Office Collaborator

A PROJECT REPORT
Submitted in partial fulfillment of the Requirementsfor the Degree of

MASTER OF COMPUTER APPLICATION

by

SHIVAM NERWAL
(Univ. Roll No.: 1900290140033)

Under the Supervision Of

Mrs. Vidushi
Assistant Professor
KIET Group of Institutions

Submitted to

DEPARTMENT OF COMPUTER APPLICATIONS
KIET Group of Institutions, Ghaziabad
Uttar Pradesh (201206)
JUNE, 2022

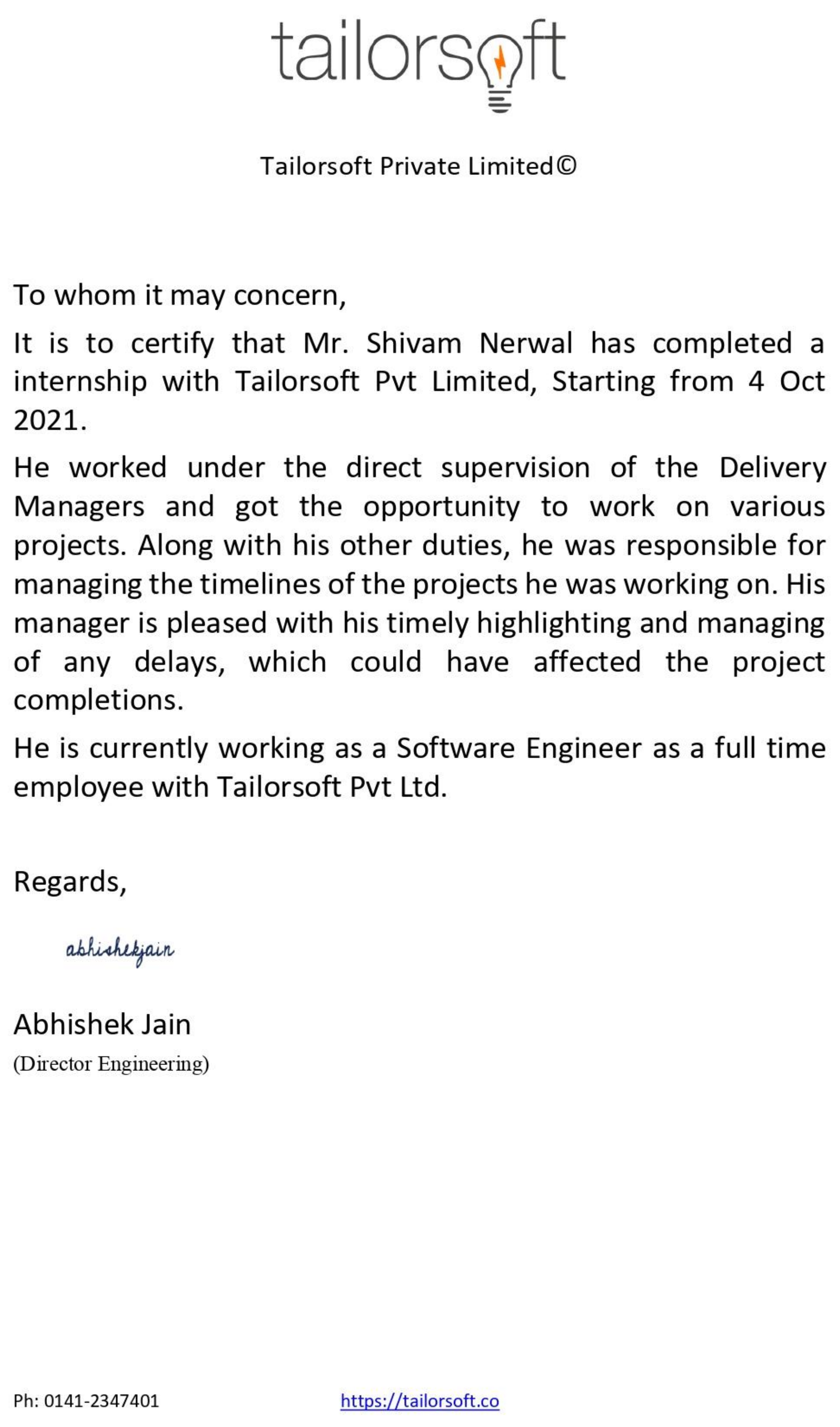
DECLARATION

I hereby declare that the work presented in this report entitled "Office Collaborator", was carried out by me. 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 results reported in the report are not manipulated. In the event of a complaint of plagiarism and the manipulation of the experiments and results, I shall be fully responsible and answerable.

Name: Shivam Nerwal
Roll.No.: 1900290140033
Branch: MCA

(Candidate Signature)

Training Certificate



Tailorsoft Private Limited©

To whom it may concern,

It is to certify that Mr. Shivam Nerwal has completed a internship with Tailorsoft Pvt Limited, Starting from 4 Oct 2021.

He worked under the direct supervision of the Delivery Managers and got the opportunity to work on various projects. Along with his other duties, he was responsible for managing the timelines of the projects he was working on. His manager is pleased with his timely highlighting and managing of any delays, which could have affected the project completions.

He is currently working as a Software Engineer as a full time employee with Tailorsoft Pvt Ltd.

Regards,

abhishekjain

Abhishek Jain

(Director Engineering)

Ph: 0141-2347401 https://tailorsoft.co

CERTIFICATE

Certified that **Shivam Nerwal** (enrollment no 190029014005176) has carried out the project work presented in this thesis entitled **“Office Collaborator”** for the award of **Master of Computer Application** from Dr. A P J Abdul Kalam Technical University, Lucknow under my supervision. The thesis embodies results of original work, and studies are carried out by the student himself and the contents of the thesis do not formthe basis for the award of any other degree to the candidate or to anybody else from this or any other University.

Date:

Ms. Vidushi
Assistant Professor
Department of Computer Applications
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

ACKNOWLEDGEMENT

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

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

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

Finally, my sincere thanks goes to my family members and all those who have directly and indirectly provided me moral support and other kinds 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.

Shivam Nerwal

Enrollment no-190029014005176

Table of Contents

LIST OF FIGURE

CHAPTER 1
INTRODUCTION

1.1 DESCRIPTION OF THE PROJECT

In this tech savvy era computer and technologies related to computer has become the gamechanger in every field we can precisely do a tremendous amount of work by applying our technical skills and knowledge. It's not incorrect to say that computer is transforming as well as giving new definition to the modern day challenges.

I pointed out a daily life problem of small scale as well as large scale organizations and first gathered all the requirements on basis of which I decided to further proceed my work to transform an Idea into a live working project, I am developing a web application name Office collaborator.

In Office collaborator the project provides two options first is to login as administrator and second one is login as employee, if any person do not having the credentials and wants to register himself the application provides the functionality to register himself as a new user (employee) in the application. We have a database which contains the information about all the enrolled managers and employees.

The administrator will able to add new employees into database, edit the information about newly added and existing employees in the database and administrator can also remove the existing employees form the database along with the functioning of administrator, I also provided various functionalities for user a user can search for its peers as well as a user will able to search the profiles of their managers, along with these I'm further planning to Improving the functioning as well as I'm planning to make the project to fulfill all the accessibility standards.

There is a detailed explanation below that how the user interface of this web application looks like and the screenshot are attached below to explain you the detailed functioning of the project.

Below is the first page of Office collaborator in which the first heading is Employee management system as it is the main function of the project to manage the data of various employees of the organization.

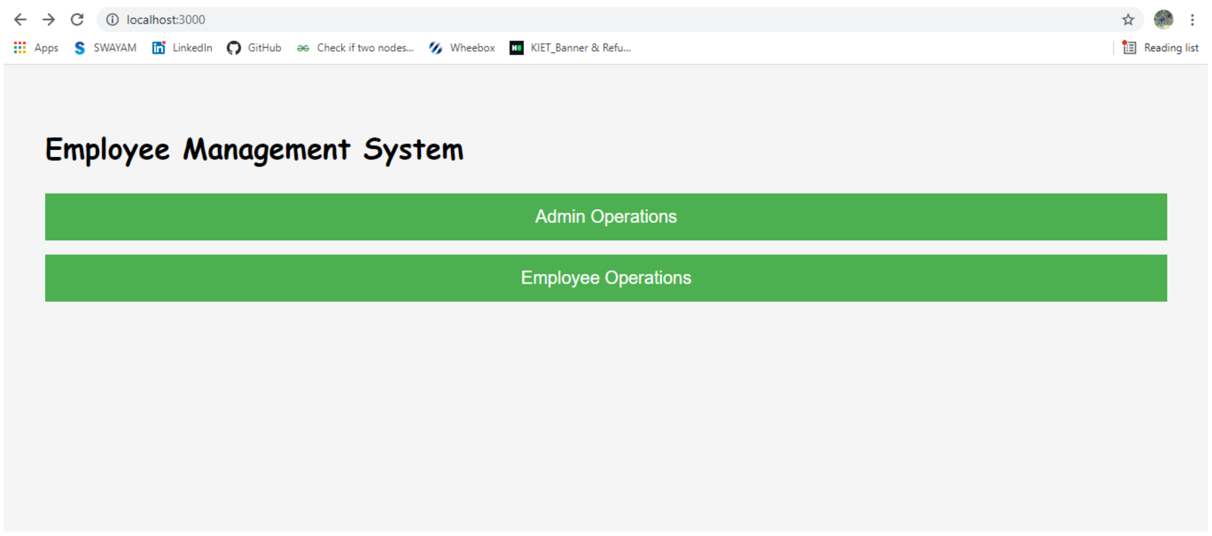


Figure 1.1 (Office collaborator first page)

There are two options provided first one is Admin operations this is for all the admin controls, admin control includes to enroll new employees and manage their information in the database.

The second provided option is Employee operation by this feature any newly enrolled employee will able to login and further use the application.

If the user clicks on admin operations he will be redirected to login as employee page (shown in figure 1.2 below), after entering correct credentials the user further redirected to employee dashboard (shown in figure 1.3 below) there are various available options like to search any existing employee in the organization, to edit your profile details and further to see the manager details.

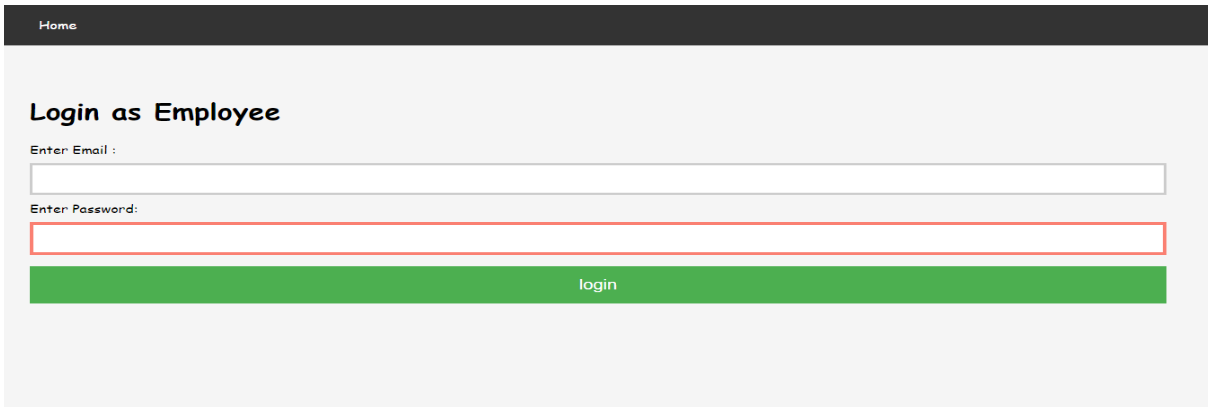


Figure 1.2 (Login as Employee)

If the user enters the valid credentials he will be redirected to Dashboard page with the various options shown in figure below

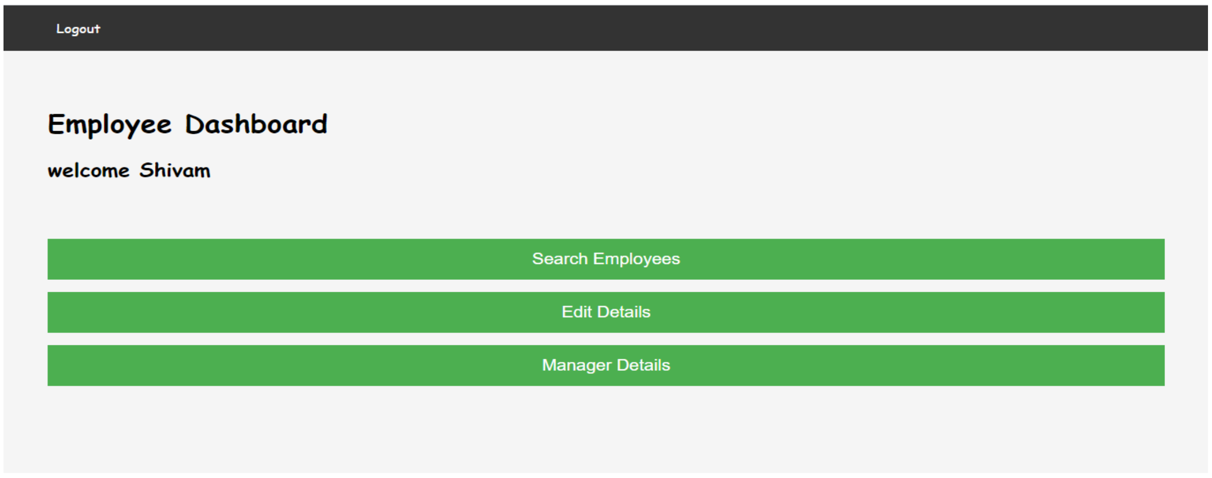


Figure 1.3 (Employee Dashboard)

If the user clicks on search employee, he will be redirected to the page shown in figure 1.3 below there are three available options to search employee :
First one is search the employee by name in this option the user will enters the name of the employee and as the button searched by name is pressed the list is displayed to the user with that particular name which he entered while searching. In similar ways we have further two options these are to search employees by their project and to search employees by their designation. Further details are shown in figure 1.4 below.

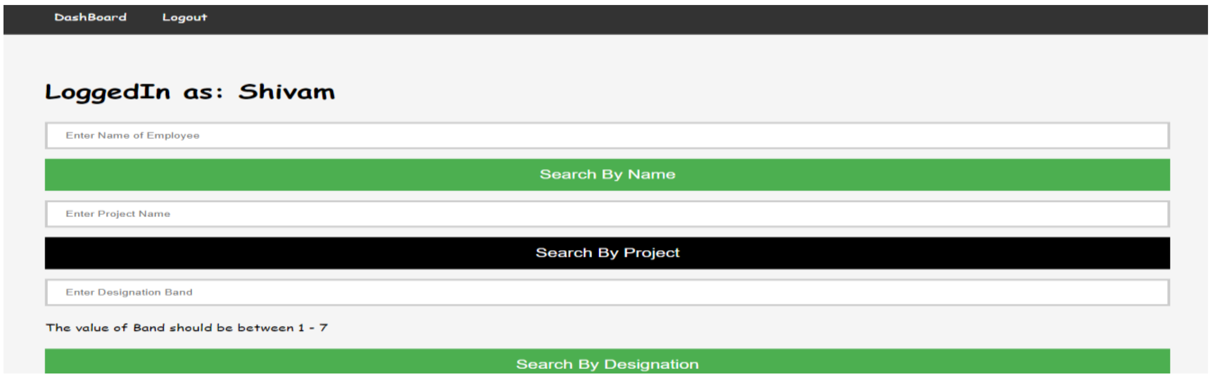


Figure 1.4 (Search employees)

As the second available options in figure 1.3 is to edit the details of the employee if the user clicks on that he will get five options by which the details of employee can be modified as shown in figure 1.5 and 1.6

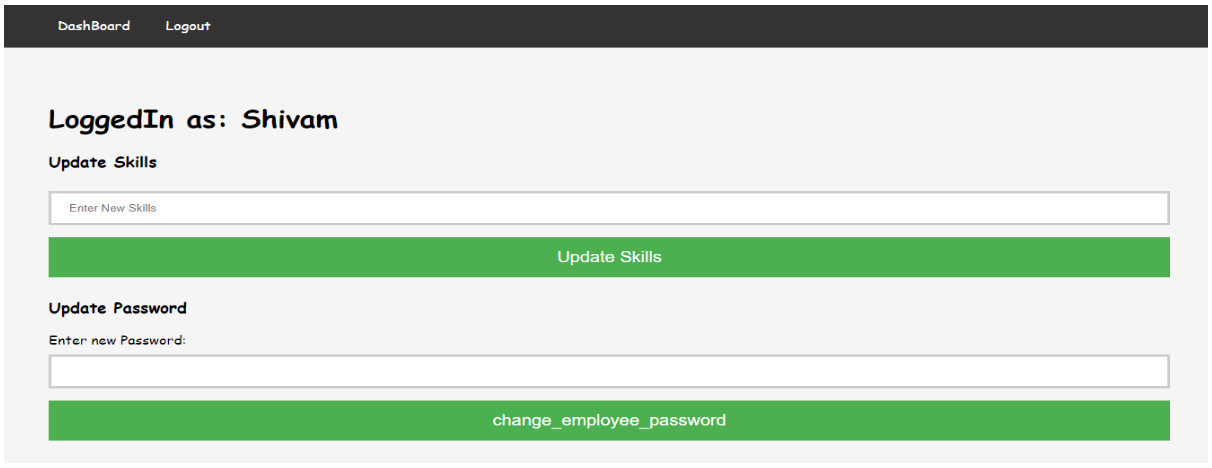


Figure 1.5 (Update details)

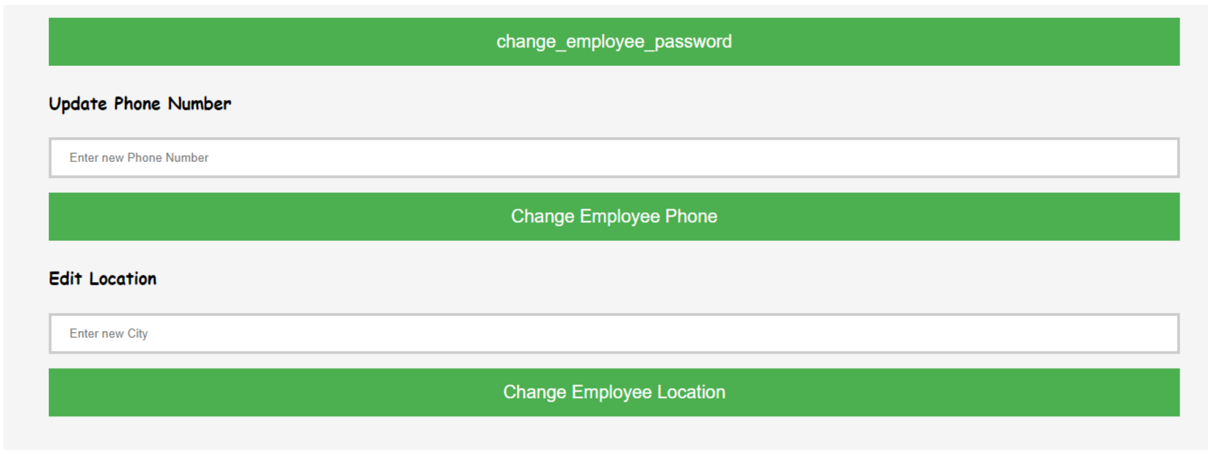
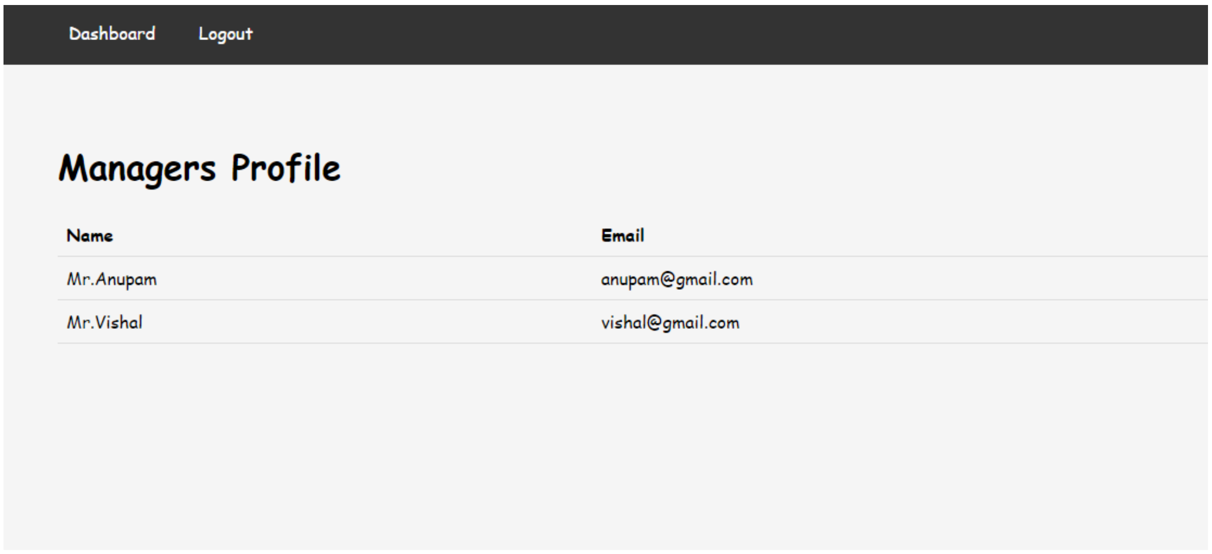


Figure 1.6 (Update details)

In employee dashboard the third option is manager details shown in figure 1.2 if user clicks on that option further the list of manager is provided which contains the managers details as shown in figure 1.7.



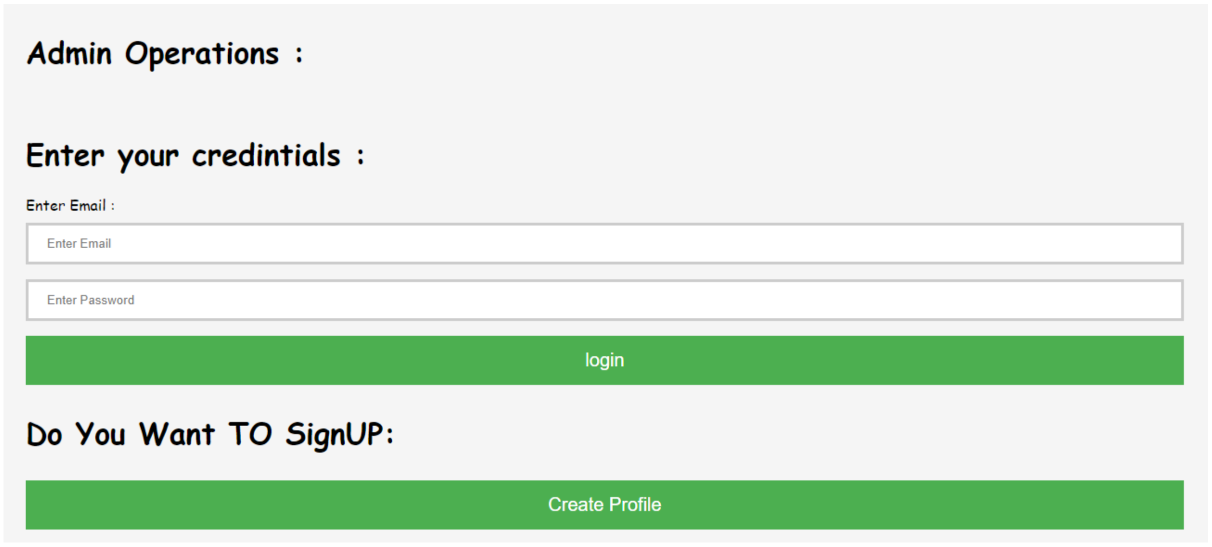
Managers Profile

Name	Email
Mr.Anupam	anupam@gmail.com
Mr.Vishal	vishal@gmail.com

Figure 1.7 (Managers list)

If the user clicks on the logout button located in the navigation bar the session will be destroyed he will be logged out form the portal.

In figure 1.1 if the user selects the admin options them he will be redirected to admin portal in the admin portal either a user login and if he is new to the application he needs to create his profile as shown in figure 1.8.



Admin Operations :

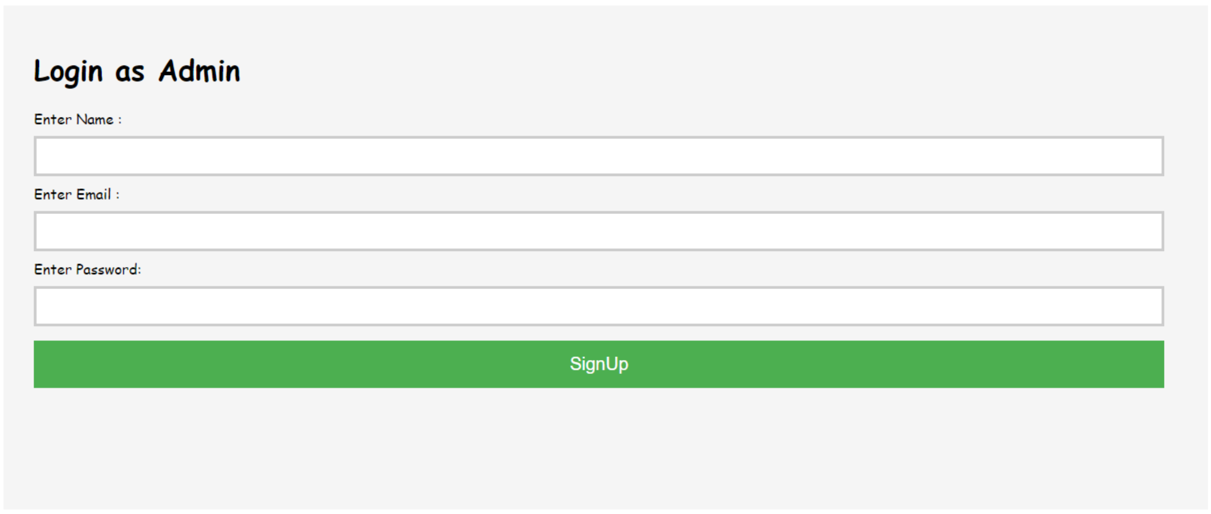
Enter your credintials :

Enter Email :

Do You Want TO SignUP:

Figure 1.8 (Admin portal)

If user clicks on sign up button he will be redirected to registration page and can create his account as shown below in figure 1.9.



Login as Admin

Enter Name :

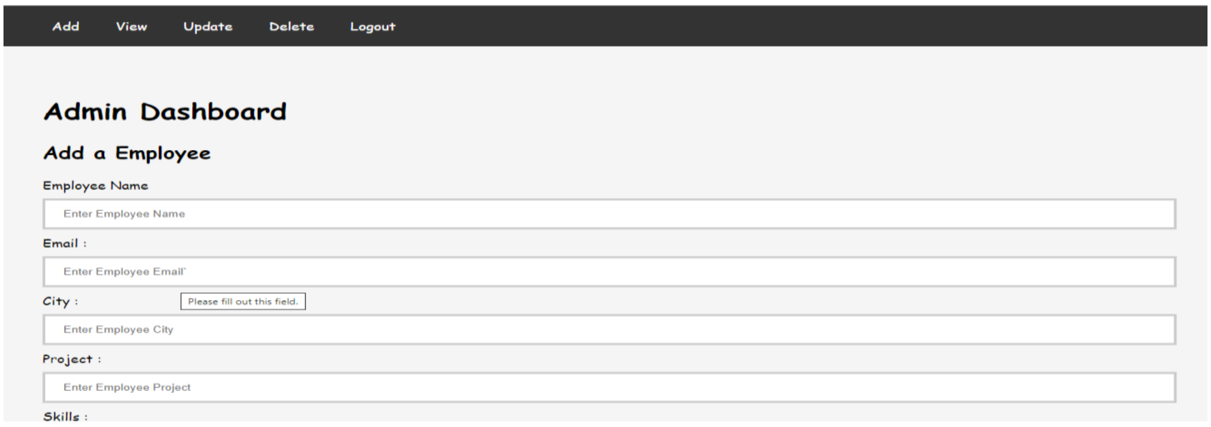
Enter Email :

Enter Password:

SignUp

Figure 1.9 (Register as admin)

after the successful login as an admin there are various functions of admin like to add a new employee, to view existing employees, to update the details of the existing employee as shown below in figure 1.9 and 1.10.



Add View Update Delete Logout

Admin Dashboard

Add a Employee

Employee Name

Enter Employee Name

Email :

Enter Employee Email

City : Please fill out this field.

Enter Employee City

Project :

Enter Employee Project

Skills :

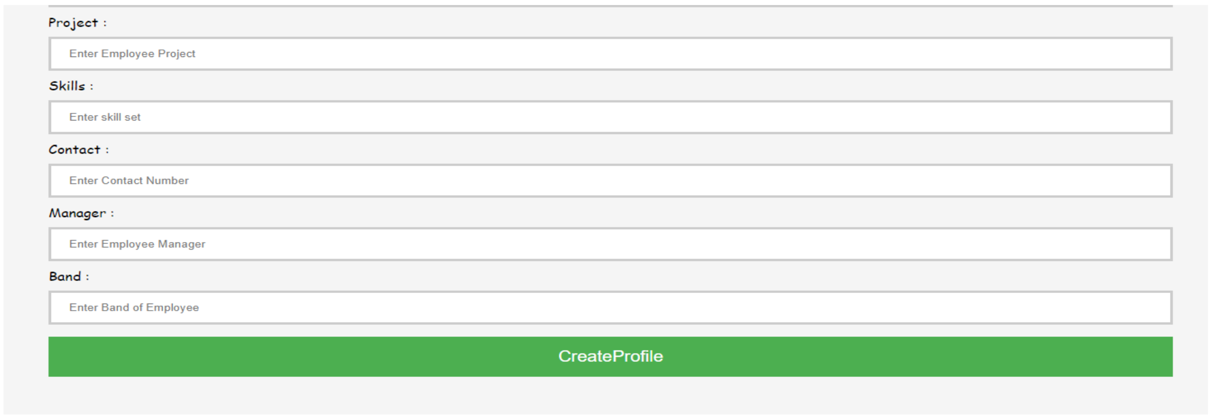
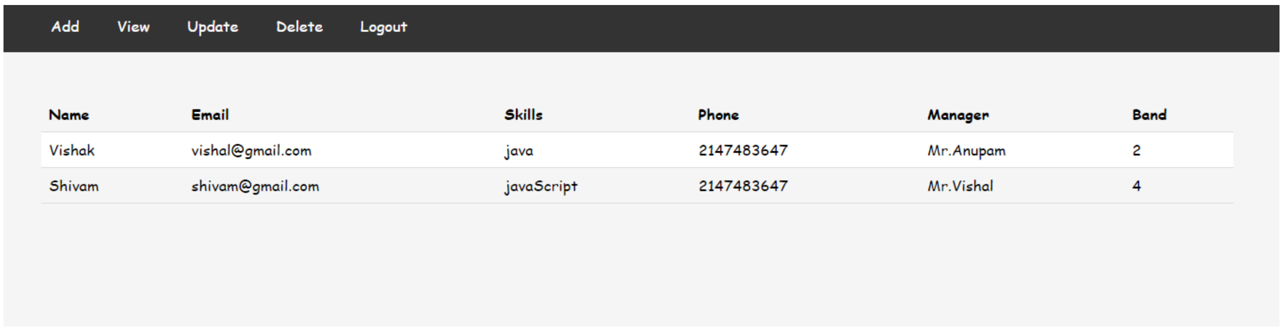


Figure 1.10 (Admin dashboard)

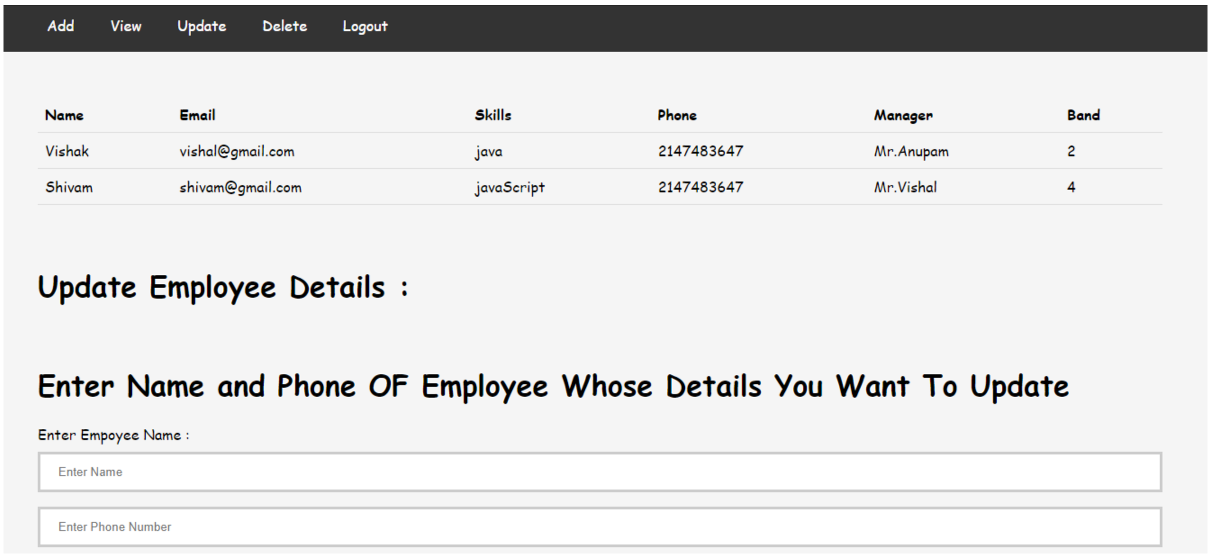
By clicking on view button in Admin dashboard navigation bar admin can view the list of all registered employees as shown in figure 1.10 below.



Name	Email	Skills	Phone	Manager	Band
Vishak	vishal@gmail.com	java	2147483647	Mr.Anupam	2
Shivam	shivam@gmail.com	javaScript	2147483647	Mr.Vishal	4

Figure 1.11 (Employee table)

By clicking on update button located in navigation bar the admin can update the details of employees and redirected to update page as shown in figure 1.11.



Name	Email	Skills	Phone	Manager	Band
Vishak	vishal@gmail.com	java	2147483647	Mr.Anupam	2
Shivam	shivam@gmail.com	javaScript	2147483647	Mr.Vishal	4

Update Employee Details :

Enter Name and Phone OF Employee Whose Details You Want To Update

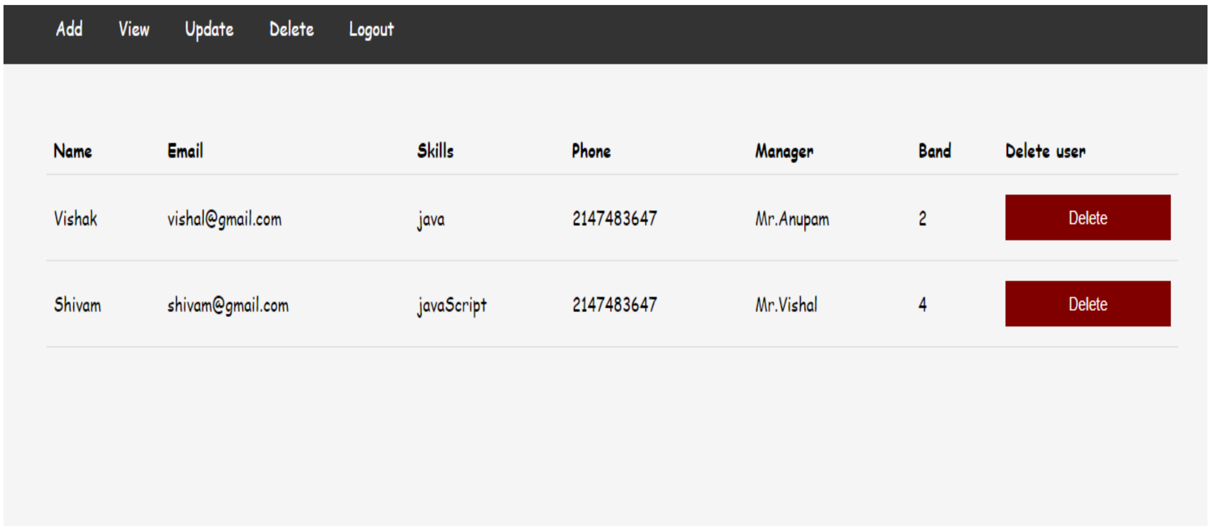
Enter Empoyee Name :

Enter Name

Enter Phone Number

Figure 1.12 (Update employee)

If admin clicks on delete button located in the navigation bar he will redirected to delete employee page as shown in figure 1.12.



Name	Email	Skills	Phone	Manager	Band	Delete user
Vishak	vishal@gmail.com	java	2147483647	Mr.Anupam	2	Delete
Shivam	shivam@gmail.com	javaScript	2147483647	Mr.Vishal	4	Delete

Figure 1.13 (Delete employee)

Further if admin clicks on logout the session will get destroyed and he will be logged out from the application.

1.2SCOPE OF THE PROJECT

Still in lots of places the traditional ways are used to manage the details of employees working in organization it needs lots of mental labor and time. So by considering this problem statement I initiated the work of this project to make these type of management tasks super easy.

Major objectives of this project are:

- To implement the power of programming in making the management of employee details easy and convenient.
- To develop a scalable web application which is easy to use.
- To use the power of database in storing the employee details in a secured and structured manner.

As described in project introduction project is very close to achieve all the listed objectives and I'm also working on the accessibility and in future I'm focused to convert this web application into a single page web application.

This project is developed by taking the reference of water fall modal of software engineering first requirement gathered and a rigorous tested rigorously after coding and implementation. The user interface of project is very user friendly so a new person can easily understand and use this project.

This web application is very helpful for the people who wants to save their time and labor by using the power of computer to manage the tasks such as to keeping, updating and deleting the records of various employees working in the organization.

Further if we focus in this project then the:

- Complete elimination of paperwork in employee management in this way office collaborator will contribute to ease of doing work.
- Employees will have access to their personal profiles and will be able to edit their details.
- The admin will add an employee and a default password and employee id will be generated and sent to the new employees.

1.3 Software used

visual studio code:

Figure 1.14 (vs code)

Visual Studio Code, also commonly referred to as VS Code is a source-code editor made by Microsoft for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality.

In the Stack Overflow 2021 Developer Survey, Visual Studio Code was ranked the most popular developer environment tool, with 70% of 82,000 respondents reporting that they use it

Visual Studio Code is a source-code editor that can be used with a variety of programming languages, including Java, JavaScript, Go, Node.js, Python, C++ and Fortran. It is based on the Electron framework, which is used to develop Node.js Web applications that run on the Blink layout engine. Visual Studio Code employs the same editor component (codenamed "Monaco") used in Azure DevOps (formerly called Visual Studio Online and Visual Studio Team Services).

Out of the box, Visual Studio Code includes basic support for most common programming languages. This basic support includes syntax highlighting, bracket matching, code folding, and configurable snippets. Visual Studio Code also ships with IntelliSense for JavaScript, TypeScript, JSON, CSS, and HTML, as well as debugging support for Node.js. Support for additional languages can be provided by freely available extensions on the VS Code Marketplace.

An orange version of the Visual Studio Code logo for the insiders version of Visual Studio Code
Visual Studio Code Insiders logo
Instead of a project system, it allows users to open one or more directories, which can then be saved in workspaces for future reuse. This allows it to operate as a language-agnostic code editor for any language. It supports many programming languages and a set of features that differs per language. Unwanted files and folders can be excluded from the project tree via the settings. Many Visual Studio Code features are not exposed through menus or the user interface but can be accessed via the command palette.

Visual Studio Code can be extended via extensions, available through a central repository. This includes additions to the editor and language support. A notable feature is the ability to create extensions that add support for new languages, themes, and debuggers, perform static code analysis, and add code linters using the Language Server Protocol.

Source control is a built-in feature of Visual Studio Code. It has a dedicated tab inside of the menu bar where you can access version control settings and view changes made to the current project. To use the feature you must link Visual Studio Code to any supported version control system (Git, Apache Subversion, Perforce, etc.). This allows you to create repositories as well as to make push and pull requests directly from the Visual Studio Code program.

Visual Studio Code includes multiple extensions for FTP, allowing the software to be used as a free alternative for web development. Code can be synced between the editor and the server, without downloading any extra software.

Visual Studio Code allows users to set the code page in which the active document is saved, the newline character, and the programming language of the active document. This allows it to be used on any platform, in any locale, and for any given programming language.

MySQL :

Figure 1.15 (MySQL)

MySQL is an open-source relational database management system (RDBMS). Its name is a combination of "My", the name of co-founder Michael Widenius's daughter, and "SQL", the abbreviation for Structured Query Language. A relational database organizes data into one or more data tables in which data may be related to each other; these relations help structure the data. SQL is a language programmers use to create, modify and extract data from the relational database, as well as control user access to the database. In addition to relational databases and SQL, an RDBMS like MySQL works with an operating system to implement a relational database in a computer's storage system, manages users, allows for network access and facilitates testing database integrity and creation of backups.

MySQL is free and open-source software under the terms of the GNU General Public License, and is also available under a variety of proprietary licenses. MySQL was owned and sponsored by the Swedish company MySQL AB, which was bought by Sun Microsystems (now Oracle Corporation). In 2010, when Oracle acquired Sun, Widenius forked the open-source MySQL project to create MariaDB.

MySQL has stand-alone clients that allow users to interact directly with a MySQL database using SQL, but more often, MySQL is used with other programs to implement applications that need relational database capability. MySQL is a component of the LAMP web application software stack (and others), which is an acronym for Linux, Apache, MySQL, Perl/PHP/Python. MySQL is used by many database-driven web applications, including Drupal, Joomla, phpBB, and WordPress.

MySQL is offered under two different editions: the open source MySQL Community Server[77] and the proprietary Enterprise Server.[78] MySQL Enterprise Server is differentiated by a series of proprietary extensions which install as server plugins, but otherwise shares the version numbering system and is built from the same code base.

Major features as available in MySQL

A broad subset of ANSI SQL 99, as well as extensions
Cross-platform support
Stored procedures, using a procedural language that closely adheres to SQL/PSM
Triggers
Cursors
Updatable views
Online Data Definition Language (DDL) when using the InnoDB Storage Engine.
Information schema
Performance Schema that collects and aggregates statistics about server execution and query performance for monitoring purposes.
A set of SQL Mode options to control runtime behavior, including a strict mode to better adhere to SQL standards.
X/Open XA distributed transaction processing (DTP) support; two phase commit as part of this, using the default InnoDB storage engine
Transactions with savepoints when using the default InnoDB Storage Engine. The NDB Cluster Storage Engine also supports transactions.
ACID compliance when using InnoDB and NDB Cluster Storage Engines
SSL support
Query caching
Sub-SELECTs (i.e. nested SELECTs)
Built-in replication support
Asynchronous replication: master-slave from one master to many slaves or many masters to one slave

Semi synchronous replication: Master to slave replication where the master waits on replication

Synchronous replication: Multi-master replication is provided in MySQL Cluster.

Virtual Synchronous: Self managed groups of MySQL servers with multi master support can be done using: Galera Cluster or the built in Group Replication plugin

Full-text indexing and searching

Embedded database library

Unicode support

Partitioned tables with pruning of partitions in optimizer

Shared-nothing clustering through MySQL Cluster

Multiple storage engines, allowing one to choose the one that is most effective for each table in the application.

Native storage engines InnoDB, MyISAM, Merge, Memory (heap), Federated, Archive, CSV, Blackhole, NDB Cluster.

Commit grouping, gathering multiple transactions from multiple connections together to increase the number of commits per second.

Though MySQL began as a low-end alternative to more powerful proprietary databases, it has gradually evolved to support higher-scale needs as well. It is still most commonly used in small to medium scale single-server deployments, either as a component in a LAMP-based web application or as a standalone database server. Much of MySQL's appeal originates in its relative simplicity and ease of use, which is enabled by an ecosystem of open source tools such as phpMyAdmin. In the medium range, MySQL can be scaled by deploying it on more powerful hardware, such as a multi-processor server with gigabytes of memory.

Deployment- MySQL can be built and installed manually from source code, but it is more commonly installed from a binary package unless special customizations are required. On most Linux distributions, the package management system can download and install MySQL with minimal effort, though further configuration is often required to adjust security and optimization settings.

There are, however, limits to how far performance can scale on a single server ('scaling up'), so on larger scales, multi-server MySQL ('scaling out') deployments are required to provide improved performance and reliability. A typical high-end configuration can include a powerful master database which handles data write operations and is replicated to multiple slaves that handle all read operations. The master server continually pushes binlog events to connected slaves so in the event of failure a slave can be promoted to become the new master, minimizing downtime. Further improvements in performance can be achieved by caching the results from database queries in memory using memcached, or breaking down a database into smaller chunks called shards which can be spread across a number of distributed server clusters.

Google chrome

Figure 1.16 (Google chrome)

Google Chrome is a cross-platform web browser developed by Google. It was first released in 2008 for Microsoft Windows, built with free software components from Apple WebKit and Mozilla Firefox. It was later ported to Linux, macOS, iOS, and Android, where it is the default browser. The browser is also the main component of Chrome OS, where it serves as the platform for web applications.

Most of Chrome's source code comes from Google's free and open-source software project Chromium, but Chrome is licensed as proprietary freeware. WebKit was the original rendering engine, but Google eventually forked it to create the Blink engine; all Chrome variants except iOS now use Blink.

Chrome was assembled from 25 different code libraries from Google and third parties such as Mozilla's Netscape Portable Runtime, Network Security Services, NPAPI (dropped as of version 45), Skia Graphics Engine, SQLite, and a number of other open-source projects. The V8 JavaScript virtual machine was considered a sufficiently important project to be split off (as was Adobe/Mozilla's Tamarin) and handled by a separate team in Denmark coordinated by Lars Bak in Aarhus. According to Google, existing implementations were designed "for small programs, where the performance and interactivity of the system weren't that important", but web applications such as Gmail "are using the web browser to the fullest when it comes to DOM manipulations and JavaScript", and therefore would significantly benefit from a JavaScript engine that could work faster.

Chrome initially used the WebKit rendering engine to display web pages. In 2013, they forked the WebCore component to create their own layout engine Blink. Based on WebKit, Blink only uses WebKit's "WebCore" components, while substituting other components, such as its own multi-process architecture, in place of WebKit's native implementation.[16] Chrome is internally tested with unit testing, automated testing of scripted user actions, fuzz testing, as well as WebKit's layout tests (99% of which Chrome is claimed to have passed), and against commonly accessed websites inside the Google index within 20–30 minutes.[29] Google created Gears for Chrome, which added features for web developers typically relating to the building of web applications, including offline support.[29] Google phased out Gears as the same functionality became available in the HTML5 standards.

In March 2011, Google introduced a new simplified logo to replace the previous 3D logo that had been used since the project's inception. Google designer Steve Rura explained the company reasoning for the change: "Since Chrome is all about making your web experience as easy and clutter-free as possible, we refreshed the Chrome icon to better represent these sentiments. A simpler icon embodies the Chrome spirit – to make the web quicker, lighter, and easier for all."

On January 11, 2011, the Chrome product manager, Mike Jazayeri, announced that Chrome would remove H.264 video codec support for its HTML5 player, citing the desire to bring Google Chrome more in line with the currently available open codecs available in the Chromium project, which Chrome is based on. Despite this, on November 6, 2012, Google released a version of Chrome on Windows which added hardware-accelerated H.264 video decoding. In October 2013, Cisco announced that it was open-sourcing its H.264 codecs and would cover all fees required.

On February 7, 2012, Google launched Google Chrome Beta for Android 4.0 devices. On many new devices with Android 4.1 and later preinstalled, Chrome is the default browser. In May 2017, Google announced a version of Chrome for augmented reality and virtual reality devices.

1.4 Desired Hardware :

RAM	-	Minimum 4 GB RAM
Processor	-	Minimum Intel i3
Operating System	–	Window 10

1.5 SDLC

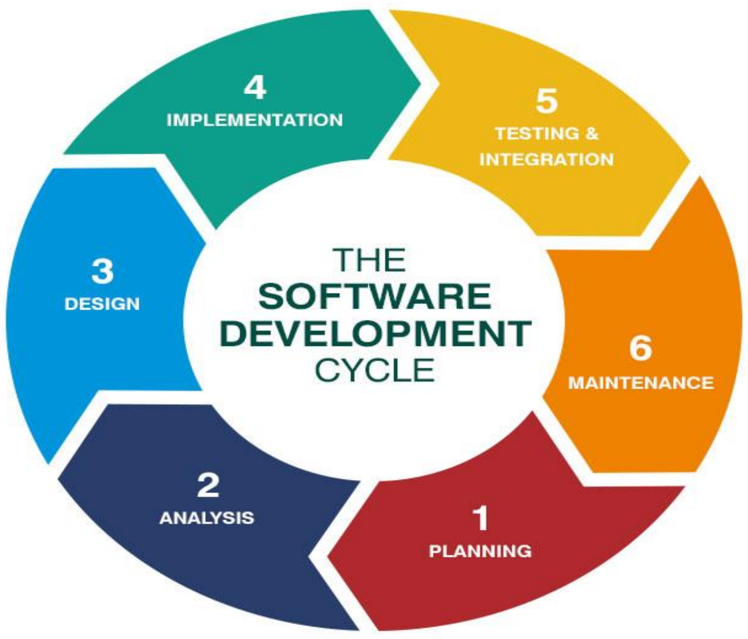


Figure 1.17 (SDLC)

SDLC or the Software Development Life Cycle is a process that produces software with the highest quality and lowest cost in the shortest time possible. SDLC provides a well-structured flow of phases that help an organization to quickly produce high-quality software which is well-tested and ready for production use.

This project is developed on the basis of water fall modal of SDLC.

The Waterfall Model was the first Process Model to be introduced. It is also referred to as a linear-sequential life cycle model. It is very simple to understand and use. In a waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases.

The Waterfall model is the earliest SDLC approach that was used for software development.

The waterfall Model illustrates the software development process in a linear sequential flow. This means that any phase in the development process begins only if the previous phase is complete. In this waterfall model, the phases do not overlap.

Design

Waterfall approach was first SDLC Model to be used widely in Software Engineering to ensure success of the project. In "The Waterfall" approach, the whole process of software development is divided into separate phases. In this Waterfall model, typically, the outcome of one phase acts as the input for the next phase sequentially.

The following illustration is a representation of the different phases of the Waterfall Model.

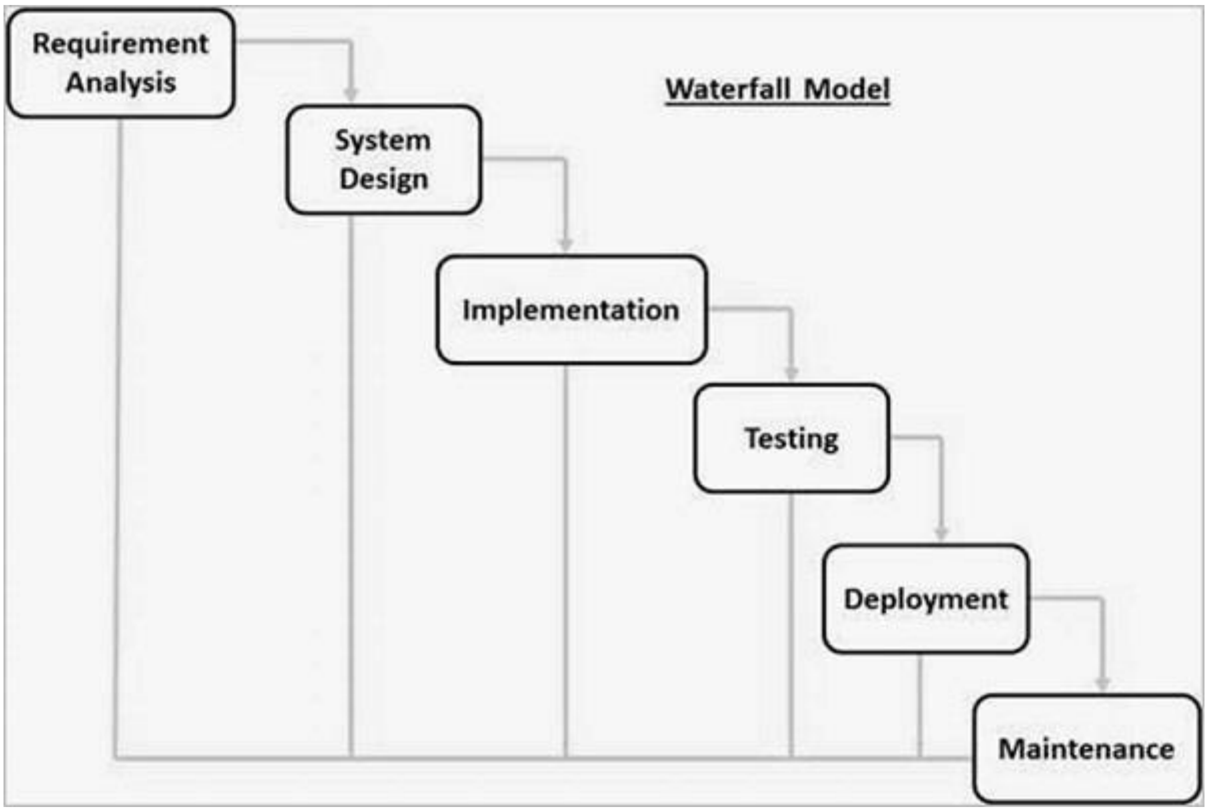


Figure 1.18 (Water fall model)

sequential phases in Waterfall model are −

Requirement Gathering and analysis − All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.

System Design − The requirement specifications from first phase are studied in this phase and the system design is prepared. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.

Implementation − With inputs from the system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.

Integration and Testing − All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.

Deployment of system − Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.

Maintenance − There are some issues which come up in the client environment. To fix those issues, patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

All these phases are cascaded to each other in which progress is seen as flowing steadily downwards (like a waterfall) through the phases. The next phase is started only after the defined set of goals are achieved for previous phase and it is signed off, so the name "Waterfall Model". In this model, phases do not overlap.

Application of the model

Every software developed is different and requires a suitable SDLC approach to be followed based on the internal and external factors. Some situations where the use of Waterfall model is most appropriate are −

- Requirements are very well documented, clear and fixed.

- Product definition is stable.
- Technology is understood and is not dynamic.
- There are no ambiguous requirements.
- Ample resources with required expertise are available to support the product.

CHAPTER 2
LITERATURE REVIEW

2.1 MYSQL

MySQL is an open-source relational database management system (RDBMS). Its name is a combination of "My", the name of co-founder Michael Widenius's daughter, and "SQL", the abbreviation for Structured Query Language. A relational database organizes data into one or more data tables in which data may be related to each other; these relations help structure the data. SQL is a language programmers use to create, modify and extract data from the relational database, as well as control user access to the database. In addition to relational databases and SQL, an RDBMS like MySQL works with an operating system to implement a relational database in a computer's storage system, manages users, allows for network access and facilitates testing database integrity and creation of backups.

MySQL is free and open-source software under the terms of the GNU General Public License, and is also available under a variety of proprietary licenses. MySQL was owned and sponsored by the Swedish company MySQL AB, which was bought by Sun Microsystems (now Oracle Corporation). In 2010, when Oracle acquired Sun, Widenius forked the open-source MySQL project to create MariaDB.

MySQL has stand-alone clients that allow users to interact directly with a MySQL database using SQL, but more often, MySQL is used with other programs to implement applications that need relational database capability. MySQL is a component of the LAMP web application software stack (and others), which is an acronym for Linux, Apache, MySQL, Perl/PHP/Python. MySQL is used by many database-driven web applications, including Drupal, Joomla, phpBB, and WordPress.

2.2 Java script (JS & JS OPTIMISER)

JavaScript (JS), is a programming language that is one of the core technologies of the World Wide Web, alongside HTML and CSS. Over 97% of websites use JavaScript on the client side for web page behavior, often incorporating third-party libraries.All major web browsers have a dedicated JavaScript engine to execute the code on users' devices.

JavaScript is a high-level, often just-in-time compiled language that conforms to the ECMAScript standard. It has dynamic typing, prototype-based object-orientation, and first-class functions. It is multi-paradigm, supporting event-driven, functional, and imperative programming styles. It has application programming interfaces (APIs) for working with text, dates, regular expressions, standard data structures, and the Document Object Model (DOM).

The ECMAScript standard does not include any input/output (I/O), such as networking, storage, or graphics facilities. In practice, the web browser or other runtime system provides JavaScript APIs for I/O.

JavaScript engines were originally used only in web browsers, but are now core components of some servers and a variety of applications. The most popular runtime system for this usage is Node.js.

Although Java and JavaScript are similar in name, syntax, and respective standard libraries, the two languages are distinct and differ greatly in design.

JavaScript has become a popular programming language. It is widely used in both client-side and server-side programming in web applications. The robustness and performance of JavaScript programs become vital. Unfortunately, real-world JavaScript programs often suffer from various issues. In this work, we present nine issue patterns derived from open-source projects and propose a general static analysis framework, JSOptimizer, to help detect such patterns of issues and optimize the code accordingly. Comparing to existing work, JSOptimizer is not only highly extensible but also performs code optimizations automatically. We applied JS Optimizer to seven real open-source JavaScript projects and five bugs detected by it have been confirmed by developers. Besides, we conducted a case study based on a popular project and found that addressing the issues detected by our framework can speed up the original project by over 300%. This shows the usefulness of JS Optimizer.

2.3 Modern JavaScript framework

With the increasing popularity of the web, some new web technologies emerged and introduced dynamics to web applications, in comparison to HTML, as a static programming language. JavaScript is the language that provided a dynamic web site which actively communicates with users. JavaScript is used in today's web applications as a client script language and on the server side. The JavaScript language supports the Model View Controller (MVC) architecture that maintains a readable code and clearly separates parts of the program code. The topic of this research is to compare the popular JavaScript frameworks: AngularJS, Ember, Knockout, Backbone. All four frameworks are based on MVC or similar architecture. In this paper, the advantages and disadvantages of each framework, the impact on application speed, the ways of testing such JS applications and ways to improve code security are presented.

2.4 Integration of HTML pages in Web Pages

The growing number of Web pages on the Internet introduces a need to combine and integrate information from HTML tables of different Web pages that contain similar information into a single Web page, especially information from the same domain of interest. This paper presents an approach of HTML table integration by combining several existing methods that are proved to solve different issues in the integration processes. The integration of HTML table consists of three phases: (1) extraction of the structure of the tables; (2) integration of the tables' schema; (3) integration of the data values. To solve the conflicts in semantics and naming in the tables schema, domain-ontology is used.

CHAPTER 3
FEASIBILITY STUDY

As the name implies, a feasibility analysis is used to determine the viability of an idea, such as ensuring a project is legally and technically feasible as well as economically justifiable. It tells us whether a project is worth the investment—in some cases, a project may not be doable. There can be many reasons for this, including requiring too many resources, which not only prevents those resources from performing other tasks but also may cost more than an organization would earn back by taking on a project that isn't profitable.

A well-designed study should offer a historical background of the business or project, such as a description of the product or service, accounting statements, details of operations and management, marketing research and policies, financial data, legal requirements, and tax obligations. Generally, such studies precede technical development and project implementation.

What is Feasibility Study?

As the name implies, a feasibility analysis is used to determine the viability of an idea, such as ensuring a project is legally and technically feasible as well as economically justifiable. It tells us whether a project is worth the investment—in some cases, a project may not be doable. There can be many reasons for this, including requiring too many resources, which not only prevents those resources from performing other tasks but also may cost more than an organization would earn back by taking on a project that isn't profitable.

A well-designed study should offer a historical background of the business or project, such as a description of the product or service, accounting statements, details of operations and management, marketing research and policies, financial data, legal requirements, and tax obligations. Generally, such studies precede technical development and project implementation.

Technical feasibility assesses the current resources (such as hardware and software) and technology, which are required to accomplish user requirements in the software within the allocated time and budget. For this, the software development team ascertains whether the current resources and technology can be upgraded or added in the software to accomplish specified user requirements. Technical feasibility also performs the following tasks.

• Analyzes the technical skills and capabilities of the software development team members.

• Determines whether the relevant technology is stable and established.

• Ascertains that the technology chosen for software development has a large number of users so that they can be consulted when problems arise or improvements are required.

Operational feasibility assesses the extent to which the required software performs a series of steps to solve business problems and user requirements. This feasibility is dependent on human resources (software development team) and involves visualizing whether the software will operate after it is developed and be operative once it is installed. Operational feasibility also performs the following tasks.

• Determines whether the problems anticipated in user requirements are of high priority.

• Determines whether the solution suggested by the software development team is acceptable.

• Analyzes whether users will adapt to a new software.

• Determines whether the organization is satisfied by the alternative solutions proposed by the software development team.

Economic feasibility determines whether the required software is capable of generating financial gains for an organization. It involves the cost incurred on the software development team, estimated cost of hardware and software, cost of performing feasibility study, and so on. For this, it is essential to consider expenses made on purchases (such as hardware purchase) and activities required to carry out software development. In addition, it is necessary to consider the benefits that can be achieved by developing the software. Software is said to be economically feasible if it focuses on the issues listed below.

• Cost incurred on software development to produce long-term gains for an organization.

• Cost required to conduct full software investigation (such as requirements elicitation and requirements analysis).

• Cost of hardware, software, development team, and training.

Feasibility Study Process

Feasibility study comprises the following steps.

• Information assessment: Identifies information about whether the system helps in achieving the objectives of the organization. It also verifies that the system can be implemented using new technology and within the budget and whether the system can be integrated with the existing system.

• Information collection: Specifies the sources from where information about software can be obtained. Generally, these sources include users (who will operate the software), organization (where the software will be used), and the software development team (which understands user requirements and knows how to fulfill them in software).

• Report writing: Uses a feasibility report, which is the conclusion of the feasibility study by the software development team. It includes the recommendations whether the software development should continue. This report may also include information about changes in the software scope, budget, and schedule and suggestions of any requirements in the system.

• General information: Describes the purpose and scope of feasibility study. It also describes system overview, project references, acronyms and abbreviations, and points of contact to be used. System overview provides description about the name of the organization responsible for the software development, system name or title, system category, operational status, and so on. Project references provide a list of the references used to prepare this document such as documents relating to the project or previously developed documents that are related to the project. Acronyms and abbreviations provide a list of the terms that are used in this document along with their meanings. Points of contact provide a list of points of organizational contact with users for information and coordination. For example, users require assistance to solve problems (such as troubleshooting) and collect information such as contact number, e-mail address, and so on.

Management summary: Provides the following information.

• Environment: Identifies the individuals responsible for software development. It provides information about input and output requirements, processing requirements of the software and the interaction of the software with other software. It also identifies system security requirements and the system's processing requirements

• Current functional procedures: Describes the current functional procedures of the existing system, whether automated or manual. It also includes the data-flow of the current system and the number of team members required to operate and maintain the software.

• Functional objective: Provides information about functions of the system such as new services, increased capacity, and so on.

• Performance objective: Provides information about performance objectives such as reduced staff and equipment costs, increased processing speeds of software, and improved controls.

• Assumptions and constraints: Provides information about assumptions and constraints such as operational life of the proposed software, financial constraints, changing hardware, software and operating environment, and availability of information and sources.

• Methodology: Describes the methods that are applied to evaluate the proposed software in order to reach a feasible alternative. These methods include survey, modeling, benchmarking, etc.

• Evaluation criteria: Identifies criteria such as cost, priority, development time, and ease of system use, which are applicable for the development process to determine the most suitable system option.

• Recommendation: Describes a recommendation for the proposed system. This includes the delays and acceptable risks.

• Proposed software: Describes the overall concept of the system as well as the procedure to be used to meet user requirements. In addition, it provides information about improvements, time and resource costs, and impacts. Improvements are performed to enhance the functionality and performance of the existing software. Time and resource costs include the costs associated with software development from its requirements to its maintenance and staff training. Impacts describe the possibility of future happenings and include various types of impacts as listed below.

• Equipment impacts: Determine new equipment requirements and changes to be made in the currently available equipment requirements.

• Software impacts: Specify any additions or modifications required in the existing software and supporting software to adapt to the proposed software.

• Organizational impacts: Describe any changes in organization, staff and skills requirement.

• Operational impacts: Describe effects on operations such as user-operating procedures, data processing, data entry procedures, and so on.

• Developmental impacts: Specify developmental impacts such as resources required to develop databases, resources required to develop and test the software, and specific activities to be performed by users during software development.

• Security impacts: Describe security factors that may influence the development, design, and continued operation of the proposed software.

• Alternative systems: Provide description of alternative systems, which are considered in a feasibility study. This also describes the reasons for choosing a particular alternative system to develop the proposed software and the reason for rejecting alternative systems.

Types of Feasibility Study

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.

3.1 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.

Best practices for conducting a technical feasibility study

Use the available tools and templates to help you collate and gather accurate information.

Gather feedback and suggestions from all stakeholders, including clients, product designers, developers and other team members.

Ask technical questions to the core team members to investigate and get reliable data.

If possible, outsource the market survey to a market research team with experience and expertise in the field.

Break the study into different parts and evaluate the information you collect separately in each stage.

Collate the feedback from each stage and develop the final review without any bias.

3.2 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.

Economic feasibility is the second part of resource determination. The basic resources to consider are your time and that of the systems analysis team, the cost of doing a full systems study (including the time of employees you will be working with), the cost of the business employee time, the estimated cost of hardware, and the estimated cost of software or software development.

The concerned business must be able to see the value of the investment it is pondering before committing to an entire systems study. If short-term costs are not overshadowed by long-term gains or produce no immediate reduction in operating costs, the system is not economically feasible and the project should not proceed any further.

In Economic Feasibility study cost and benefit of the project is analyzed. Means under this feasibility study a detail analysis is carried out what will be cost of the project for development which includes all required cost for final development like hardware and software resource required, design and development cost and operational cost and so on. After that it is analyzed whether project will be beneficial in terms of finance for organization or not.

Economic feasibility studies are an evaluation of effectiveness of candidate systems by combining cost analysis with benefits analysis. Benefits and costs of a candidate system are compared to find out the net benefits of an organization's system.

2.3 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.

In Legal Feasibility study project is analyzed in legality point of view. This includes analyzing barriers of legal implementation of project, data protection acts or social media laws, project certificate, license, copyright etc. Overall it can be said that Legal Feasibility Study is study to know if proposed project conform legal and ethical requirements.

legitimateness perspective of the project is analyzed. This incorporates breaking down boundaries to the legitimate usage of the project, information protection acts or web-based media laws, project declaration, permit, copyright, etc. In general, it very well may be said that this study is an analysis to know whether the proposed project adjusts to legal and moral necessities.

3.3. 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.

Operational feasibility is the measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development.

The operational feasibility assessment focuses on the degree to which the proposed development project fits in with the existing business environment and objectives with regard to development schedule, delivery date, corporate culture and existing business processes.

To ensure success, desired operational outcomes must be imparted during design and development. These include such design-dependent parameters as reliability, maintainability, supportability, usability, producibility, disposability, sustainability, affordability and others. These parameters are required to be considered at the early stages of design if desired operational behaviours are to be realised. A system design and development requires appropriate and timely application of engineering and management efforts to meet the previously mentioned parameters. A system may serve its intended purpose most effectively when its technical and operating characteristics are engineered into the design.

In Operational Feasibility degree of providing service to requirements is analyzed along with how much easy product will be to operate and maintenance after deployment. Along with this other operational scopes are determining usability of product, Determining suggested solution by software development team is acceptable or not etc.

3.4. 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, etc.

The process of assessing the degree to which the potential time frame and completion dates for all major activities within a project meet organizational deadlines and constraints for affecting change.

How long a project will run, analyzed by the schedule feasibility study?

A feasibility study not only collects the entire requirement and estimates the cost, but also calculate the total time for the completion of any project. Sometimes customer gives the timeline of the completion of the project, that time you have to set all the development according to the timelines. And then helps the schedule feasibility study. In this analysis we analyzed the project and estimates the time for the whole project.

This phase will help in many ways. For example, in the middle of the project you realize that the project is not moving in the speed you want, you can hire more people to increase the efforts and can complete the project in the given time. Also as the alternative, if your budget is not allowing you to hire more staffs you can retrain your existing ones in order to make your work in speed.

You can also do the assessment of the risks. Like whether the project is going to complete in the estimated timeline. And if not what are the problems going to come and what will the solutions of these problems. Analyst can mention all the possible risk and their solutions to avoid any kind of problem in the middle of the project. And these are to be done in schedule feasibility study.

Also in this assessment their must include some extension solutions like if project cant be completed in the estimated time then is there any possibility to extend the time. And if possible then what will be the extra cost coming and then also are we in the budget or not.

We can also include as an alternate for the timelines that we can avoid the holidays providing to the staffs. That means if project is about to finish and there is not much time left to complete, you can avoid the unwanted or not so much necessary holidays and can complete the project in the given time. For this the experts should have the good communications and understanding between the staffs and senior executives

Importance of Feasibility Study

The importance of a feasibility study is based on organizational desire to "get it right" before committing resources, time, or budget. A feasibility study might uncover new ideas that could completely change a project's scope. It's best to make these determinations in advance, rather than to jump in and to learn that the project won't work. Conducting a feasibility study is always beneficial to the project as it gives you and other stakeholders a clear picture of the proposed project.

Below are some key 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

Apart from the approaches to feasibility study listed above, some projects also require other constraints to be analyzed -

Internal Project Constraints: Technical, Technology, Budget, Resource, etc.

Internal Corporate Constraints: Financial, Marketing, Export, etc.

External Constraints: Logistics, Environment, Laws, and Regulations, etc.

3.5 Benefits of a Feasibility Study

Preparing a project's feasibility study is an important step that may assist project managers in making informed decisions about whether or not to spend time and money on the endeavor.

Feasibility studies may also help a company's management avoid taking on a tricky business endeavor by providing them with critical information.

An additional advantage of doing a feasibility study is that it aids in the creation of new ventures by providing information on factors such as how a company will work, what difficulties it could face, who its competitors are, and how much and where it will get its funding from. These marketing methods are the goal of feasibility studies, which try to persuade financiers and banks whether putting money into a certain company venture makes sense.

The results of your feasibility studies study are summarized in a feasibility report, which typically comprises the following sections.

CHAPTER 4
DESCRIPTION OF THE TECHNOLOGY USED

4.1 Java Script

JavaScript (JS), is a programming language that is one of the core technologies of the World Wide Web, alongside HTML and CSS. Over 97% of websites use JavaScript on the client side for web page behavior, often incorporating third-party libraries.All major web browsers have a dedicated JavaScript engine to execute the code on users' devices.

JavaScript is a high-level, often just-in-time compiled language that conforms to the ECMAScript standard. It has dynamic typing, prototype-based object-orientation, and first-class functions. It is multi-paradigm, supporting event-driven, functional, and imperative programming styles. It has application programming interfaces (APIs) for working with text, dates, regular expressions, standard data structures, and the Document Object Model (DOM).

The ECMAScript standard does not include any input/output (I/O), such as networking, storage, or graphics facilities. In practice, the web browser or other runtime system provides JavaScript APIs for I/O.

JavaScript engines were originally used only in web browsers, but are now core components of some servers and a variety of applications. The most popular runtime system for this usage is Node.js.

Although Java and JavaScript are similar in name, syntax, and respective standard libraries, the two languages are distinct and differ greatly in design.

History

Creation at Netscape

The first web browser with a graphical user interface, Mosaic, was released in 1993. Accessible to non-technical people, it played a prominent role in the rapid growth of the nascent World Wide Web. The lead developers of Mosaic then founded the Netscape corporation, which released a more polished browser, Netscape Navigator, in 1994. This quickly became the most-used.

During these formative years of the Web, web pages could only be static, lacking the capability for dynamic behavior after the page was loaded in the browser. There was a desire in the burgeoning web development scene to remove this limitation, so in 1995, Netscape decided to add a scripting language to Navigator. They pursued two routes to achieve this: collaborating with Sun Microsystems to embed the Java programming language, while also hiring Brendan Eich to embed the Scheme language.

Netscape management soon decided that the best option was for Eich to devise a new language, with syntax similar to Java and less like Scheme or other extant scripting languages. Although the new language and its interpreter implementation were called LiveScript when first shipped as part of a Navigator beta in September 1995, the name was changed to JavaScript for the official release in December.

The choice of the JavaScript name has caused confusion, implying that it is directly related to Java. At the time, the dot-com boom had begun and Java was the hot new language, so Eich considered the JavaScript name a marketing ploy by Netscape.

Adoption by Microsoft

Microsoft debuted Internet Explorer in 1995, leading to a browser war with Netscape. On the JavaScript front, Microsoft reverse-engineered the Navigator interpreter to create its own, called JScript.

JScript was first released in 1996, alongside initial support for CSS and extensions to HTML. Each of these implementations was noticeably different from their counterparts in Navigator. These differences made it difficult for developers to make their websites work well in both browsers, leading to widespread use of "best viewed in Netscape" and "best viewed in Int

The rise of JScript

In November 1996, Netscape submitted JavaScript to Ecma International, as the starting point for a standard specification that all browser vendors could conform to. This led to the official release of the first ECMAScript language specification in June 1997.

The standards process continued for a few years, with the release of ECMAScript 2 in June 1998 and ECMAScript 3 in December 1999. Work on ECMAScript 4 began in 2000.

Meanwhile, Microsoft gained an increasingly dominant position in the browser market. By the early 2000s, Internet Explorer's market share reached 95%. This meant that JScript became the de facto standard for client-side scripting on the Web.

Microsoft initially participated in the standards process and implemented some proposals in its JScript language, but eventually it stopped collaborating on Ecma work. Thus ECMAScript 4 was mothballed.

Growth and standardization

During the period of Internet Explorer dominance in the early 2000s, client-side scripting was stagnant. This started to change in 2004, when the successor of Netscape, Mozilla, released the Firefox browser. Firefox was well received by many, taking significant market share from Internet Explorer.

In 2005, Mozilla joined ECMA International, and work started on the ECMAScript for XML (E4X) standard. This led to Mozilla working jointly with Macromedia (later acquired by Adobe Systems), who were implementing E4X in their ActionScript 3 language, which was based on an ECMAScript 4 draft. The goal became standardizing ActionScript 3 as the new ECMAScript 4. To this end, Adobe Systems released the Tamarin implementation as an open source project. However, Tamarin and ActionScript 3 were too different from established client-side scripting, and without cooperation from Microsoft, ECMAScript 4 never reached fruition.

Meanwhile, very important developments were occurring in open-source communities not affiliated with ECMA work. In 2005, Jesse James Garrett released a white paper in which he coined the term Ajax and described a set of technologies, of which JavaScript was the backbone, to create web applications where data can be loaded in the background, avoiding the need for full page reloads. This sparked a renaissance period of JavaScript, spearheaded by open-source libraries and the communities that formed around them. Many new libraries were created, including jQuery, Prototype, Dojo Toolkit, and MooTools.

A major addition to the specification were event listeners, which date back to at least the early 2000s. However, Microsoft Internet Explorer only supported a proprietary method named "attachEvent" before version 9, released in 2011, making "onclick" events preferred for compatibility.

Google debuted its Chrome browser in 2008, with the V8 JavaScript engine that was faster than its competition. The key innovation was just-in-time compilation (JIT), so other browser vendors needed to overhaul their engines for JIT.

In July 2008, these disparate parties came together for a conference in Oslo. This led to the eventual agreement in early 2009 to combine all relevant work and drive the language forward. The result was the ECMAScript 5 standard, released in December 2009.

Reaching maturity

Ambitious work on the language continued for several years, culminating in an extensive collection of additions and refinements being formalized with the publication of ECMAScript 6 in 2015.

The creation of Node.js in 2009 by Ryan Dahl sparked a significant increase in the usage of JavaScript outside of web browsers. Node combines the V8 engine, an event loop, and I/O APIs, thereby providing a stand-alone JavaScript runtime system. As of 2018, Node had been used by millions of developers, and npm had the most modules of any package manager in the world.

The ECMAScript draft specification is currently maintained openly on GitHub, and editions are produced via regular annual snapshots. Potential revisions to the language are vetted through a comprehensive proposal process. Now, instead of edition numbers, developers check the status of upcoming features individually.

The current JavaScript ecosystem has many libraries and frameworks, established programming practices, and substantial usage of JavaScript outside of web browsers. Plus, with the rise of single-page applications and other JavaScript-heavy websites, several transpilers have been created to aid the development process.

Website client-side usage

JavaScript is the dominant client-side scripting language of the Web, with 97% of websites using it for this purpose. Scripts are embedded in or included from HTML documents and interact with the DOM. All major web browsers have a built-in JavaScript engine that executes the code on the user's device.

Run-time environment

JavaScript typically relies on a run-time environment (e.g., a web browser) to provide objects and methods by which scripts can interact with the environment (e.g., a web page DOM). These environments are single-threaded. JavaScript also relies on the run-time environment to provide the ability to include/import scripts (e.g., HTML <script> elements). This is not a language feature per se, but it is common in most JavaScript implementations. JavaScript processes messages from a queue one at a time. JavaScript calls a function associated with each new message, creating a call stack frame with the function's arguments and local variables. The call stack shrinks and grows based on the function's needs. When the call stack is empty upon function completion, JavaScript proceeds to the next message in the queue. This is called the event loop, described as "run to completion" because each message is fully processed before the next message is considered. However, the language's concurrency model describes the event loop as non-blocking: program input/output is performed using events and callback functions. This means, for instance, that JavaScript can process a mouse click while waiting for a database query to return information.

Variadic functions

An indefinite number of parameters can be passed to a function. The function can access them through formal parameters and also through the local arguments object. Variadic functions can also be created by using the bind method.

Array and object literals

Like many scripting languages, arrays and objects (associative arrays in other languages) can each be created with a succinct shortcut syntax. In fact, these literals form the basis of the JSON data format.

Regular expressions

JavaScript also supports regular expressions in a manner similar to Perl, which provide a concise and powerful syntax for text manipulation that is more sophisticated than the built-in string functions.

Promises and Async/await

JavaScript supports promises and Async/ await for handling asynchronous operations. A built-in Promise object provides functionality for handling promises and associating handlers with an asynchronous action's eventual result.

4.2 Node JS

Node.js is an open-source, cross-platform, back-end JavaScript runtime environment that runs on the V8 engine and executes JavaScript code outside a 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 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).

The Node.js distributed development project was previously governed by the Node.js Foundation, and has now merged with the JS Foundation to form the OpenJS Foundation, which is facilitated by the Linux Foundation's Collaborative Projects program.

Corporate users of Node.js software include GoDaddy, Groupon, IBM, LinkedIn, Microsoft, Netflix, PayPal, Rakuten, SAP, Walmart, Yahoo!, and Amazon Web Services.

Node.js is an open-source and cross-platform runtime environment for executing JavaScript code outside a browser. NodeJS is not a framework and it's not a programming language. Most people are confused and understand it's a framework or a programming language. We often use Node.js for building back-end services like APIs like Web App or Mobile App. It's used in production by large companies such as Pay pal, Uber, Netflix, Walmart, and so on.

BENEFITS OF NODE JS

The ability to scale up quickly — Each of the nodes in Node.js is based around an "event." A customer buys an in-app purchase, or sends an email to customer service, for instance. The amount of nodes you can add to your core programming function are nearly limitless. This means you can scale vertically, adding new capability paths that lead back to your core application code. Or, you can scale horizontally, adding new resources to existing programming. Either way, scalability gives your application room to grow, and that's one of the key benefits of using Node.js.

Speed and Performance — Its non-blocking, input-output operations make the environment one of the speediest options available. Code runs quickly, and that enhances the entire run-time environment. This is largely due to its sectioned-off system. But it also has to do with the fact that it runs on Google's V8 JavaScript engine. Its apps are more likely to be programmed end-to-end in Javascript, and that plug and play interoperability contributes to speed and performance.

Flexibility — In a discussion of Node.js pros and cons programming flexibility is perhaps the biggest benefit of all. When you make a change in Node.js, only that node is affected. Where other run time environments or frameworks may require you to make changes all the way back to the core programming, it doesn't require anything more than a change to the node. And that is great not just for your initial build, but for your ongoing maintenance as well. And best of all, when you combine JSON with Node.js, you can easily exchange information between client servers and the webserver. Programmers can also use APIs to build TCP, HTTP, DNS, etc. into the server.

The accessibility of a single programming language — Because it Is powered by JavaScript, programmers can easily tie nodes into the rest of the full-stack development. This makes it

easier for front-end developers to take on more difficult back-end programming tasks. No other server-side languages are necessary. And that’s a good thing because it speeds up development processing in almost every area and allows programmers of all levels easier access to your mobile apps’ back end. It’s one of the key benefits of Node.js.
Efficient caching — In a debate over the pros and cons of Node.js, caching always comes up as a key Node.js benefit. It has a powerful ability to cache data. When requests are made to the app, they are cached in-app memory. Consequently, when requests cycle through execution and re-execution, the nodes are still able to run efficiently and not be bogged down by historical data.
Fast-to-market-development — Node’s basis in JavaScript brings many benefits to the table, especially the ease at which developers can add more features and predesigned tools and templates. In fact, it has an extensive package management library, with thousands of open-source options that can be added to your app project immediately. These pre-packaged options not only reduce your time to market, but they reduce your programming budget, as well. When evaluating Node’s pros and cons, this is a bottom-line benefit that can’t be ignored.
An active user community — No discussion of Node.js advantages and disadvantages would be complete without discussing the huge benefits of its enormous user community. Many thousands of programmers around the globe belong to this user community, and they are very helpful to each other, providing each other with new, open-source code for every type of mobile application need. If there’s something that’s hanging up your design team, chances are, quick answers can be found through the community.
Efficient Queueing of Requests — A critical benefit of using Node.js is its ability to handle multiple requests at once. How does it do this? By offering users an optional nob-block I/O system. The system works by giving priority to those requests that take the lowest response time. This prioritization helps speed up the overall running of your app, especially when comparing it to other languages like Python or Ruby on Rails.
Easy to master — Node JS is easy to learn, primarily because it’s based on JavaScript. This makes it easier for your front-end programmers or designers to learn more complicated server-side programming with ease.

A better choice for mobile — This may very well be the biggest benefit of using node JS of all. Its fast development times, ease of use, and ability to scale up with increased traffic makes it an indispensable tool for getting your company into the mobile mainstream. If you’re looking to create a healthcare app, for instance, and grow your business with this immersive, fast-running mobile application, Node makes it easy for you. And in today’s rapid mobile deployment environment, this is no small thing.

Real-time communication — It maintains a steady connection between the user and the server, and that means there’s no lag between what a user asks for, and the server processing the request. While other run-time environments are trying to parse out a message letter by letter, it has already processed the request whole. It’s also important to note that it supports Websockets, one of the industry’s most popular real-time communication solutions. With Node.js benefits like this one, you can ensure users are always in sync with you.

Node Package Manager for Enterprise — Its many available packages are available free and open-source, over the open web. And that can be a security risk, especially for large, enterprise-level organizations. In the debate about Node.js pros and cons, this decision can be tricky. But we believe it has turned this into an advantage with its package manager for enterprise. The enterprise manager neatly circumvents this risk by allowing the packages to run behind the company’s firewall. It also offers businesses a private registry with high-end security features. This allows the company to identify vulnerabilities, replace unsafe code, and control who can and cannot see their code. Enterprise-level companies looking to hire a dedicated team of Node.js developers should definitely make sure they have experience with its enterprise-level software solutions, and how UX/UI is affected in this environment.

Cross-platform development — Do you need to have a mobile app that also links to a desktop app? Node has you covered here, too. Much of the code you use for your mobile app can be directly transferred to desktop applications, especially in the macOS ecosystem. And that cuts down on development time and costs significantly. And who couldn’t use that particular Node.js advantage?

Disadvantages of Using Node.JS

The disadvantages are a far shorter list. But no software is perfect, and Node is no exception. The program does have some serious drawbacks that are worth considering.

Inability to process CPU bound tasks quickly — It is considered single-threaded, because it

processes JavaScript, which is, of course, single-threaded. Its non-blocking input/output model uses an event loop to process threads asynchronously. And this works great—until Node receives a CPU-bound task. It prioritizes these heavy, CPU bound tasks first, and that results in slow processing and overall delay in the event loop. This is why so many programmers say Node.js is not a good fit for apps that require heavy computation. It's important to note that in 2018, it rolled out a multi-threading tool called worker threads as part of its 10.5.0 update. This module can leverage additional threads from a thread pool, so heavy parallel processes can be executed on different threads. It's still considered experimental, but it could go a long way towards alleviating this obvious flaw in the future. Still, it's one of Node's pros and cons you have to weigh very carefully.
Callback hell can put your code in a loop — If you're keeping a number of queued tasks in running in the background, lookout. You could fall into what Node programmers often refer to as "callback hell." This happens callbacks get nested several layers deep.
Let's talk about how that happens. Callbacks are the functions that run after a task is finished. Have too many of these running in the background, and their callbacks can get nested within each other, as is noted in this graphic below. Node's asynchronous nature is partly to blame for this, but really, it happens when you're not running clean code, or bad code structures.

An overgrown, yet immature npm module registry — Node.js has an enormous user community that has produced thousands of open source modules that are there for the taking. And that's a problem because it can be very difficult to know if the node you are downloading is a quality program that's vetted or properly documented. It's hard to search the community for nodes based on quality or ratings. And with only a portion of its core technology monitored and vetted by Joyent, that can make it hard to find enterprise-ready nodes to use.

Hot competition for Node programmers — While a majority of computer programmers and mobile app developers can program in Javascript, not all of them have the engineering expertise to work with this environment. And that means node developers are a hot commodity, and competition for talent is fierce.

Unstable API — Its Application User Interface has gone through several changes. And not all those changes are backwards compatible. That means developers have to make changes in the accessible code bases, just so they can maintain compatibility with the latest version of the

Node.js API. Sometimes, it’s not exactly ideal.
A weak library system — In spite of JavaScript’s global popularity, its library system leaves much to be desired. Because of this, programmers often have to take on the support of some of its more complex functions, like XML parsing, processing of images, database operations, or object-relational mapping. It means that several foundational tasks in Node can be hard to implement.

4.3 MYSQL

MySQL is an open-source relational database management system (RDBMS). Its name is a combination of "My", the name of co-founder Michael Widenius's daughter, and "SQL", the abbreviation for Structured Query Language. A relational database organizes data into one or more data tables in which data may be related to each other; these relations help structure the data. SQL is a language programmers use to create, modify and extract data from the relational database, as well as control user access to the database. In addition to relational databases and SQL, an RDBMS like MySQL works with an operating system to implement a relational database in a computer's storage system, manages users, allows for network access and facilitates testing database integrity and creation of backups.

MySQL is free and open-source software under the terms of the GNU General Public License, and is also available under a variety of proprietary licenses. MySQL was owned and sponsored by the Swedish company MySQL AB, which was bought by Sun Microsystems (now Oracle Corporation).In 2010, when Oracle acquired Sun, Widenius forked the open-source MySQL project to create MariaDB.

MySQL has stand-alone clients that allow users to interact directly with a MySQL database using SQL, but more often, MySQL is used with other programs to implement applications that need relational database capability. MySQL is a component of the LAMP web application software stack (and others), which is an acronym for Linux, Apache, MySQL, Perl/PHP/Python. MySQL is used by many database-driven web applications, including Drupal, Joomla, phpBB, and WordPress. MySQL is also used by many popular websites, including Facebook, Flickr, MediaWiki, Twitter, and YouTube.

Advantages of SQL

Faster Query Processing
Large amount of data is retrieved quickly and efficiently. Operations like Insertion, deletion, manipulation of data is also done in almost no time.

No Coding Skills
For data retrieval, large number of lines of code is not required. All basic keywords such as SELECT, INSERT INTO, UPDATE, etc are used and also the syntactical rules are not complex in SQL, which makes it a user-friendly language.

Standardized Language
Due to documentation and long establishment over years, it provides a uniform platform worldwide to all its users.

Portable –
It can be used in programs in PCs, server, laptops independent of any platform (Operating System, etc). Also, it can be embedded with other applications as per need/requirement/use.

Interactive Language
Easy to learn and understand, answers to complex queries can be received in seconds.

Disadvantages of SQL

Complex Interface
SQL has a difficult interface that makes few users uncomfortable while dealing with the database.

Cost
Some versions are costly and hence, programmers cannot access it.

Partial Control

Due to hidden business rules, complete control is not given to the database.

4.4 HTML

HTML (HyperText Markup Language) is the most basic building block of the Web. It defines the meaning and structure of web content. Other technologies besides HTML are generally used to describe a web page's appearance/presentation (CSS) or functionality/behavior (JavaScript).

"Hypertext" refers to links that connect web pages to one another, either within a single website or between websites. Links are a fundamental aspect of the Web. By uploading content to the Internet and linking it to pages created by other people, you become an active participant in the World Wide Web.

HTML uses "markup" to annotate text, images, and other content for display in a Web browser. HTML markup includes special "elements" such as <head>, <title>, <body>, <header>, <footer>, <article>, <section>, <p>, <div>, <span>, <img>, <aside>, <audio>, <canvas>, <datalist>, <details>, <embed>, <nav>, <output>, <progress>, <video>, <ul>, <ol>, <li> and many others.

An HTML element is set off from other text in a document by "tags", which consist of the element name surrounded by "<" and ">". The name of an element inside a tag is case insensitive. That is, it can be written in uppercase, lowercase, or a mixture. For example, the <title> tag can be written as <Title>, <TITLE>, or in any other way. However, the convention and recommended practice is to write tags in lowercase.

4.5 CSS (CASCADING STYLE SHEETS)

Cascading Style Sheets (CSS) is a stylesheet language used to describe the presentation of a document written in HTML or XML (including XML dialects such as SVG, MathML or XHTML). CSS describes how elements should be rendered on screen, on paper, in speech, or on other media.

CSS is among the core languages of the open web and is standardized across Web browsers according to W3C specifications. Previously, development of various parts of CSS specification was done synchronously, which allowed versioning of the latest recommendations. You might have heard about CSS1, CSS2.1, CSS3. However, CSS4 has never become an official version.

From CSS3, the scope of the specification increased significantly and the progress on different CSS modules started to differ so much, that it became more effective to develop and release recommendations separately per module. Instead of versioning the CSS specification, W3C now periodically takes a snapshot of the latest stable state of the CSS specification.

CSS plays an important role, by using CSS you simply got to specify a repeated style for element once & use it multiple times as because CSS will automatically apply the required styles.

Web designers needs to use few lines of programming for every page improving site speed. Cascading sheet not only simplifies website development, but also simplifies the maintenance as a change of one line of code affects the whole web site and maintenance time.

It is less complex therefore the effort are significantly reduced.

It helps to form spontaneous and consistent changes.

CSS changes are device friendly. With people employing a batch of various range of smart devices to access websites over the web, there’s a requirement for responsive web design.
It has the power for re-positioning. It helps us to determine the changes within the position of web elements who are there on the page.

These bandwidth savings are substantial figures of insignificant tags that are indistinct from a mess of pages.

Easy for the user to customize the online page

It reduces the file transfer size.

There are some disadvantages of CSS

CSS, CSS 1 up to CSS3, result in creating of confusion among web browsers.

With CSS, what works with one browser might not always work with another. The web developers need to test for compatibility, running the program across multiple browsers.

There exists a scarcity of security.

After making the changes we need to confirm the compatibility if they appear. The similar change affects on all the browsers.

The programing language world is complicated for non-developers and beginners. Different levels of CSS i.e. CSS, CSS 2, CSS 3 are often quite confusing.

Browser compatibility (some styles sheet are supported and some are not).
CSS works differently on different browsers. IE and Opera supports CSS as different logic.

There might be cross-browser issues while using CSS.

There are multiple levels which creates confusion for non-developers and beginners.

HTML

HTML (HyperText Markup Language) is the most basic building block of the Web. It defines the meaning and structure of web content. Other technologies besides HTML are generally used to describe a web page's appearance/presentation (CSS) or functionality/behavior (JavaScript).

"Hypertext" refers to links that connect web pages to one another, either within a single website or between websites. Links are a fundamental aspect of the Web. By uploading content to the Internet and linking it to pages created by other people, you become an active participant in the World Wide Web.

HTML uses "markup" to annotate text, images, and other content for display in a Web browser. HTML markup includes special "elements" such as <head>, <title>, <body>, <header>, <footer>, <article>, <section>, <p>, <div>, <span>, <img>, <aside>, <audio>, <canvas>, <datalist>, <details>, <embed>, <nav>, <output>, <progress>, <video>, <ul>, <ol>, <li> and many others.

An HTML element is set off from other text in a document by "tags", which consist of the element name surrounded by "<" and ">". The name of an element inside a tag is case insensitive. That is, it can be written in uppercase, lowercase, or a mixture. For example, the <title> tag can be written as <Title>, <TITLE>, or in any other way. However, the convention and recommended practice is to write tags in lowercase.

CSS (CASCADING STYLE SHEETS)

Cascading Style Sheets (CSS) is a stylesheet language used to describe the presentation of a document written in HTML or XML (including XML dialects such as SVG, MathML or XHTML). CSS describes how elements should be rendered on screen, on paper, in speech, or on other media.

CSS is among the core languages of the open web and is standardized across Web browsers according to W3C specifications. Previously, development of various parts of CSS specification was done synchronously, which allowed versioning of the latest recommendations. You might have heard about CSS1, CSS2.1, CSS3. However, CSS4 has never become an official version.

From CSS3, the scope of the specification increased significantly and the progress on different CSS modules started to differ so much, that it became more effective to develop and release recommendations separately per module. Instead of versioning the CSS specification, W3C now periodically takes a snapshot of the latest stable state of the CSS specification.

CSS plays an important role, by using CSS you simply got to specify a repeated style for element once & use it multiple times as because CSS will automatically apply the required styles.

Web designers needs to use few lines of programming for every page improving site speed. Cascading sheet not only simplifies website development, but also simplifies the maintenance as a change of one line of code affects the whole web site and maintenance time.

It is less complex therefore the effort are significantly reduced.

It helps to form spontaneous and consistent changes.

CSS changes are device friendly. With people employing a batch of various range of smart devices to access websites over the web, there’s a requirement for responsive web design.
It has the power for re-positioning. It helps us to determine the changes within the position of web elements who are there on the page.
These bandwidth savings are substantial figures of insignificant tags that are indistinct from a mess of pages.

Easy for the user to customize the online page

It reduces the file transfer size.

There are some disadvantages of CSS

CSS, CSS 1 up to CSS3, result in creating of confusion among web browsers.

With CSS, what works with one browser might not always work with another. The web developers need to test for compatibility, running the program across multiple browsers.

There exists a scarcity of security.

After making the changes we need to confirm the compatibility if they appear. The similar change affects on all the browsers.

The programing language world is complicated for non-developers and beginners. Different levels of CSS i.e. CSS, CSS 2, CSS 3 are often quite confusing.

Browser compatibility (some styles sheet are supported and some are not).
CSS works differently on different browsers. IE and Opera supports CSS as different logic.

There might be cross-browser issues while using CSS.

There are multiple levels which creates confusion for non-developers and beginners.

CHAPTER 5
CODING AND IMPLEMENTATION

Manager_details.ejs

```
<!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">
<link href="/public/assets/style.css" rel="stylesheet" type="text/css" />
<title>Document</title>
</head>

<body>
<!-- Nav Bar -->
<ul>
<li><a href="/employee_dashboard">Dashboard</a></li>

<li><a href="/logout">Logout</a></li>
</ul>
<!-- Nav Bar ends -->

<div class="main">
<h1>Managers Profile</h1>
<table>
<tr>
<th>Name</th>
<th>Email</th>
</tr>
<% users.forEach(function(user){ %>
<tr>
<td><%= user.Name %></td>
<td><%= user.Email %></td>
</tr>
<%
})
%>


</table>
</div>
</body>

</html>
```

admin_login.ejs

```
<!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">
  <link href= "/public/assets/style.css" rel="stylesheet" type="text/css" />
  <title>Document</title>
</head>
<body>
    <!-- Nav Bar -->
    <ul>
      <li><a href="/">Home </a></li>

    </ul>
    <!-- Nav bar ends  -->
  <div class="main">
  <h1> Admin Operations : </h1></br>

  <h1> Enter your credintials : </h1>
  <form method="POST" action="/login" >

    <label for="Email"> Enter Email :</label>
    <input type="email" name = "Email" placeholder="Enter Email" required/>
  </br>
    <label for="password"> </label>
    <input type="password" name="Password" Enter Password: placeholder="Enter
Password" required/>
</br>
    <input type ="submit" name = "Login" value="login"/>

  </form>

  <h1> Do You Want TO SignUP: </h1>
  <a href="/SignUp"> <input type="button" value="Create Profile" name="Profile">
   </a>
   </br>
  </div>
</body>
```

displayTable.ejs

```
<!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">
  <link href="/style/style.css" rel="stylesheet" type="text/css">

  <link href= "/public/assets/style.css" rel="stylesheet" type="text/css" />

  <title>Document</title>
</head>
<body>
   <!-- Nav Bar -->
   <ul>
    <li><a href="/welcome">Add</a></li>
    <li><a href="/show">View</a></li>
    <li><a href="/updateEmployee">Update</a></li>
    <li><a href="/deleteEmployee">Delete</a></li>
    <li><a href="/logout">Logout</a></li>
  </ul>

  <!-- NavBar ends  -->

  <div class="main">
  <table>
    <tr>
      <th>
        Name
      </th>
      <th>
        Email
      </th>
      <th>
        Skills
      </th>
      <th>
        Phone
      </th>
      <th>
       Manager
      </th>
      <th>
        Band
      </th>

    </tr>
    <% users.forEach(function(user){ %>
      <tr>
        <td><%= user.Name %></td>
        <td><%= user.Email %></td>
        <td><%= user.Skills %></td>
        <td><%= user.Phone %></td>
        <td><%= user.Manager %></td>
        <td><%= user.Band %></td>

           </tr>
      <%
    })
    %>
  </table>
  </div>

</body>
</html>
```

edit_employee_details.ejs

```
<!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">
  <link href="/public/assets/style.css" rel="stylesheet" type="text/css" />
  <title>Document</title>
</head>

<body>

  <ul>
    <li><a href="/employee_dashboard">DashBoard</a></li>

    <li ><a href="/employee_logout">Logout</a></li>
  </ul>

  <div class="main">
      <h1>LoggedIn as: <% users.forEach((user)=>{ %>
      <%= user.Name %>
    <% }) %> </h1>

    <form method="post" action="/updateSkills">
      <h3> Update Skills  </h3>
      <input type="text" placeholder="Enter New Skills" Name="skill" required/>
      <input type="submit" value="Update Skills"/>
    </form>

    <form method="post" action="/change_employee_password" >
      <h3> Update Password </h3>
      <label for="password"> Enter new Password: </label>
      <input type="password" name="Password"/>
      <input type ="submit" name = "Change Password"
value="change_employee_password"/>
    </form>

    <form method="post" action="/change_employee_phone" >
      <h3> Update Phone Number </h3>
      <input type="tel" name="Phone" placeholder="Enter new Phone Number"/>
      <input type ="submit" value="Change Employee Phone"/>
    </form>


    <form method="post" action="/change_employee_location" >
      <h3> Edit Location </h3>
      <input type="text" name="Location" placeholder="Enter new City"/>
      <input type ="submit" value="Change Employee Location"/>
    </form>
  </div>
</body>

</html>
```

employee_login.ejs

```
<!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">
  <link href= "/public/assets/style.css" rel="stylesheet" type="text/css" />
  <title>Document</title>
</head>
```

```
<body>
  <ul>
    <li><a href="/"> Home</a></li>
  </ul>
  <div class="main">

  <h1> Login as Employee </h1>
  <form method="post" action="/SignUpEmployee" >



    <label for="Email"> Enter Email :</label>
    <input type="email" name = "Email"/>
  </br>
    <label for="password"> Enter Password: </label>
    <input type="password" name="Password"/>
</br>
    <input type ="submit" name = "login" value="login"/>

  </form>
  </div>
</body>
</html>
```

employee_main.ejs

```
<!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">
  <link href="/public/assets/style.css" rel="stylesheet" type="text/css" />
  <title>Document</title>
</head>

<body>
   <!-- Nav Bar -->
   <ul>

     <li ><a href="/employee_logout">Logout</a></li>
   </ul>

   <!-- NavBar ends  -->
  <div class="main">
   <h1>Employee Dashboard</h1>

     <h2>welcome <% users.forEach( function (user) { %>
     <%=  user.Name %>
    <% }) %> </h2>
    <br/>
    <br/>
    <a href="/searchEmployee" ><input type="button" value= "Search Employees" /></a>
    <a href="/edit_employee_details"><input type="button" value= "Edit Details" /></a>
    <a href='/Manager_details'><input type="button" value = "Manager Details"></a>



  </div>
</body>

</html>
```

login.ejs

```
<!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">
  <link href= "/public/assets/style.css" rel="stylesheet" type="text/css" />
  <title>Document</title>
</head>
<body>
  <div class="main">
  <h1> Login as Admin </h1>
  <form method="post" action="/SignUp" >

    <label for="name"> Enter Name :</label>
    <input type="text" name = "Name"/>
  </br>
    <label for="Email"> Enter Email :</label>
    <input type="email" name = "Email"/>
  </br>
    <label for="password"> Enter Password: </label>
    <input type="password" name="Password"/>
</br>
    <input type ="submit" name = "SignUP" value="SignUp"/>

  </form>
  </div>
</body>
</html>
```

main.ejs

```
<!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">
  <link href= "/public/assets/style.css" rel="stylesheet" type="text/css" />
  <title>Document</title>
</head>
<body>
  <div class="main">
  <h1 id="emp">  Employee Management System   </h1>
  <a href="/login">
  <input type = "button" value="Admin Operations" name="Admin"/>
  </a>
  <a href="/employee_login">
    <input type = "button" value="Employee Operations" name="Employee"/>
  </a>

  </div>
</body>
```

searchEmployees.ejs

```
<!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">
  <link href="/public/assets/style.css" rel="stylesheet" type="text/css" />
  <title>Document</title>
</head>

<body>
```

```
  <ul>
    <li><a href="/employee_dashboard">DashBoard</a></li>

    <li ><a href="/employee_logout">Logout</a></li>
  </ul>

  <div class="main">
      <h1>LoggedIn as: <% users.forEach((user)=>{ %>
      <%= user.Name %>
    <% }) %> </h1>

    <form method = "post" action="/search_by_name">
      <input type="text" placeholder="Enter Name of Employee"  Name = "Name" />
      <input type ="submit" value= "Search By Name" />
    </form>

    <form method = "post" action="/search_by_project">
      <input type="text" placeholder="Enter Project Name"  Name = "Project" />
      <input type ="submit" value= "Search By Project" />
    </form>

    <form method = "post" action="/search_by_designation">

      <input type="Number" placeholder="Enter Designation Band "  Name = "Designation"
max="7" min="1" />
      <p> The value of Band should be between 1 - 7  </p>
      <input type ="submit" value= "Search By Designation" />
    </form>

    <form method = "post" action="/search_by_address">

      <input type="text" placeholder="City "  Name = "Location" />
      <p>Enter employee's City</p>
      <input type ="submit" value= "Search By Address" />
    </form>

  </div>
</body>

</html>
```

showEmployee.ejs

```
<!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">
  <link href="/style/style.css" rel="stylesheet" type="text/css">

  <link href= "/public/assets/style.css" rel="stylesheet" type="text/css" />

  <title>Document</title>
</head>
<body>
    <!-- Nav Bar -->
    <ul>
      <li><a href="/welcome">Add</a></li>
      <li><a href="/show">View</a></li>
      <li><a href="/updateEmployee">Update</a></li>
      <li><a href="/deleteEmployee">Delete</a></li>
      <li><a href="/logout">Logout</a></li>
    </ul>
  <div class = "main" >

  <table>
    <tr>
```

```
      <th>
        Name
      </th>
      <th>
        Email
      </th>
      <th>
        Skills
      </th>
      <th>
        Phone
      </th>
      <th>
       Manager
      </th>
      <th>
        Band
      </th>
    <th>
      Delete user
    </th>
    </tr>
    <% users.forEach(function(user){ %>
      <tr>
        <td><%= user.Name %></td>
        <td><%= user.Email %></td>
        <td><%= user.Skills %></td>
        <td><%= user.Phone %></td>
        <td><%= user.Manager %></td>
        <td><%= user.Band %></td>

        <td><a href='/delete/<%= user.Phone %>' ><input type="button" class = "delete"
value="Delete"/></a>
        </td>
      </tr>
      <%
    })
    %>
  </table>
  </div>
</body>
</html>
```

show_employ_side.ejs

```
<!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">
  <link href="/style/style.css" rel="stylesheet" type="text/css">

  <link href= "/public/assets/style.css" rel="stylesheet" type="text/css" />

  <title>Document</title>
</head>
<body>
   <!-- Nav Bar -->
   <ul>
    <li><a href="/employee_dashboard">Dashboard</a></li>

    <li><a href="/logout">Logout</a></li>
  </ul>
```

```
<div class="main">
<table>
  <tr>
    <th>
      Name
    </th>
    <th>
      Email
    </th>
    <th>
      Skills
    </th>
    <th>
      Phone
    </th>
    <th>
     Manager
    </th>
    <th>
      Band
    </th>

  </tr>
  <% users.forEach(function(user){ %>
    <tr>
      <td><%= user.Name %></td>
      <td><%= user.Email %></td>
      <td><%= user.Skills %></td>
      <td><%= user.Phone %></td>
      <td><%= user.Manager %></td>
      <td><%= user.Band %></td>
```

```
          </tr>
      <%
    })
    %>
  </table>
  </div>

</body>
</html>
```

updateEmployee.ejs

```
<!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">
  <link href= "/public/assets/style.css" rel="stylesheet" type="text/css" />
  <title>Document</title>
</head>
<body>
    <!-- Nav Bar -->
    <ul>
      <li><a href="/welcome">Add </a></li>
      <li><a href="/show">View </a></li>
      <li><a href="/updateEmployee">Update</a></li>
      <li><a href="/deleteEmployee">Delete</a></li>
      <li ><a href="/logout">Logout</a></li>
    </ul>
    <!-- Nav bar ends  -->
  <div class = "main">
  <table>
    <tr>
    <th>
      Name
    </th>
    <th>
      Email
    </th>
    <th>
      Skills
    </th>
    <th>
      Phone
    </th>
    <th>
     Manager
    </th>
    <th>
      Band
    </th>

  </tr>
  <% users.forEach(function(user){ %>
    <tr>
      <td><%= user.Name %></td>
      <td><%= user.Email %></td>
      <td><%= user.Skills %></td>
      <td><%= user.Phone %></td>
      <td><%= user.Manager %></td>
      <td><%= user.Band %></td>


         </tr>
    <%
  })
  %>
</table>
```

```
    </br>
    </br>
  <h1> Update Employee Details : </h1></br>



  <form method="POST" action="/update" >
    <h1> Enter Name and Phone OF Employee Whose Details
      You Want To Update </h1>
    <label for="Name"> Enter Empoyee Name :</label>
    <input type="text" name = "Name" placeholder="Enter Name" required/>
  </br>
    <label for="Phone"> </label>
    <input type="tel" name="Phone"  placeholder="Enter Phone Number" required/>
</br>
<h1>Enter New Details:</h1>
  <label for="Project"> New Project</label>
  <input type="text" name="Project"  placeholder="Enter New Project " required/>
</br>
<label for="Manager">New Manager</label>
<input type="text" name="Manager"  placeholder="Enter New Project " required/>
<label for="Manager">New Band</label>
<input type="number" name="Band"  placeholder="Enter New Band " min="0" max="7"
required/>
</br>



    <input type ="submit" name = "Update" value="Update"/>

  </form>



  </div>
</body>
</html>
```

welcome.ejs

```
<!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">
  <link href= "/public/assets/style.css" rel="stylesheet" type="text/css" />
  <title>Document</title>
</head>
<body>
  <!-- Nav Bar -->
  <ul>
    <li><a href="/welcome">Add</a></li>
    <li><a href="/show">View</a></li>
    <li><a href="/updateEmployee">Update</a></li>
    <li><a href="/deleteEmployee">Delete</a></li>
    <li><a href="/logout">Logout</a></li>
  </ul>

  <!-- NavBar ends  -->




<div class="main">
  <h1>Admin Dashboard</h1>
  <h2> Add a Employee  </h2>

  <form method="POST" action="/insertEmployee">
```

```
   <label for="Name"> Employee Name </label> </br>
   <input type ="text" name="Name" placeholder="Enter Employee Name" required/>

   <label for="Email"> Email :</label> </br>
   <input type ="text" name="Email" placeholder="Enter Employee Email`" required/>

   <label for="Location"> City :</label> </br>
   <input type ="text" name="Location" placeholder="Enter Employee City" required/>

   <label for="Project"> Project :</label> </br>
   <input type ="text" name="Project" placeholder="Enter Employee Project" required/>

   <label for="Skills"> Skills :</label> </br>
   <input type ="text" name="Skills" placeholder="Enter skill set " required/>
   <label for="Email" name="phone" > Contact :</label> </br>
   <input type ="tel" name="Phone" placeholder="Enter Contact Number" required/>

   <label for="Manager"> Manager :</label> </br>
   <input type ="text" name="Manager" placeholder="Enter Employee Manager" required/>


   <label for="Band"> Band :</label> </br>
   <input type ="number" name="Band" placeholder="Enter Band of Employee" min="0"
max="7" required/>

   <input type ="submit" name = "Create Profile" value="CreateProfile"/>



  </form>

</div>
</body>
</html>
```

CHAPTER 6
REFERENCES

WEBSITES

https://developer.mozilla.org/en-US/docs/Web/JavaScript

https://en.wikipedia.org/wiki/Main_Page
(I took the help of the content of Wikipedia to get an authentic and latest information in writing technical description and on other chapters also)

https://www.geeksforgeeks.org/
(Took the help in writing description and various merits and demerits of technologies)

https://www.tutorialspoint.com/index.htm

https://www.javatpoint.com/

(Took the help in writing description and various merits and demerits of technologies)

Other references

- Fajfar, I. (2015). Start Programming Using HTML, CSS, and JavaScript (1st ed.). Chapman and Hall/CRC. https://doi.org/10.1201/b19402
- Wang, P.S. (2013). Dynamic Web Programming and HTML5 (1st ed.). Chapman and Hall/CRC.

- What part writer? What part programmer? A survey of practices and knowledge used in programmer writing

- M. A. P. Subali and S. Rochimah, "A new model for measuring the complexity of SQL commands," 2018 10th International Conference on Information Technology and Electrical Engineering (ICITEE), 2018, pp. 1-5, doi: 10.1109/ICITEED.2018.8534782.

- Extracting structures of HTML documents (Lim and Y)
  Published in: Proceedings Twelfth International Conference on Information Networking (ICOIN-12)
- Do you Really Code? Designing and Evaluating Screening Questions for Online Surveys with Programmers (2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)).