software etc.) and to what extent it can support the proposed additional equipment .in this stage of study, we have collected information about technical tools available by which I could decide my system design as the technical requirements.

#### **3.2.2 Operational Feasibility:**

In this stage of study we have checked the staff availability. I concentrate on knowledge of end users that are going to use the system. This is also called as behavioral feasibility in which I have studied on following aspects; people are inherently resistant to change, and computers have been known to facilitate change .An estimate has been made to how strong a reaction the user staff is having toward the development of a computerized system. It is common knowledge that computer installations have something to do with turnover. I had explained that there is need to educate and train the staff on new ways of conducting business.

#### **3.2.3 Economical feasibility:**

Economical analysis is the most frequently used method for evaluating the effectiveness of candidate system. More commonly known as cost\benefit analysis, the procedure is to determine the benefits and savings that benefits outweigh costs. The decision was to design and implement system because it is for having chanced to be approved. This is an on going effort that improves the accuracy at each phase of the system life cycle.

In developing cost estimates for a system I need to consider several cost elements. Among these is hardware personal facility. Operating and supply costs.

3.

### **3 BENEFITS**

Benefits of conducting a feasibility study:

- Improves project teams' focus
- Identifies new opportunities
- Provides valuable information for a "go/no-go" decision
- Narrows the business alternatives
- Identifies a valid reason to undertake the project
- Enhances the success rate by evaluating multiple parameters
- Aids decision-making on the project
- Identifies reasons not to proceed

3.

#### 4 SYSTEM REQUIREMENT SPECIFICATION

Any system can be designed after specifies the requirement of the user about that system. For this first of all gathered information from user by the preliminary investigation which is starting investigation about user requirement..

The data that the analysts collect during preliminary investigation are gathered through the various preliminary methods.

##### Documents Reviewing Organization

The analysts conducting the investigation first learn the organization involved in, or affected by the project. Analysts can get some details by examining organization charts and studying written operating procedures.

Collected data is usually of the current operating procedure:

- The information relating to clients, projects and students and the relationship between them was held manually.
- Managing of follow-ups was through manual forms.
- Complaints require another tedious work to maintain and solve.
- Payments details had to be maintained differently.

##### Gathering Information By Asking Questions

Interviewing is the most commonly used techniques in analysis. It is always necessary first to approach someone and ask them what their problems are, and later to discuss with them the result of your analysis.

##### Questionnaires

Questionnaires provide an alternative to interviews for finding out information about a system. Questionnaires are made up of questions about information sought by analyst. The questionnaire is then sent to the user, and the analyst analyzes the replies.

##### Electronic Data Gathering

Electronic communication systems are increasingly being used to gather information. Thus it is possible to use electronic mail to broadcast a question to a number of users in an organization to obtain their viewpoint on a particular issue.

In my project, with the help of Marg software solutions, I have send questionnaire through electronic mail to twenty employees of the company and retrieved the information regarding the problem faced by existing system.

##### Interviews

Interview allows the analysts to learn more about the nature of the project request and reason of submitting it. Interviews should provide details that further explain the project and show whether assistance is merited economically, operationally or technically.

One of the most important points about interviewing is that what question you need to ask.

It is often convenient to make a distinction between three kinds of question that is

- Open questions
- Closed question
- Probes

Open questions are general question that establish a persons view point on a particular subject.

Closed questions are specific and usually require a specific answer.

Probes are question that follow up an earlier answer.

## **CHAPTER 4**

### **DESIGN**

#### **4.1 INTRODUCTION**

System is created to solve problems. One can think of the systems approach as an organized way of dealing with a problem. In this dynamic world, the subject system analysis and design, mainly deals with the software development activities.

Since a new system is to be developed, the one most important phases of software development life cycle is system requirement gathering and analysis. Analysis is a detailed study of various operations performed by a system and their relationship within and outside the system. Using the following steps it becomes easy to draw the exact boundary of the new system under consideration.

All procedures, requirements must be analysed and documented in the form of detailed DFDs, logical data structure and miniature specifications.

System analyses also include sub-dividing of complex process involving the entire system, identification of data store and manual processes.



## 4. SYSTEM DESIGN

System design is the process of planning a new system or to replace the existing system. Simply, system design is like the blueprint for building, it specifies all the features that are to be in the finished product.

System design phase follows system analysis phase. Design is concerned with identifying functions, data streams among those functions, maintaining a record of the design decisions and providing a blueprint the implementation phase.

Design is the bridge between system analysis and system implementation. Some of the essential fundamental concepts involved in the design of application software are:

- Abstraction
- Modularity
- Verification

**Abstraction** is used to construct solutions to problem without having to take account of the intricate details of the various component sub problems. Abstraction allows system designer to make step-wise refinement, which at each stage of the design may hide, unnecessary details associated with representation or implementation from the surrounding environment.

**Modularity** is concerned with decomposing of main module into well-defined manageable units with well-defined interfaces among the units. This enhances design clarity, which in turn eases implementation, Debugging, Testing, Documenting and Maintenance of the software product. Modularity viewed in this sense is a vital tool in the construction of large software projects.

**Verification** is fundamental concept in software design. A design is verifiable if it can be demonstrated that the design will result in implementation that satisfies the customer's requirements. Verification is of two types namely,

- Verification that the software requirements analysis satisfies the customer's needs.
- Verification that the design satisfies the requirement analysis.

Some of the important factors of quality that are to be considered in the design of application software are:

### **Reliability:**

The software should behave strictly according to the original specification and should function smoothly under normal conditions.

### **Extensibility:**

The software should be capable of adapting easily to changes in the specification.

### **Reusability:**

The software should be developed using a modular approach, which permits modules to be reused by other application, if possible.

The System Design briefly describes the concept of system design and it contains four sections. The first section briefly describes the features that the system is going to provide to the user and the outputs that the proposed system is going to offer.

The second section namely Logical Design describes the Data Flow Diagrams, which show clearly the data movements, the processes and the data sources, and sinks, E-R diagrams which represent the overall logical design of the database, and high-level process structure of the system.

#### **Preliminary Design:**

Preliminary design is basically concerned with deriving an overall picture of the system. Deriving entire system into modules and sub-modules while keeping Cohesion and Coupling factors in mind. Tools, which assist in preliminary design process, are Data Flow Diagrams.

#### **Code design:**

The purpose of code is to facilitate the identification and retrieval for items of information. A code is an ordered collection of symbols designed to provide unique identification of an entity or attribute. To achieve unique identification there must be only one place where the identified entity or the attribute can be entered in the code; conversely there must be a place in the code for every thing that is to be identified. This mutually exclusive feature must be built into any coding system.

The codes for this system are designed with two features in mind. Optimum human oriented use and machine efficiency They are also operable i.e., they are adequate for present and anticipate data processing both for machine and human use.

#### **Input /Output design :**

is a part of overall system design, which requires very careful attention. The main objectives of input design are:

- To produce a cost-effective method of input.
- To achieve the highest possible level of accuracy.
- To ensure that the input is acceptable to and understood by the user staff.

Outputs from computer systems are required primarily to communicate the results of processing to users. They are also to provide a permanent hard copy of these results for later consultation.

The various types of outputs are required by this system are given below:

- External outputs, whose destination is outside the concern and which require special attention because they, project the image of the concern.
- Internal outputs, whose destination is within the concern and which require careful design because they are the user's main interface within the computer.
- Operation outputs, whose use is purely within the computer department, E.g., program listings, usage statistics etc,

#### 4. S 3 DLC

Software Development Life Cycle (SDLC) is a framework that defines the steps involved in the development of software at each phase. It covers the detailed plan for building, deploying and maintaining the software.

SDLC defines the complete cycle of development i.e. all the tasks involved in planning, creating, testing, and deploying a Software Product.



Figure 4.1: Above image depicting the planning step

#### SDLC Phases

Given below are the various phases:

- Requirement gathering and analysis
- Design
- Implementation or coding
- Testing
- Deployment
- Maintenance

**Requirement Gathering and Analysis**

During this phase, all the relevant information is collected from the customer to develop a product as per their expectation. Any ambiguities must be resolved in this phase only.

Business analyst and Project Manager set up a meeting with the customer to gather all the information like what the customer wants to build, who will be the end-user, what is the purpose of the product. Before building a product a core understanding or knowledge of the product is very important.

Once the requirement gathering is done, an analysis is done to check the feasibility of the development of a product. In case of any ambiguity, a call is set up for further discussion.

Once the requirement is clearly understood, the SRS (Software Requirement Specification) document is created. This document should be thoroughly understood by the developers and also should be reviewed by the customer for future reference.

**Design**

In this phase, the requirement gathered in the SRS document is used as an input and software architecture that is used for implementing system development is derived.

**Implementation or Coding**

Implementation/Coding starts once the developer gets the Design document. The Software design is translated into source code. All the components of the software are implemented in this phase.

**Testing**

Testing starts once the coding is complete and the modules are released for testing. In this phase, the developed software is tested thoroughly and any defects found are assigned to developers to get them fixed.

**Deployment**

Once the product is tested, it is deployed in the production environment or first UAT (User Acceptance testing) is done depending on the customer expectation.

**Maintenance**

After the deployment of a product on the production environment, maintenance of the product i.e. if any issue comes up and needs to be fixed or any enhancement is to be done is taken care by the developers.

#### 4.4 S

##### **SOFTWARE ENGG. PARADIGM APPLIED**

S t

Software engineering is a layered technology. The foundation for software engineering is the process layer. Software engineering processes the glue that holds the technology layers together and enables rapid and timely development of computer software. Process defines a framework for a set of key process areas that must be established for effective delivery of software engineering technology.

Software engineering methods provide the technical how-to's for building software. Methods encompass a broad array of tasks that include requirements analysis, design, program construction, testing and support. Software engineering tools provide automated or semi-automated support for the process and the methods. When tools are integrated so that information created by one tool can be used by another tool, a system for the support of software development, called computer-aided software engineering is established.

**The following paradigms are available:**

1. The Waterfall Model
  2. The Prototyping Model
  3. The Spiral model
- Etc.

#### 4.4

##### .1 The Prototype model

The prototype model requires that before carrying out the development of actual software, a working prototype of the system should be built. A prototype is a toy implementation of the system. A prototype usually turns out to be a very crude version of the actual system, possibly exhibiting limited functional capabilities, low reliability, and inefficient performance as compared to actual software. In many instances, the client only has a general view of what is expected from the software product. In such a scenario where there is an absence of detailed information regarding the input to the system, the processing needs, and the output requirement, the prototyping model may be employed.

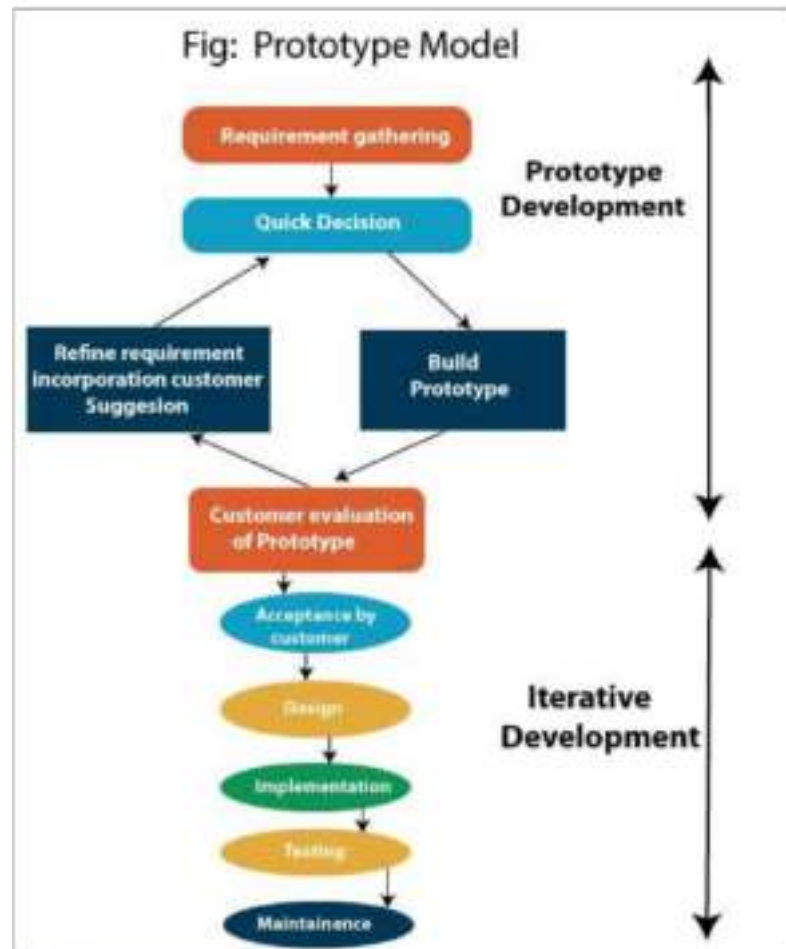


Figure 4.2: Prototype model

## 4.4

### 4.4.1 Advantage of Prototype Model

1. Reduce the risk of incorrect user requirement
2. Good where requirement are changing/uncommitted
3. Regular visible process aids Presidency
4. Support early product marketing
5. Reduce Maintenance cost.
6. Errors can be detected much earlier as the system is made side by side.

### 4.4.1.2 Disadvantage of Prototype Model

1. An unstable/badly implemented prototype often becomes the final product.
2. Require extensive customer collaboration
  - o Costs customer money
  - o Needs committed customer
  - o Difficult to finish if customer withdraw
  - o May be too customer specific, no broad market
3. Difficult to know how long the project will last.
4. Easy to fall back into the code and fix without proper requirement analysis, design, customer evaluation, and feedback.
5. Prototyping tools are expensive.
6. Special tools & techniques are required to build a prototype.
7. It is a time-consuming process.

#### 4. 5 DFD

**DFD** is the abbreviation for **Data Flow Diagram**. The flow of data of a system or a process is represented by DFD. It also gives insight into the inputs and outputs of each entity and the process itself. DFD does not have control flow and no loops or decision rules are present. Specific operations depending on the type of data can be explained by a flowchart. Data Flow Diagram can be represented in several ways.

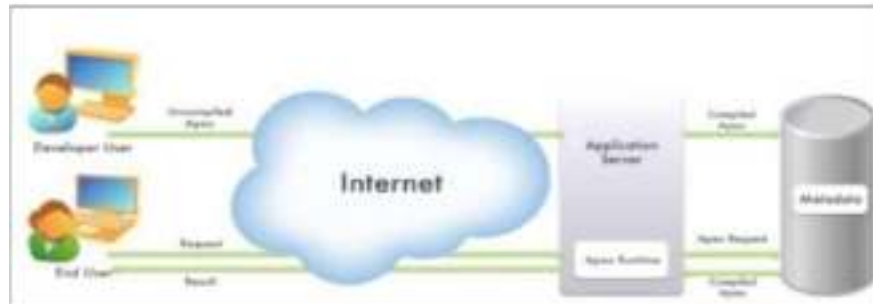


Figure 4.3 Apexworking



## 4.7 ER DIAGRAM

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

An ER diagram has three main components:

1. Entity
2. Attribute
3. Relationship

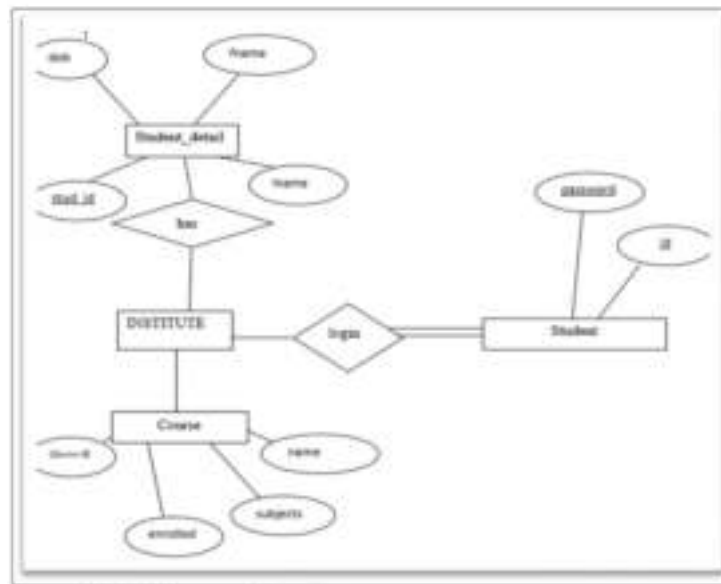


Figure 4.7:ER diagram of system

### 4.7.1 ER- Diagram Notations

ER- Diagram is a visual representation of data that describe how data is related to each other.

- **Rectangles:** This symbol represent entity types
- **Ellipses :** Symbolrepresent attributes
- **Diamonds:** This symbolrepresents relationship types
- **Lines:** It links attributes to entity types and entity types with other relationship types
- **Primary key:** attributes are underlined
- **Double Ellipses:** Represent multi-valued attributes

## CHAPTER 5

### REPORT

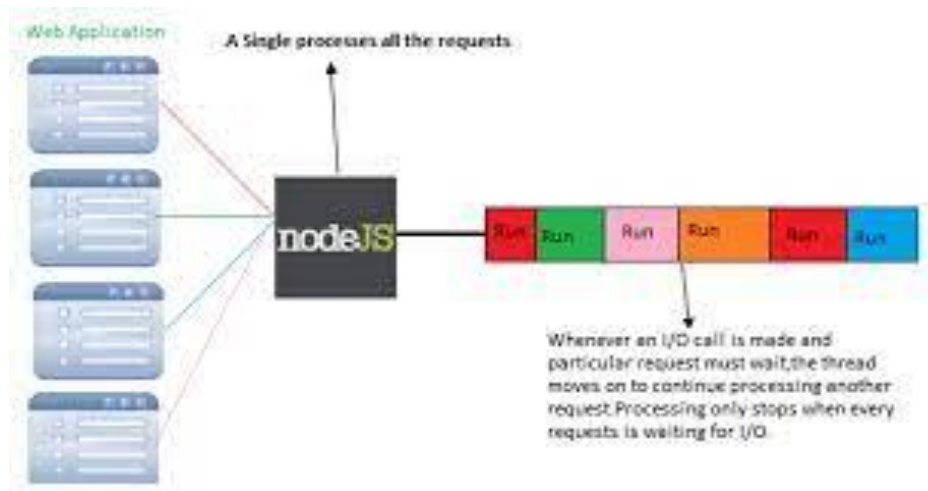
#### 5.1 GIST

the diagram figure 5.1 depicting our system

We have designed and developed an easy useful, reliable system



## 5.2 SOME SCREENSHOTS

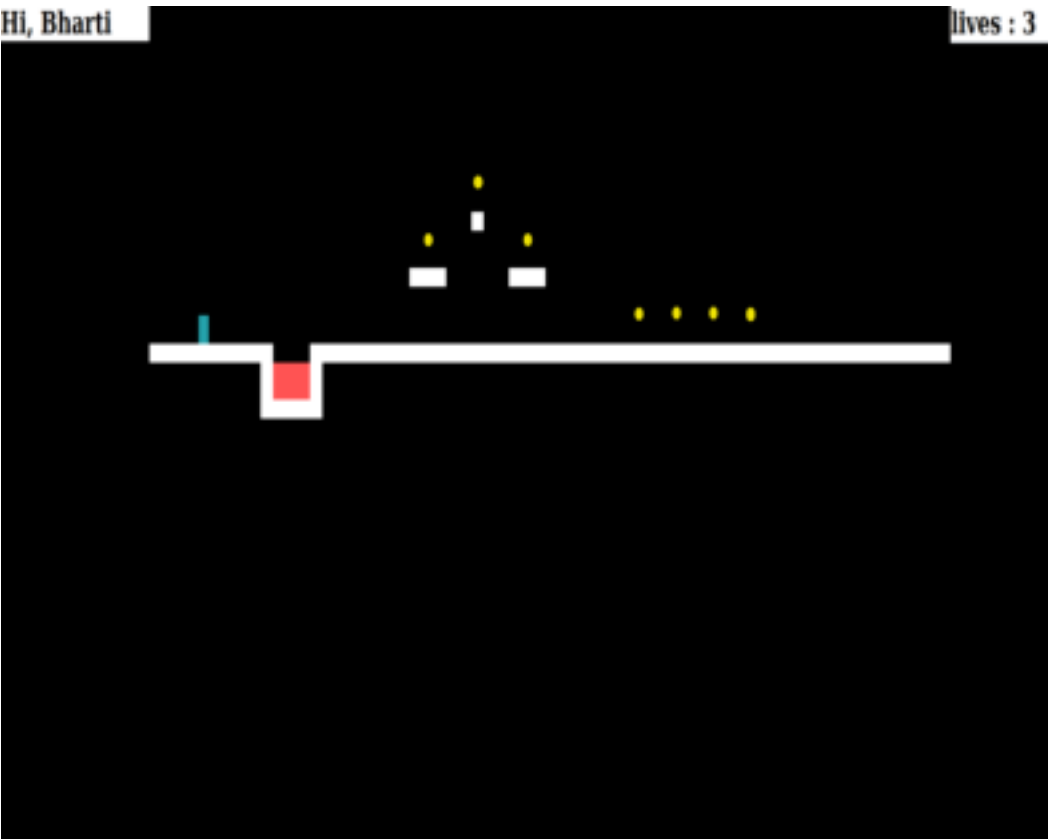


HOME PAGE

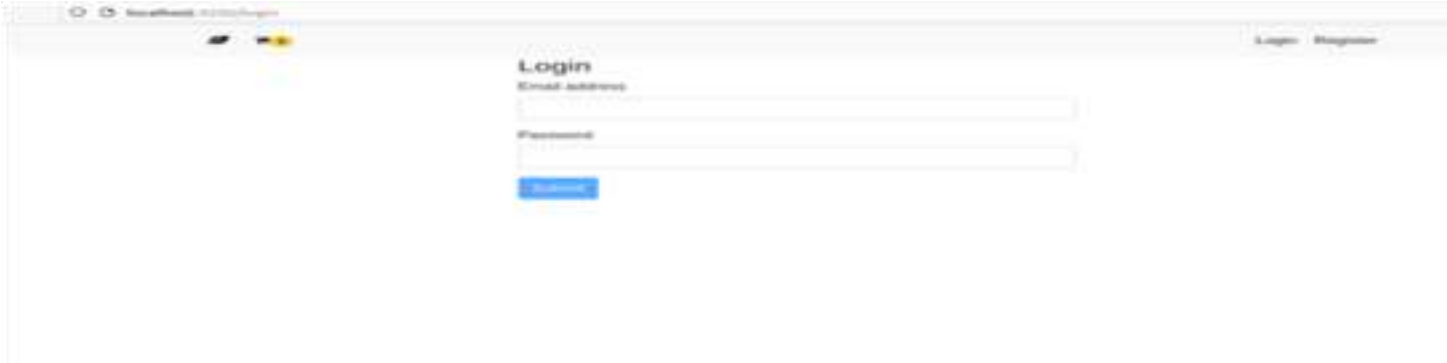


Hi, Bharti





LOGIN PAGE



## CHAPTER 6

### CODING

This chapter contains some codes of the project. The goal of the coding is to translate the design of the system into code in a given programming language. For a given design the aim of this phase is to implement the design in the best possible manner. The coding phase affects both testing and maintenance profoundly.

### SOME CODES ARE AS WRITTEN BELOW

```
const aim = document.querySelector('.aim');
const target = document.querySelector('.target');
const scoreText = document.querySelector('.score');
const timerText = document.querySelector('.timer');
const highscoreText = document.querySelector('.highscore');
const ClickToBegin = document.querySelector('.ClickToBegin');
const landingpage = document.querySelector('.landingpage');
const shootSound = document.querySelector('.shootSound') ;
const welcomeSound = document.querySelector('.welcomeSound') ;
const gameOverSound = document.querySelector('.gameoverSound');
const quackSound = document.querySelector('.quackSound');

var score = 0;
var timeLeft = 30;
var highscore = 0;

welcomeSound.currentTime=0;
welcomeSound.play();

const play =
()=>{
  setInterval(() =>
  { timer();
  }, 1000);
}
```

```
ClickToBegin.addEventListener('click', ()=>{
```



```
document.querySelector('.landingpage').style.display = "none";
```

```

if(localStorage.getItem('highscore'))
{
highscore = localStorage.getItem('highscore');
highscoreText.innerHTML = `High Score:
${highscore}`;
}
else{
highscoreText.innerHTML = `High Score: ${0}`;
}
scoreText.innerHTML = `Score: ${score}`
timerText.innerHTML = `${timeLeft} s Left`
spawnTarget();
play();
})

document.addEventListener('mousemove', (e) =>{
aim.style.left = `${e.clientX}px`
aim.style.top = `${e.clientY}px`

})

const spawnTarget = ()=>{
const top = Math.floor(Math.random() * (window.innerHeight-200))
const left = Math.floor(Math.random() * (window.innerWidth-120))
target.style.top = `${top}px`
target.style.left = `${left}px`
}

const gameOver = () =>{
timeLeft += 2;
alert(`\nGAME OVER\nSCORE: ${score}\n\nClick on "OK" or Press "ENTER" to play
again.`) if( score > localStorage.getItem('highscore'))
{
localStorage.setItem('highscore', score);
highscore = score;
highscoreText.innerHTML = `High Score: ${highscore}`;
}
window.location.reload();
}

const timer = () =>{
if(timeLeft === 0)
{
gameoverSound.currentTime=0;
gameoverSound.play();
gameOver();
}
}

```

```

}
timeLeft -= 1;
timerText.innerHTML = `${timeLeft} s Left`
}
function getRandomTarget(){
var randImg = new Array();
randImg[0] = "images/RedTopLeft.png";
randImg[1] = "images/RedRight.png";
randImg[2] = "images/RedLeft.png";
randImg[3] = "images/DuckTopRight.png";
randImg[4] = "images/DuckLeft.png";
var number = Math.floor(Math.random() * randImg.length);
target.querySelector('img').src = randImg[number];
quackSound.currentTime=0;
quackSound.play();
}

target.addEventListener('click', () =>{
getRandomTarget();
shootSound.currentTime = 0;
shootSound.play();
score +=1;
scoreText.innerHTML = `Score: ${score}`
spawnTarget();
})

```

## INDEX.HTML

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Duck Hunt</title>
<link rel="stylesheet" href="style.css">
</head>
<body>
<div class="landingpage">
<audio src="sounds/WelcomeSound.mp3" class="welcomeSound"></audio>
<div class="ClickToBegin">PLAY</div>
</div>

```

```

<div class="aim">

</div>

<div class="target">

</div>

<div class="score"></div>
<div class="timer"></div>

<div class="highscore"></div>

<audio src="sounds/QuackSound.mp3" class="quackSound"></audio>
<audio src="sounds/ShootSound.mp3" class="shootSound"></audio>
<audio src="sounds/GameOver.mp3" class="gameoverSound"></audio>

<script src="script.js"></script>
</body>
</html>

```

```

const aim = document.querySelector('.aim');
const target = document.querySelector('.target');
const scoreText = document.querySelector('.score');
const timerText = document.querySelector('.timer');
const highscoreText = document.querySelector('.highscore');
const ClickToBegin = document.querySelector('.ClickToBegin');
const landingpage = document.querySelector('.landingpage');
const shootSound = document.querySelector('.shootSound') ;
const welcomeSound = document.querySelector('.welcomeSound') ;
const gameoverSound = document.querySelector('.gameoverSound');
const quackSound = document.querySelector('.quackSound');

var score = 0;
var timeLeft = 30;
var highscore = 0;

welcomeSound.currentTime=0;
welcomeSound.play();

const play =
()=>{
  setInterval(() =>
  { timer();
  }, 1000);
}

```

```

ClickToBegin.addEventListener('click', ()=>{
document.querySelector('.landingpage').style.display = "none";
if(localStorage.getItem('highscore'))
{
highscore = localStorage.getItem('highscore');
highscoreText.innerHTML = `High Score: ${highscore}`;
}
else{
highscoreText.innerHTML = `High Score: ${0}`;
}
scoreText.innerHTML = `Score: ${score}`
timerText.innerHTML = `${timeLeft} s Left`
spawnTarget();
play();
})

document.addEventListener('mousemove', (e) =>{
aim.style.left = `${e.clientX}px`
aim.style.top = `${e.clientY}px`
})

const spawnTarget = ()=>{
const top = Math.floor(Math.random() * (window.innerHeight-200))
const left = Math.floor(Math.random() * (window.innerWidth-120))
target.style.top = `${top}px`
target.style.left = `${left}px`
}

const gameOver = () =>{
timeLeft += 2;
alert(`\nGAME OVER\nSCORE: ${score}\n\nClick on "OK" or Press "ENTER" to play
again.`) if( score > localStorage.getItem('highscore'))
{
localStorage.setItem('highscore', score);
highscore = score;
highscoreText.innerHTML = `High Score: ${highscore}`;
}
window.location.reload();
}

const timer = () =>{
if(timeLeft === 0)
{
gameoverSound.currentTime=0;
gameoverSound.play();
}
}

```

```

gameOver();
}
timeLeft -= 1;
timerText.innerHTML = `${timeLeft} s Left`
}
function getRandomTarget(){
var randImg = new Array();
randImg[0] = "images/RedTopLeft.png";
randImg[1] = "images/RedRight.png";
randImg[2] = "images/RedLeft.png";
randImg[3] = "images/DuckTopRight.png";
randImg[4] = "images/DuckLeft.png";
var number = Math.floor(Math.random() * randImg.length);
target.querySelector('img').src = randImg[number];
quackSound.currentTime=0;
quackSound.play();
}

target.addEventListener('click', () =>{
getRandomTarget();
shootSound.currentTime = 0;
shootSound.play();
score +=1;
scoreText.innerHTML = `Score: ${score}`
spawnTarget();
})

```

```
);
insertacct;
Contactc=newContact(
  AccountId=acct.Id,
  FirstName='Test',
  LastName='Contact'
);
insertc; Tasktsk=newTask(
  Subject='TestTask',
  WhoId=c.Id,
  Status='NotStarted'
);
inserttsk;
Eventevt=newEvent( Subject
='TestEvent', WhoId=c.Id,

  StartDateTime=Date.today().addDays(5),

  EndDateTime=Date.today().addDays(6)

);
insertevt;
```

```
Contactc=newContact(  
    AccountId=acct.Id,  
    FirstName='Test',  
    LastName='Contact'
```

```
);
```

```
insertc; Tasktsk=newTask(  
    Subject='TestTask',  
    WhoId=c.Id,  
    Status='Completed'
```



```
);  
inserttsk;  
Eventevt=newEvent( Subject  
='TestEvent', WhoId=c.Id,  
StartDateTime=Date.today().addDays(-6),  
EndDateTime=Date.today().addDays(-5)  
);  
insertevt; Casecse=newCase(  
Subject='TestCase',  
ContactId=c.Id,  
Status='Closed'  
);  
insertcse;  
List<Contact>contacts=ContactsTodayController.getContactsForToday();System.asse  
rtEquals(0,contacts.size());  
}  
}
```

```
List<Contact>contacts=ContactsTodayController.getContactsForToday();System
m.assertEquals(0,contacts.size());
}
}
```

```
System.assert(contacts[0].Description.containsIgnoreCase(tsk.Subject));System.ass
ert(contacts[0].Description.containsIgnoreCase(evt.Subject));
System.assert(contacts[0].Description.containsIgnoreCase(cse.Subject));
```

```
);
insertcse;
List<Contact>contacts=ContactsTodayController.getContactsForToday();Syste
m.assertEquals(0,contacts.size());
}
}
```

## **CHAPTER 7**

### **TESTING**

#### **7.1 INTRODUCTION**

Software Testing is a critical element of software quality assurance and represents the ultimate review of specification, design and code generation. The increasing visibility of software as a system element and the attendant "costs" associated with a software failure are motivating forces for well planned, thorough testing.

##### **7.1.1 Testing Objectives**

The following are the testing objectives:

- Testing is a process of executing a program with the intent of finding an error.
- A good test case is one that has a high probability of finding an as-yet-undiscovered error
- successful test is one that uncovers an as yet undiscovered error.

##### **7.1.2 Testing Principles**

The basic principles that guide software testing are as follows:

- All tests should be traceable to customer requirements.
- Tests should be planned long before testing begins.
- The separate principle applies to software testing.

Pareto principle states that 80 percent of all errors uncovered during testing will likely be traceable to 20 percent of all program components.

Testing should begin "in the small" and progress toward testing "in the large."

Exhaustive testing is not possible.

## 7.2 LEVEL OF TESTING

There are different levels of testing

->Unit Testing

->Integration Testing

->System Testing

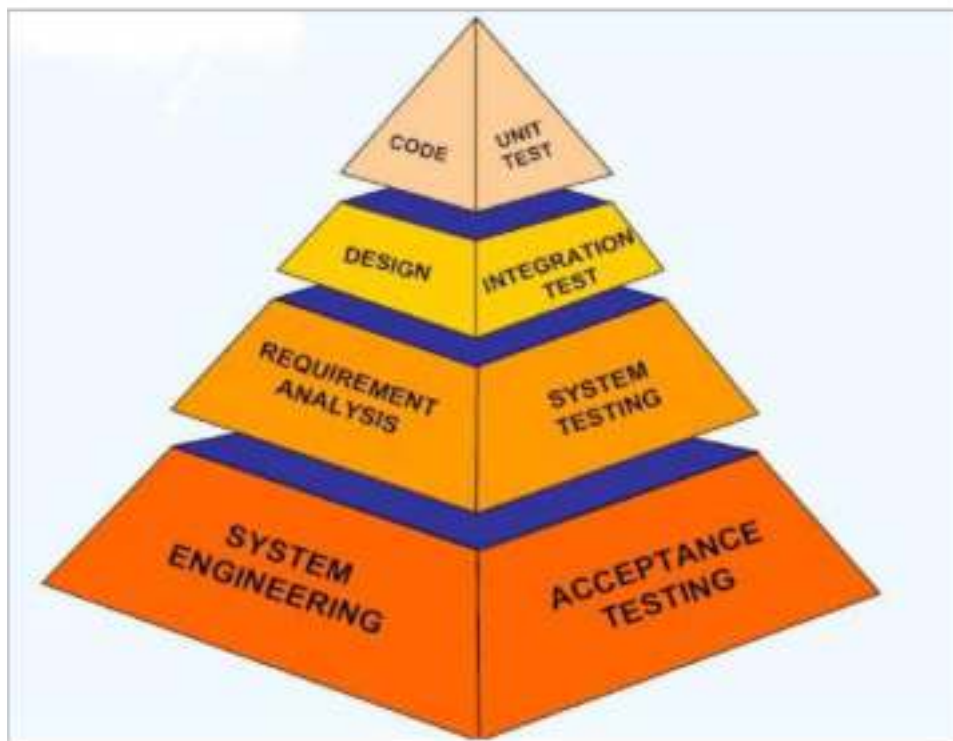


Figure 7.1:Testing pyramid

### 7.2.1 Unit testing

Unit testing focuses verification effort on the smallest unit of software design, the module. The important control parts are tested to uncover with in the boundary of the module. The module interface is tested to ensure that the information properly flows into and out of the program unit and boundary conditions are tested to ensure that the modules operate properly at boundaries established to limit or restrict processing. Test date is provided through testing screens.

### **7.2.2 Integration testing**

Integrating testing is a systematic technique for constructing Program structure while conducting tests to uncover error associates with interfacing .The objective is to take unit modules and built a program structure that has been directed by design.

- Integration Testing will test whether the modules work well together.
- This will check whether the design is correct.
- Integration can be done in 4 different ways:

### **7.2.3 System testing**

System testing is the process of testing the completed software as a part of the environment it was created for. It is done to ensure that all the requirements specified by the customer are met. System testing involves functional testing and performance testing.

- System Testing will contain the following testing :
  - Functional Testing.
  - Performance Testing.
- Function Testing will test the implementation of the business needs.
- Performance Testing will test the non-functional requirements of the system like the speed, load etc

## **7.3 SOME IMPORTANT OBSERVATION**

### **7.3.1 System Testing and Validation Results.**

System testing was done after the system was duly coded. Individual modules of the system were checked to ensure they are fully functional units before the integrating them. This was done by examining each unit; each script was checked to ensure that it functions as required and that it performed exactly as intended. The success of each individual unit gave us the go ahead to carryout integration testing.

The system was validated using a short questionnaire that was filled by representatives of the users who were let to interact with the system using test data and provided feedback about the system features. This was done to assess if the system met their needs and requirements as regards paying fees to the university. It was found out that the system performed in conformance to the then defined user needs and requirements. Results of the validation are shown as percentages of respondents against each requirement.

### 7.3.1 Testing Test Scenarios

1. Check if the page load time is within the acceptable range.
  2. Check the page load on slow connections.
  3. Check the response time for any action under a light, normal, moderate, and heavy load conditions.
  4. Check the performance of database stored procedures and trigger
  5. Check the execution time of the database
  6. Check for load testing of the application.
  7. Check for the Stress testing of the application.
  8. Check CPU and memory usage under peak load conditions.
- We have checked for scenarios and find that our system performing well in the circumstances.

## **TEST CASE Result SUMMARY**

<b>TEST CASE</b>	<b>DESCRIPTION</b>	<b>RESULT</b>
TC#1	Loading the login page	Passed
TC#2	Read me	Passed
TC#3	Script.js	Passed
TC#4	Style.js	Passed
TC#5	index.html	Passed
TC#6	images	Passed
TC#7	sound	Passed

## **CHAPTER 8**

### **CONCLUSION AND FUTURE SCOPE**

#### **8.1 CONCLUSION**

A software project means a lot of experience. I learned a lot through this project. This project has sharpened our concept node.js and javascript.

It provides online game and it has also the concepts of html and css.

It is much fast and more efficient so that the users can easily be available at every time

This project has given me an ample opportunity to design, code, test and implements an application. This has helped in putting into practice of various Software Engineering principles concepts like maintaining integrity and consistency of data.

#### **8.2 FUTURE SCOPE**

- The Future scope is to make the system more user friendly and enhanced.
- And we will make Mobile app for our system.
  - I will add Helping BOT in the system.
- Online gaming module would be introduced to conduct online game to users.



## BIBLIOGRAPHY

The books , which are referred and which really helped me in building this system in time, are as follows:-

Books::

FRONT PAGE	Daniel A. Tauber, Brenda Kienan
CUSTOMER RELATIONSHIP PRESIDENCY	Francis Buttle
SYSTEM ANALYSIS &DESIGN	Ellias M. Award
SOFTWARE ENGINEERING	Roger S. Pressman
LEARNING APEX PROGRAMMING	

Websites:

- [1] [https://developer.mozilla.org/en-US/docs/Games/Tutorials/2D\\_Breakout\\_game\\_pure\\_JavaScript](https://developer.mozilla.org/en-US/docs/Games/Tutorials/2D_Breakout_game_pure_JavaScript)
- [2] [https://eloquentjavascript.net/16\\_game.html](https://eloquentjavascript.net/16_game.html)
- [3] <https://www.makeschool.com/mediabook/oa/tutorials/build-a-game-of-concentration-with-javascript/client-side-javascript/>