**PROJECT REPORT**

# on

**Web chef**

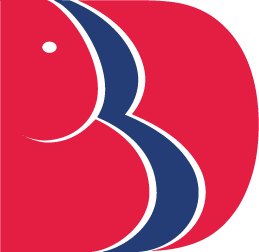
# for

# Digipodium

# Towards partial fulfillment of the requirement for the award of degree of

**Bachelor of Computer Applications**

# from

Babu Banarasi Das University Lucknow

**Academic Session 2022 - 23**

**School of Computer Applications**

|  |
| --- |
| **I Floor, H-Block, BBDU, BBD City, Faizabad Road, Lucknow (U. P.) INDIA 226028**  **PHONE: HEAD:** 0522-3911127, 3911321 **Dept. Adm. & Exam Cell:** 0522-3911326 **Dept. T&P Cell:** 0522-3911128; **E-Mail:** [head.sca@gmail.com](mailto:head.sca@gmail.com) |
| **w w w . b b d u . a c . i n** |

**PROJECT REPORT**

# on

**Web Chef**

# for

**Digipodium**

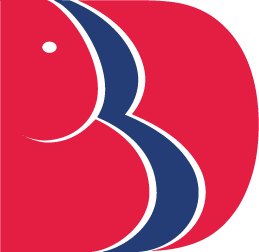
# towards partial fulfillment of the requirement for the award of degree of

**Bachelor of Computer Applications**

# from

**Babu Banarasi Das University**

**Lucknow**



**Developed and Submitted by- Under Guidance of-**

Gaurav Goel Mohammad Mubassir

1200211207

**Academic Session 2022 - 23**

**School of Computer Applications**

|  |
| --- |
| **I Floor, H-Block, BBDU, BBD City, Faizabad Road, Lucknow (U. P.) INDIA 226028**  **PHONE: HEAD:** 0522-3911127, 3911321 **Dept. Adm. & Exam Cell:** 0522-3911326 **Dept. T&P Cell:** 0522-3911128; **E-Mail:** [head.sca@gmail.com](mailto:head.sca@gmail.com) |
| **w w w . b b d u . a c . i n** |

**CERTIFICATE**

### This is to certify that Project Report entitled

### Web Chef

**being submitted by**

# Gaurav Goel

### towards the partial fulfilment of the requirement for the award of the degree of

**Bachelor of Computer Applications**

# to

**Babu Banarasi Das University**

**Lucknow**

### In The Academic Year 2022 - 23

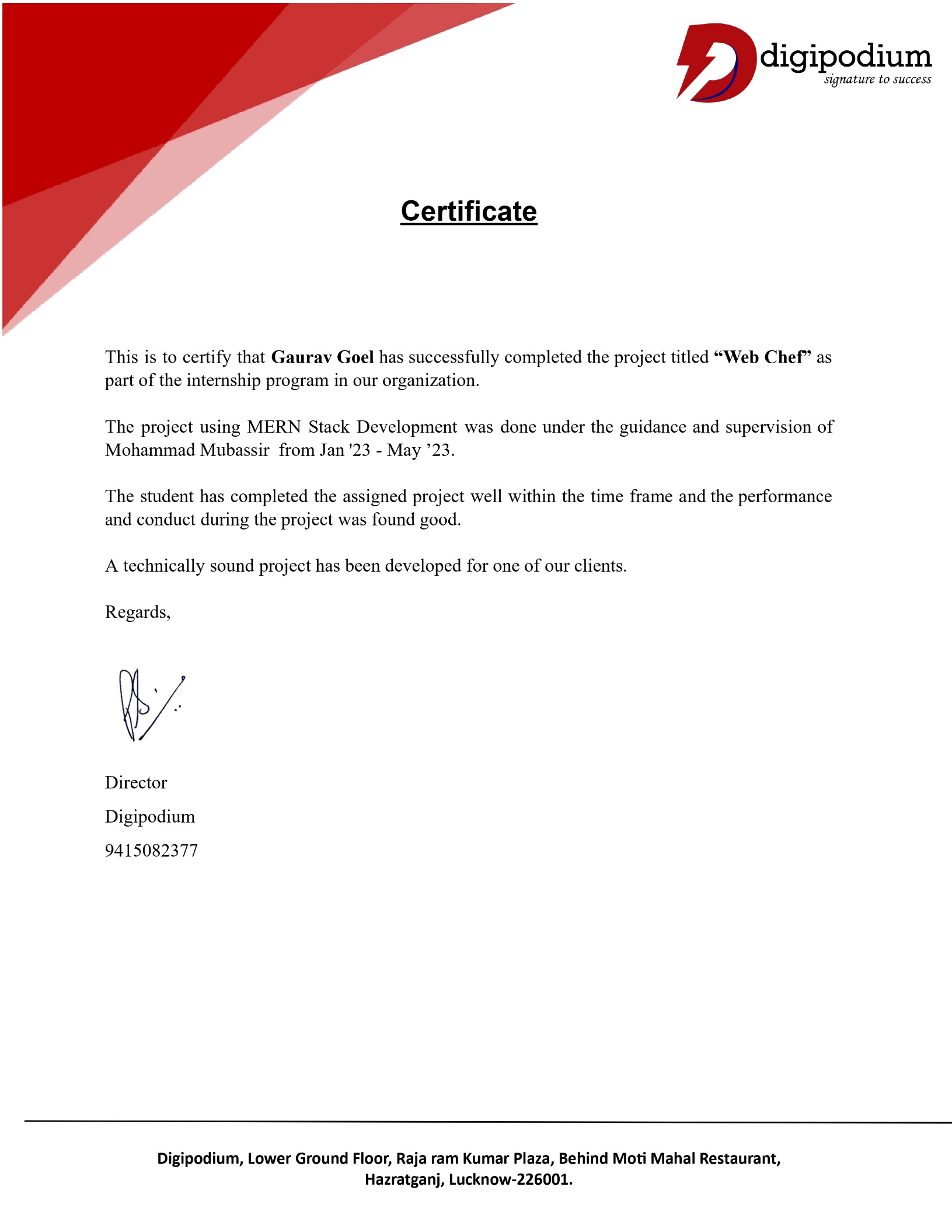
**Is a record of the student’s own work carried out at**

# Digipodium

**And to the best of our knowledge the work reported herein does not form a part of any other thesis or work on the basis of which degree or award was conferred on an earlier occasion to this or any other candidate.**

# Prabhash Ch. Pathak

**HEAD (School of Computer Applications)**



# ACKNOWLEDGEMENT

I have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. I would like to extend my sincere thanks to all of them.

I am highly indebted to DIGIPODIUM INSTITUTE for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project.

I would like to express my gratitude towards my friends & member of DIGIPODIUM for their kind co-operation and encouragement which help me in completion of this project.

I would like to express my special gratitude and thanks to group members for giving me such attention and time.

My thanks and appreciations also go to my colleague in developing the project and people who have willingly helped me out with their abilities.

# DECLARATION

I **Gaurav Goel & Ashutosh Tiwari** hereby declare that this project report entitled **WEB-CHEF,** submitted by us, under the guidance of Mohammad Mubassir of **DIGIPODIUM INSTITUTE, LUCKNOW** is our own and has not been submitted to any other University or Institute or published earlier.

**Signature of Student**:

**Gaurav Goel**

**Ashutosh Tiwari**

**BCA VIth Semester**

**Date: 14/06/2023**

# ABSTRACT

The web chef project aims to enable the training of machine learning (ML) models directly within web browsers, revolutionizing the way ML is performed and expanding the accessibility of AI capabilities. By leveraging technologies like JavaScript, NodeJS, ReactJS, etc., developers can build and deploy ML models that run entirely within the browser environment, reducing the dependency on cloud resources and enhancing privacy and data security. Web Chef empowers users to train ML models on their own devices without the need for extensive computational resources or specialized hardware, making ML training faster, more efficient, and more accessible. This project opens up new possibilities for a wide range of applications, such as natural language processing, computer vision, and recommendation systems, by enabling the creation of interactive websites and web applications that leverage ML capabilities. The Web Chef project represents a significant step towards democratizing ML and bringing the benefits of AI to a broader audience.

# TABLE OF CONTENTS

|  |  |  |
| --- | --- | --- |
| **S.No.** | **CONTENT** | **Page No.** |
|  | Introduction | 6 |
|  | Need of Identification   * Objective * Scope | 7 |
|  | Problem Statement | 8 |
|  | Proposed System | 9 |
|  | System Analysis   * Iterative Waterfall Model * Feasibility Study * Database Design * Use Case Diagrams * Class Diagrams * Modules * Module Description * Gantt Chart | 10-11  12-13-14  15  16-17-18  19-20-21  22  23  24 |
|  | * Software Requirement Specification (SRS) * Software Requirement * Hardware Requirement * Functional Requirement * Non- functional Requirement | 25  25  26  27-28  29-30 |
|  | Design & Implementation Constraints | 31-32 |
|  | Tools & Technology | 33-37 |
|  | Cost Estimation and Cost Constructive Model | 38-41 |
|  | Testing   * + Testing Phase   + Level of testing   + Verification and Validation (V&V) | 42  43-44  45 |
|  | Screenshots | 47-54 |
|  | Conclusion | 55 |
|  | Bibliography | 56 |

# INTRODUCTION

The web development project at hand is focused on revolutionizing the process of converting mockups and wireframes into fully functional web pages. Its primary objective is to provide a streamlined and efficient solution that saves designers and developers valuable time and effort. By automating this conversion process, the project aims to generate HTML, CSS, and JavaScript code that is clean, maintainable, and adheres to industry best practices. With a strong emphasis on optimization, the resulting code will be fine-tuned for optimal performance and cross-browser compatibility, ensuring a seamless user experience across different platforms. The project encompasses a wide range of features, including design file parsing, HTML structure generation, CSS styling, responsive design implementation, JavaScript interactivity, and customizable options, to cater to the diverse needs of web development professionals. Additionally, the project aims to provide a user-friendly interface, comprehensive documentation, and ongoing support to empower designers and developers in efficiently translating their visual designs into remarkable web experiences.

In summary, the web development project is dedicated to simplifying and enhancing the process of converting mockups and wireframes into HTML, CSS, and JavaScript code. Through automation, optimization, and a comprehensive set of features, the project aims to empower professionals in the web development community by offering a reliable tool that streamlines their workflow and enables them to deliver exceptional results.

# NEED OF IDENTIFICATION

**Objective:**

* **Reduced Development cost:** Start-Ups will not have to hire a web developer for developing a customizable dashboard or purchase a quite expensive dashboard from the market.

* **Detailed Insights:** Start-ups can have better and detailed insights about their sales and customer preferences, which will help them grow better and in right direction.
* **Better Marketing:** Start-ups can target their audience, and customize their data and Advertisements better with detailed insights, resulting in their growth.

**Scope:**

Our project would mainly focus on the start-ups and critics of their business or individual developers. It will help these start-ups or small companies to grow better with detailed insights, and better marketing, and individual developers by reducing their development cost, work and giving better insights about their application.

# PROBLEM STATEMENT

* The traditional website creation process involves finding a web designer and developer
* There are different platforms for creating Mockups.
* All of the few code generation Plugins are costly.

# PROPOSED SYSTEM

* **Get access to our web site**
* **Create account on the website.**
* **Upload mockup**
* **Apply ML Algorithm**
* **Generate Code**
* **Place Code into File**

# SYSTEM ANALYSIS

##### Iterative Waterfall

The iterative waterfall model is an approach that combines elements of both the traditional waterfall model and iterative development methodologies. In this model, the development process is divided into sequential phases, similar to the waterfall model, but with the flexibility to revisit and iterate on previous stages. Each iteration follows a linear flow, starting from requirements gathering, analysis, design, implementation, testing, and deployment. However, after each iteration, feedback is collected, and adjustments are made based on lessons learned and evolving requirements. This iterative approach allows for incremental development, enabling early user feedback, and the ability to refine and enhance the product over multiple iterations. By incorporating iterative cycles within a linear framework, the iterative waterfall model strikes a balance between structure and flexibility, making it suitable for projects that benefit from periodic evaluation and adaptation.

Performance, and quality of the system. The iterative waterfall model consists of the following phases:

* Requirements Gathering: Gathering and documenting the project requirements, objectives, and constraints.
* System Design: Creating a high-level system design based on the gathered requirements.
* Implementation: Developing and coding the system components based on the design specifications.
* Testing: Conducting testing activities to verify the functionality,
* Deployment: Deploying the system to the production environment for end-users to access and utilize.
* Evaluation and Feedback: Collecting feedback from users and stakeholders to assess the system's performance and identify areas for improvement.

# FEASIBILITY STUDY

All projects are feasible given unlimited resources and infinite time. Unfortunately, the development of computer-based system in many cases is more likely to be plagued by scarcity of resources and delivery date. Hence, we have made use the concept of reusability that is what Object-Oriented Programming (OOP) is all about.

The feasibility report of the project holds the advantages and flexibility of the project. This is divided into three sections: -

1. Operational Feasibility
2. Technical Feasibility
3. Economic Feasibility

## Operational Feasibility:

It determines how acceptable the software is within the organization. The evaluations must then determine the general attitude and skills. Such restriction of the job will be acceptable. To the users are enough to run the proposed budget, hence the system is supposed to the feasible regarding all except of feasibility. In operational Feasibility, we attempt to ensure that every user can access the system easily.

Operational feasibility of the project also exists because in today’s world most of the people are using the internet and are purchasing the products online. There is nothing complex in the system that cannot be used by people. It is socially accessible feasible as well because of its usefulness and easiness in getting information. Time feasibility also exists because it can be developed and implemented in the given time. As far as legal feasibility is concerned there is no such restriction faced by the system.

## Technical Feasibility:

Technical feasibility centres on the existing computer system (Hardware and Software etc.) and to what extend it support the proposed addition. In this project, all the necessary cautions have been taken care to make it technically feasible. Using a key, the display of text/object is very fast. Also, the tools, operating system and programming language used in this localization process is compatible with the existing one. The technical needs of the system vary considerably but might include: -

* 1. The facility to produce outputs in a given time.
  2. Response time under certain conditions.

The project is technical feasible because of the availability of the required software hardware and technology. The changes can be made be made in the system as and when required.

## Economic Feasibility:

Economic analysis is the most frequently used method for evaluating the effectiveness of the candidate system. More commonly known as cost/benefit analysis, the procedure is to be determining the benefits and savings that are expected from a candidate and compare them with costs. If benefits outweigh costs, then the decision is made to design and implement the system. A systems financial benefit must exceed the cost of developing that system. i.e., a new system being developed should be a good investment for the organization.

Economic feasibility considers the following:

* 1. The cost to conduct a full system investigation.
  2. The cost of hardware and software for the class of application.
  3. The benefits in the form of reduced cost or fewer costly errors.

# DATABASE DESIGN

Database is critical for all businesses. A good database does not allow any form of anomalies and stores only relevant information in an ordered manner. If a database has anomalies, it is affecting the efficiency and data integrity. For example, delete anomaly arise upon the deletion of a row which also forces other useful data to be lost. As such, the tables need to be normalized. This fulfils the last objective of ensuring data are accurate and retrieved correctly. Database files are the key source of information into the system. It is the process of designing database files, which are the key source of information to the system. The files should be properly designed and planned for collection, accumulation, editing and retrieving the required information. The organization of data in database aims to achieve three major objectives: -

* + - Data integration
    - Data integrity
    - Data independence

# Use Case Diagram

A use case diagram is a graphical representation that depicts the interactions between users (actors) and a system in terms of specific actions or use cases. It is a high-level overview of the functionalities provided by a system and the roles played by different actors in utilizing those functionalities. Use case diagrams are commonly used in software development to capture and communicate the system's behaviour and requirements.

In brief, the purposes of use case diagrams can be said to be as follows −

* Used to gather the requirements of a system.
* Used to get an outside view of a system.
* Identify the external and internal factors influencing the system.
* Show the interaction among the requirements are actors.

In a use case diagram, the following elements are typically present:

1. **Actors:** These represent the users or external systems that interact with the system. Actors are depicted as stick figures. They can be individuals, organizations, or even other software systems.
2. **Use Cases:** These represent specific actions or functionalities provided by the system. Use cases describe the interactions between actors and the system to achieve a particular goal or perform a specific task. Use cases are depicted as ovals.
3. **Relationships:** Relationships show the associations and dependencies between actors and use cases. The primary relationship is the "association" between an actor and a use case, indicating that the actor is involved in that particular use case.
4. **System Boundary:** This is a box that encapsulates the use cases and actors, representing the boundaries of the system being modelled. It defines what is included within the scope of the system and what is outside of it.

## Use Case Diagram:

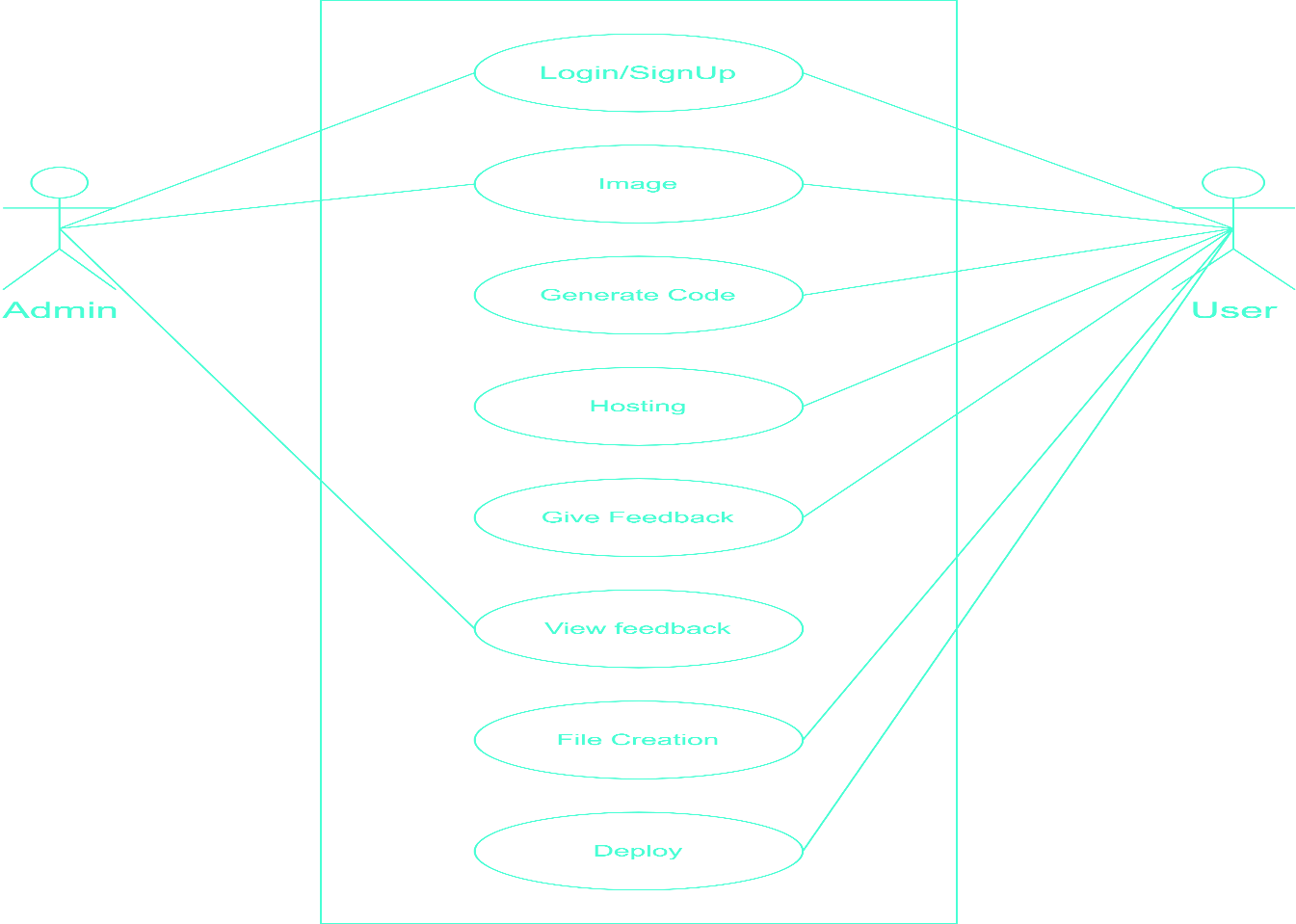


Figure -1

# Class Diagram

A class diagram is a type of static structure diagram in the Unified Modelling Language (UML) that represents the structure and relationships of the classes in a system. It provides a visual representation of the classes, their attributes, methods, and associations.

Purpose of Class Diagrams: -

The purpose of class diagram is to model the static view of an application. Class diagrams are the only diagrams which can be directly mapped with object-oriented languages and thus widely used at the time of construction.

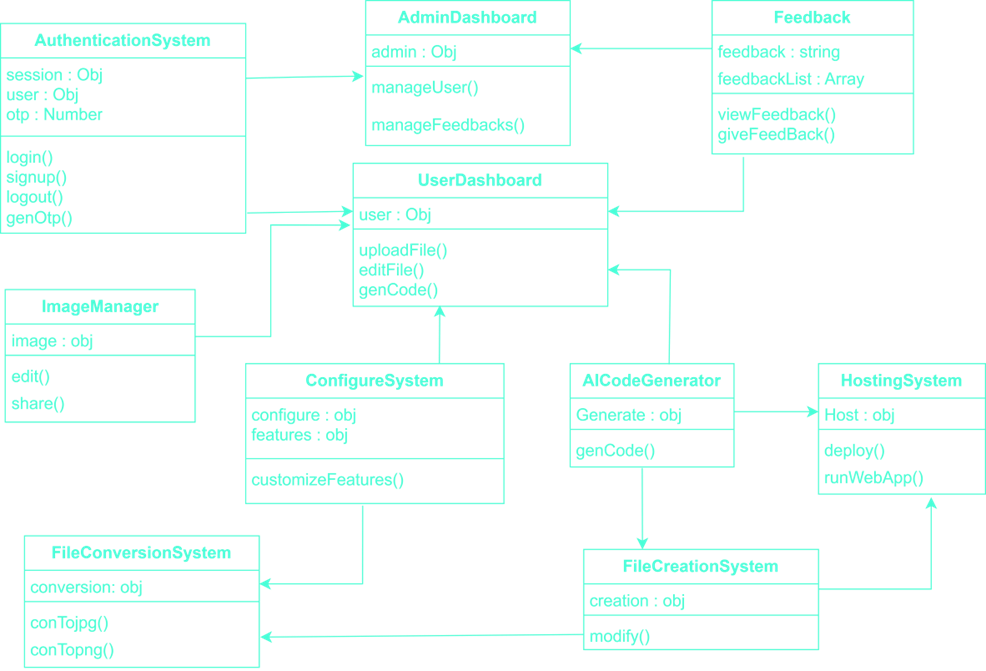
UML diagrams like activity diagram, sequence diagram can only give the sequence flow of the application, however class diagram is a bit different. It is the most popular UML diagram in the coder community.

* + Analysis and design of the static view of an application.
  + Describe responsibilities of a system.
  + Base for component and deployment diagrams.

In a class diagram, the following elements are typically present:

1. **Class:**A class represents a blueprint or template for creating objects. It defines the properties (attributes) and behaviours (methods) that objects of that class possess. A class is depicted as a rectangle with three compartments: the top compartment contains the class name, the middle compartment lists the attributes, and the bottom compartment lists the methods.
2. **Association:**An association represents a relationship between two or more classes, indicating that instances of those classes are connected or interact with each other in some way. Associations can be one-to-one, one-to-many, or many-to-many. They are depicted by lines connecting the classes, with optional arrows indicating the directionality of the relationship.
3. **Inheritance/Generalization:**Inheritance represents an "is-a" relationship between classes, where one class (the child or subclass) inherits the attributes and behaviours of another class (the parent or superclass). It is depicted as an arrow with an open triangle pointing from the child class to the parent class.
4. **Aggregation/Composition:**Aggregation and composition represent a "whole-part" relationship between classes. Aggregation implies a weaker association, where one class is composed of or contains instances of another class, but the existence of one class does not depend on the other. Composition implies a stronger association, where one class is composed of or contains instances of another class, and the existence of the whole depends on the existence of the parts. Aggregation is depicted as a line with an empty diamond at the whole class end, while composition is depicted with a filled diamond.
5. **Multiplicity:**Multiplicity specifies the number of instances that participate in a relationship. It is indicated by numbers or ranges near the ends of the association lines, representing how many instances can be associated.

## Class Diagram:



**Figure - 2**

**MODULE DESCRIPTION**

Module 1. Authentication System : An authentication system is a process or mechanism used to verify the identity of a user or system attempting to access a resource or service

Module 2. Image Manager : An image manager is a application or web service that allows users to organize, edit, store, and share digital images.

Module 3. AI Code Generator : An AI code generator is a tool that uses artificial intelligence techniques to automatically generate code for software applications.

Module 4. File creation system : A file creation system is a tool that enables users to create, modify, and manage files on a computer or other electronic device.

Module 5. Hosting System : A hosting system is a infrastructure that enables users to deploy and run web applications, websites, or other online services on remote servers or cloud-based platforms

Module 6. Admin Dashboard : An admin dashboard is a interface that provides a centralized view of the key performance indicators, metrics, and data related to an organization's operations, resources, and activities.

Module 7. User Dashboard : A user dashboard is a interface that provides a personalized view of the data, features, and functionality available to individual users of a web application or service

Module 8. Feedback System : A feedback system is a tool that enables users to provide feedback, comments, and suggestions about a product, service, or experience, and for organizations to collect, analyze, and act on this feedback.

Module 9. Configuration system : A configuration system is a tool that enables users to configure and customize the settings, features, and functionality of a software application, system, or device

Module 10. File conversion System : A file conversion system is a tool that enables users to convert files from one format to another.

# GANTT CHART

A Gantt chart is popular type of chart that illustrates a project schedule. Gantt Chart illustrates the start and finish dates of the terminal elements and summary elements of a project.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Task** | **24jan-25Feb** | **26Feb-7Mar** | **8Mar-31Mar** | **1Apr-30Apr** | **1May-6May** | **7May-11May** |
| **Develop project proposal** | 31 days |  |  |  |  |  |
| **Analysis** |  | 10 days |  |  |  |  |
| **Designing** |  |  | 24 days |  |  |  |
| **Coding** |  |  |  | 30 days |  |  |
| **Unit Testing** |  |  |  |  | 6 days |  |
| **Implementation** |  |  |  |  |  | 5 days |

# SOFTWARE REQUIREMENT SPECIFICATION (SRS) SOFTWARE REQUIREMENT

## Client Side:

* Web Browser (Google Chrome, Firefox, IE9 or above)

## Server Side:

* Web Browser (Google Chrome, Firefox)
* Node JS 14 or above
* Vs code
* Frontend Framework – React 17
* Backend Framework – Express JS
* Database – MongoDB

# HARDWARE REQUIREMENTS

## CLIENT SIDE:

|  |  |
| --- | --- |
| **Processor** | **Intel i3, AMD Ryzen 5 or above** |
| **RAM** | 4 GB or above |
| **Image Input** | Webcam |
| **Display** | 720p |
| **Others** | Internet Connection |

**DEVELOPER SIDE:**

|  |  |
| --- | --- |
| **Processor** | **Intel i5, AMD Ryzen 5 or above** |
| **RAM** | 8 GB or above |
| **Image Input** | Webcam |
| **Display** | 1080p |
| **Others** | Keyboard, mouse, Internet Connection |

# FUNCTIONAL REQUIREMENTS

**User Authentication**: Users should be able to create accounts, log in, and log out of the application. This functionality ensures secure access to the conversion tool and allows users to save their progress or access previously converted mockups.

**Mockup Upload:** Users should be able to upload mockup files in various formats, such as PSD (Photoshop), Sketch, or JPEG/PNG images. The system should support the conversion of these files into HTML, CSS, and JavaScript code.

**Conversion Options:** Provide users with options to customize the conversion process. For example, they may want to specify the desired layout, responsiveness, font styles, or other design aspects during the conversion.

**Conversion Output:** The system should generate well-structured HTML, CSS, and JavaScript code based on the uploaded mockup. The code should accurately represent the visual design and should be optimized for performance and compatibility across different browsers.

**Preview Functionality:** Users should be able to preview the converted web pages to ensure the accuracy of the conversion.

**Download Output:** Provide users with the ability to download the converted code as a package or individual files. This allows users to easily access and integrate the generated code into their web development projects.

**Error Handling:** The system should handle errors gracefully and provide meaningful error messages to users when issues occur during the conversion process. This helps users troubleshoot problems and take appropriate actions.

**Version Control:** If the project supports multiple versions of the converted code, implement a version control mechanism to track changes, roll back to previous versions, and manage different iterations of the converted code.

**Documentation and Help:** Include documentation or a user guide that explains how to use the conversion tool effectively. Provide instructions, tips, and examples to assist users in understanding the features and functionalities of the tool.

**Performance Optimization:** Ensure that the conversion process is efficient and does not excessively consume system resources. Optimize the code generation algorithm to produce clean and optimized code that loads quickly and performs well

**NON-FUNCTIONAL REQUIREMENTS**

**Performance:** The system should be responsive and provide fast conversion times, even for large or complex mockup files. It should handle multiple user requests simultaneously without significant degradation in performance.

**Scalability:** The application should be designed to handle a growing number of users and increasing data volume. It should be able to scale horizontally or vertically to accommodate increased demand without compromising performance.

**Usability:** The user interface should be intuitive, user-friendly, and easy to navigate. It should require minimal training or technical expertise for users to operate the conversion tool effectively.

**Compatibility:** The converted code should be compatible with a wide range of web browsers, including popular ones like Chrome, Firefox, Safari, and Edge. It should also support different operating systems, such as Windows, macOS, and Linux.

**Security:** The system should implement appropriate security measures to protect user data and prevent unauthorized access. This includes secure user authentication, encryption of sensitive information, and protection against common web vulnerabilities (e.g., cross-site scripting, SQL injection).

**Reliability:** The application should be reliable and available for use whenever needed. It should have minimal downtime, and any scheduled maintenance or updates should be communicated to users in advance.

**Accessibility**: The conversion tool should adhere to accessibility standards (e.g., WCAG 2.0) to ensure that it can be used by individuals with disabilities. This includes providing alternative text for images, using appropriate color contrast, and offering keyboard accessibility.

**Maintainability:** The code should be well-organized, modular, and written with best practices to facilitate future maintenance and updates. It should be easy for other developers to understand and modify the codebase.

**Extensibility:** The system should be designed in a way that allows for easy integration with other tools or services. This may include providing APIs or hooks for extending the functionality or integrating with popular web development frameworks.

**Compliance:** The application should comply with relevant legal and industry standards, such as data protection regulations (e.g., GDPR) or accessibility guidelines (e.g., ADA compliance).

# DESIGN & IMPLEMENTATION CONSTRAINTS

**Technology Stack:** Your project may have specific requirements or constraints regarding the technology stack to be used. For example, you might need to use a particular programming language, framework, or content management system (CMS) due to compatibility or organizational guidelines.

**Integration Requirements:** If your project needs to integrate with other systems, such as databases, third-party APIs, or payment gateways, you may have constraints related to compatibility, security protocols, or specific data formats.

**Performance Constraints:** There may be performance requirements or constraints that need to be considered during the design and implementation. For example, the project may need to handle a certain number of concurrent users or maintain a specific response time for critical operations.

**Platform Compatibility:** If the web application is intended to run on multiple platforms or devices, such as desktops, smartphones, or tablets, you may need to ensure compatibility and responsiveness across different screen sizes and operating systems.

**Legal and Regulatory Compliance:** Depending on the nature of your project, there may be legal or regulatory constraints that need to be addressed. For instance, you may need to comply with data protection regulations, privacy laws, or industry-specific standards.

**Time Constraints:** Your project may have time limitations or deadlines that must be considered during the design and implementation phases. This could be driven by business requirements, project timelines, or external factors.

**Budget Constraints:** The availability of resources, including budgetary constraints, may impact the design and implementation decisions. You may need to optimize the project within a certain budget or work within limited resources.

**Security Requirements:** Security constraints may impose specific design considerations, such as encryption, secure user authentication, protection against common vulnerabilities, and adherence to security best practices.

**Accessibility Guidelines:** If your project needs to comply with accessibility guidelines, such as Web Content Accessibility Guidelines (WCAG), you'll need to consider design and implementation constraints related to providing accessible features for users with disabilities.

**Organizational Standards:** Your project may need to align with organizational standards, coding conventions, or architectural principles

# TOOLS & TECHNOLOGY

## Web Browser -

A web browser (commonly referred to as a browser or internet browser) is an application software for accessing the World Wide Web. When a user requests a web page from a particular website, the web browser retrieves the necessary content from a web server and then displays the page on the user's device.

## HTML -

The Hypertext Markup Language, or HTML is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.

Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated <img /> and <input /> <p> by tags, written using angle brackets. Tags such as directly introduce content into the page. Other tags such as surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

HTML can embed programs written in a scripting language such as JavaScript, which affects the behaviour and content of web pages. Inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), former maintainer of the HTML and current maintainer of the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.

## Cascading Style Sheets (CSS) -

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file which reduces complexity and repetition in the structural content as well as enabling the .css file to be cached to improve the page load speed between the pages that share the file and its formatting.

## JavaScript -

JavaScript often abbreviated as JS, is a programming language that conforms to the ECMAScript specification. JavaScript is high-level, often just-in-time compiled, and multiparadigm. It has curly- bracket syntax, dynamic typing, prototype-based object-orientation, and first-class functions.

Alongside HTML and CSS, JavaScript is one of the core technologies of the World Wide Web.

Over 97% of websites use it client-side for web page behaviour, often incorporating third-party libraries. All major web browsers have a dedicated JavaScript engine to execute the code on the user's device.

As a multi-paradigm language, JavaScript supports 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 they are now core components of other software systems, most notably servers and a variety of applications.

## Node JS 14 -

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,[6] unifying web application development around a single programming language, rather than different languages for server- side and client-side scripts.

## VS code -

Visual Studio Code is a source-code editor made by Microsoft for Windows, Linux and macOS.[9] 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.

Microsoft has released most of Visual Studio Code's source code on the Microsoft/vscode repository of GitHub using the "Code – OSS" name, under the permissive MIT License, while the releases by Microsoft are proprietary freeware

## React JS -

React (also known as React.js or ReactJS) is an open-source front-end JavaScript library for building user interfaces or UI components. It is maintained by Facebook and a community of individual developers and companies. React can be used as a base in the development of single- page or mobile applications. However, React is only concerned with state management and rendering that state to the DOM, so creating React applications usually requires the use of additional libraries for routing, as well as certain client-side functionality

### Express JS -

Express.js, or simply Express, is a back-end web application framework for Node.js, released as free and open-source software under the MIT License. It is designed for building web applications and APIs. It has been called the de facto standard server framework for Node.js. The original author, TJ Holowaychuk, described it as a Sinatra-inspired server,[5] meaning that it is relatively minimal with many features available as plugins. Express is the back-end component of popular development stacks like the MEAN, MERN or MEVN stack, together with the MongoDB database software and a JavaScript front-end framework or library

## MongoDB -

MongoDB is a source-available cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas. MongoDB is developed by MongoDB Inc. and licensed under the Server-Side Public License (SSPL).

# Cost estimation and cost constructive model

The proposed COCOMO (Constructive Cost Estimation Model) COCOMO is one of the most generally used software estimation models in the world. COCOMO predicts the efforts and schedule of a software product based on the size of the software.

The key parameters which define the quality of any software products, which are also an outcome of the CoCoMo, are primarily Effort & Schedule:

* + Effort: Amount of labor that will be required to complete a task. It is measured in person- months units.
  + Schedule: Simply means the amount of time required for the completion of the job, which is, of course, proportional to the effort put. It is measured in the units of time such as weeks, months.

The initial estimate (also called nominal estimate) is determined by an equation of the form used in the static single variable models, using KDLOC as the measure of the size. To determine the initial effort Ei in person-months the equation used is of the type is shown below

**Ei=a\*(KDLOC)b**

Where The value of the constant a and b are depends on the project type

**Basic COCOMO Model:**The basic COCOMO model provides an accurate size of the project parameters. The following expressions give the basic COCOMO estimation model:

Effort=a1\*(KLOC)a2 PM Tdev=b1\*(efforts)b2 Months Where-

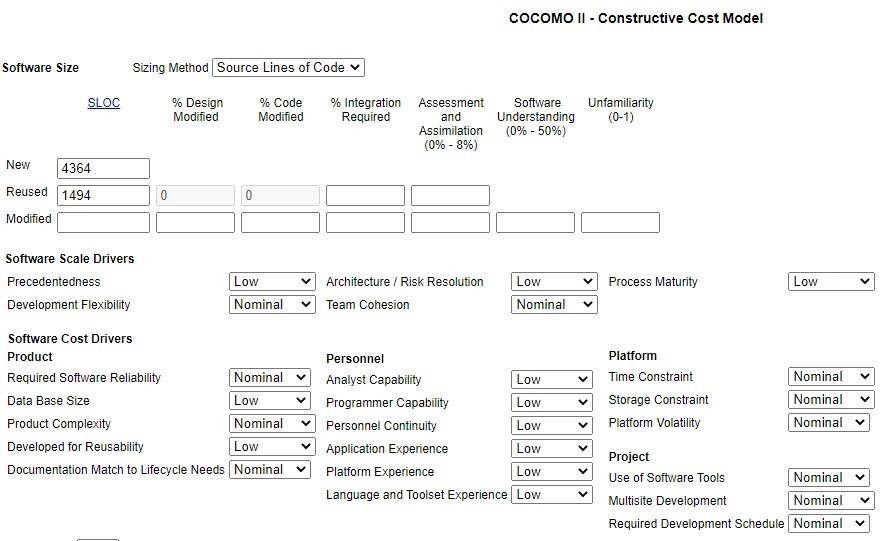
**KLOC** is the estimated size of the software product indicate in Kilo Lines of Code,

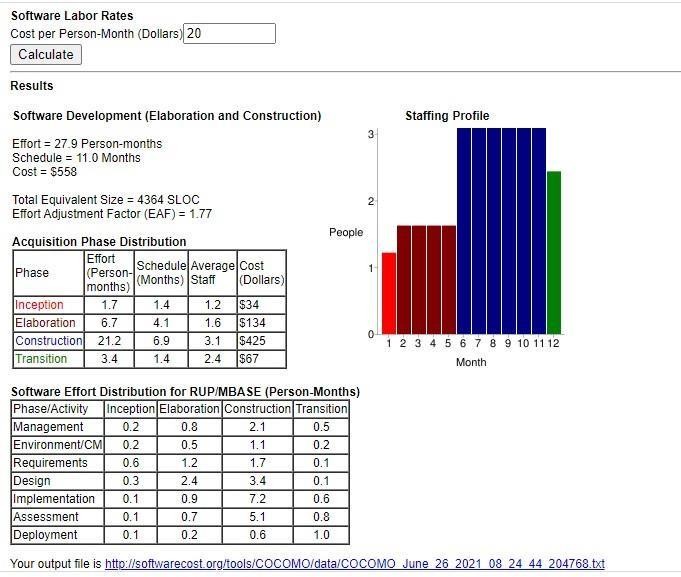
**a1, a2, b1, b2** are constants for each group of software products,

**Tdev** is the estimated time to develop the software, expressed in months,

**Effort** is the total effort required to develop the software product, expressed in person months (PMs).

Thus, in our project the following considerations is taken so as to get the cost estimation of this project.





**TESTING**

## Testing Phase:

One of the purposes of the testing is to validate and verify the system. Verification means checking the system to ensure that it is doing what the function is supposed to do and Validation means checking to ensure that system is doing what the user wants it to do. No program or system design is perfect; communication between the user and the designer is not always complete or clear, and time is usually short. The result is errors and more errors. Theoretically, a newly design Lal system should have all the pieces in working order, but in reality, each piece works independently. Now is the time to put all the pieces into one system and test it to determine whether it meets the user's requirements. This is the best chance to detect and correct errors before the system is implemented. The purpose of system testing is to consider all the likely variations to which it will be subjected and then push the system to its limits. If we implement the system without proper testing then it might cause the problems.

1. Communication between the user and the designer.
2. The programmer's ability to generate a code that reflects exactly the system specification.
3. The time frame for the design.

Theoretically, a new designed system should have all the pieces in working order, but in reality, each piece works independently. Now is the time to put all the pieces into one system and test it to determine whether it meets the requirements of the user. The process of system testing and the steps taken to validate and prepare a system for final implementation are:

# LEVELS OF TESTING-

The different types of testing are as follow:

1.Unit Testing

2.Integration Testing

3.Acceptance testing

## Unit Testing:

This is the smallest testable unit of a computer system and is normally tested using the white box testing. The author of the programs usually carries out unit tests.

## Integration Testing:

In integration testing, the different units of the system are integrated together to form the complete system and this type of testing checks the system as whole to ensure that it is doing what is supposed to do. The testing of an integrated system can be carried out top- down, bottom-up, or big-bang. In this type of testing, some parts will be tested with white box testing and some with black hox testing techniques. This type of testing plays very important role in increasing the systems productivity. We have checked our system by using the integration testing techniques.

## Acceptance Testing:

The user to ensure that the system functions, as the user actually wanted performs this testing. With prototyping techniques, this stage becomes very much a formality to check the accuracy and completeness of processing- The screen layouts and output should already have been tested during the prototyping phase. An error in the program code can remain undetected indefinitely. To prevent this from happening the code was tested at various levels. To successfully test a system, each condition, and combinations of conditions had to be tested. Each program was tested and linked to other programs. This unit of program is tested and linked to other units and so on until the complete system has been tested. The purpose of testing is to ensure that catch program is fully tested. To do so a test plan had to be created. The test plan consists of a number of tests runs such as the valid paths through the code, and the exception and error handling paths. For each lest run there is list of conditions tested, the test data used and the result expected. The test plan was then. reviewed to check that cache path through the code is tested correctly. It is the responsibility of the programmer to collect the data that will produce the required test condition.

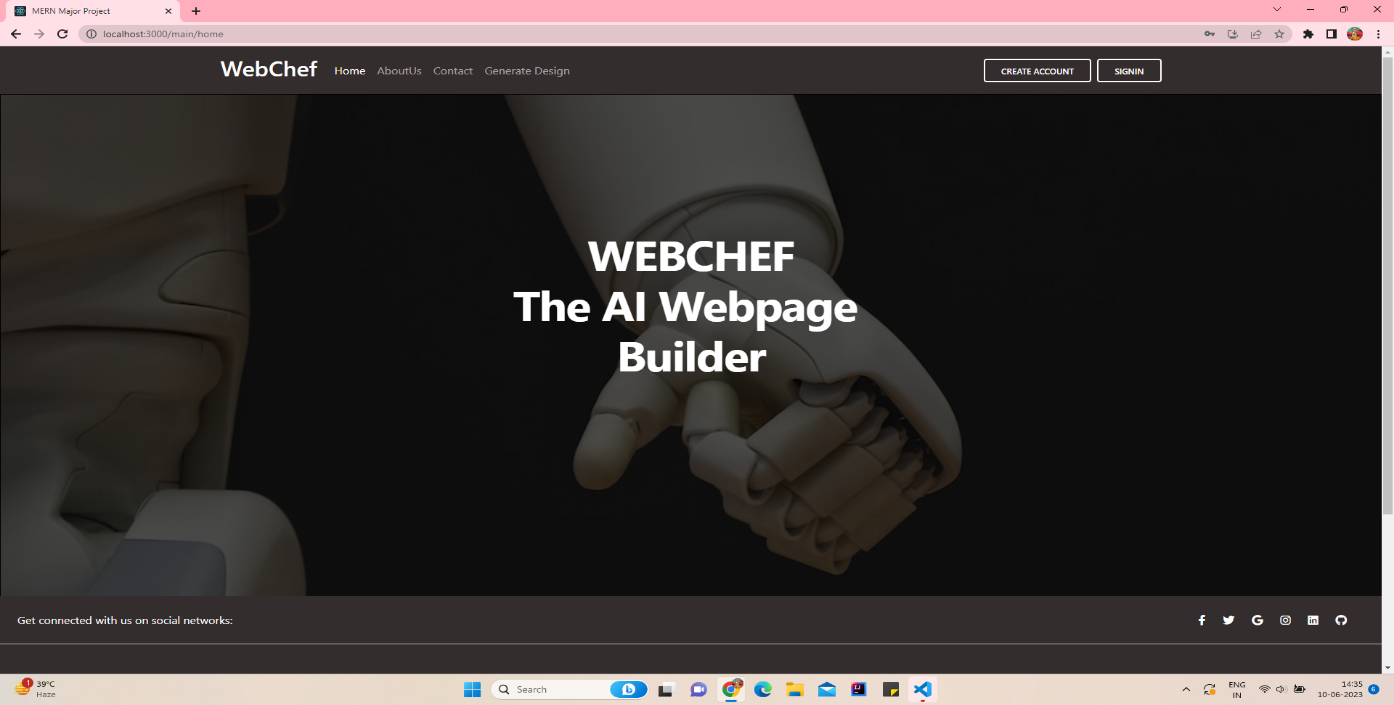
# VERIFICATION AND VALIDATION (V&V)

The objectives of verification, validity nativities are to assess and improve the quality of the work products generated during development and modification of the Soll ware. Quality depends upon the various attributes like correctness, completeness, consistency, reliability, usefulness, usability, efficiency and conformance standards. The terms verification and validation are used synonymously. These are defined a pun Verification: "Are we building the product, right?" Validation: "Are we building the right product?" Verification activities include proving. testing, and reviews. Validation is the process of evaluating software at the end of the software development to ensure compliance with the software requirements. Testing is a common method of validation. Clearly, for high reliability we need to perform both activities. Together, they are often called V&V activities. The major V&V activities for software development are inspection, reviews, and testing (both static and dynamic). The V&V plan identifies the different V&V tasks for the different phases and specifies how these tasks contribute to the project V&V goals. The methods to be used for performing these V&V activities, the responsibilities and milestones for each of these activities, inputs and outputs for cache V&V task, and criteria for evaluating the outputs are also specified.

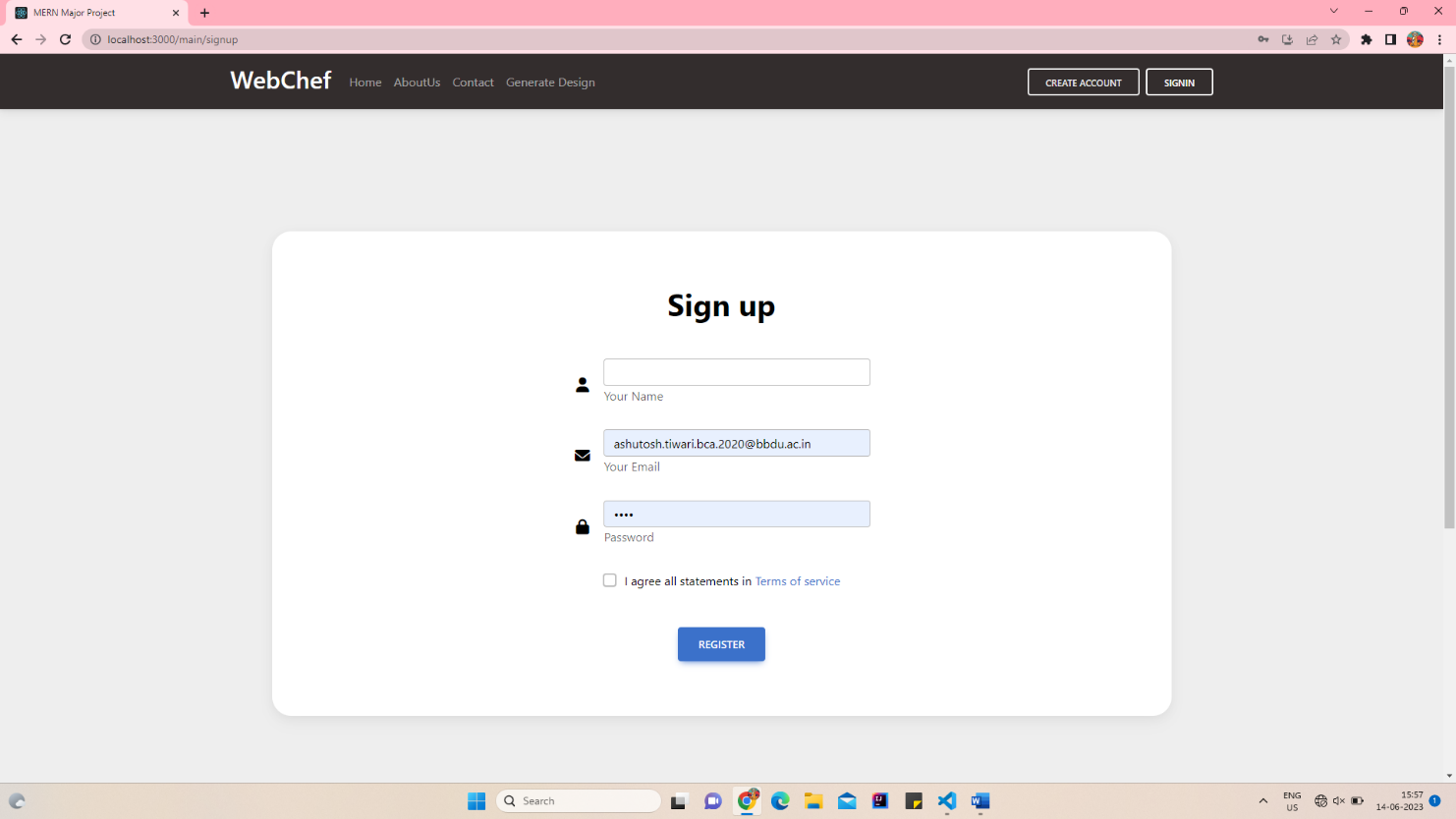
The two major V&V approaches are testing and inspections. Testing is an activity that can be generally performed only on code. It is an important activity and is discussed in detail in a later chapter. Inspection is a more general activity that can be applied to any work product, including code. Many of the V&V tasks are such that for them, an inspection type of activity is the only possible way to perform the tasks (e.g., trace ability and document evaluation). Due to this, inspections play a significant role in verification.

**Screen Shot**

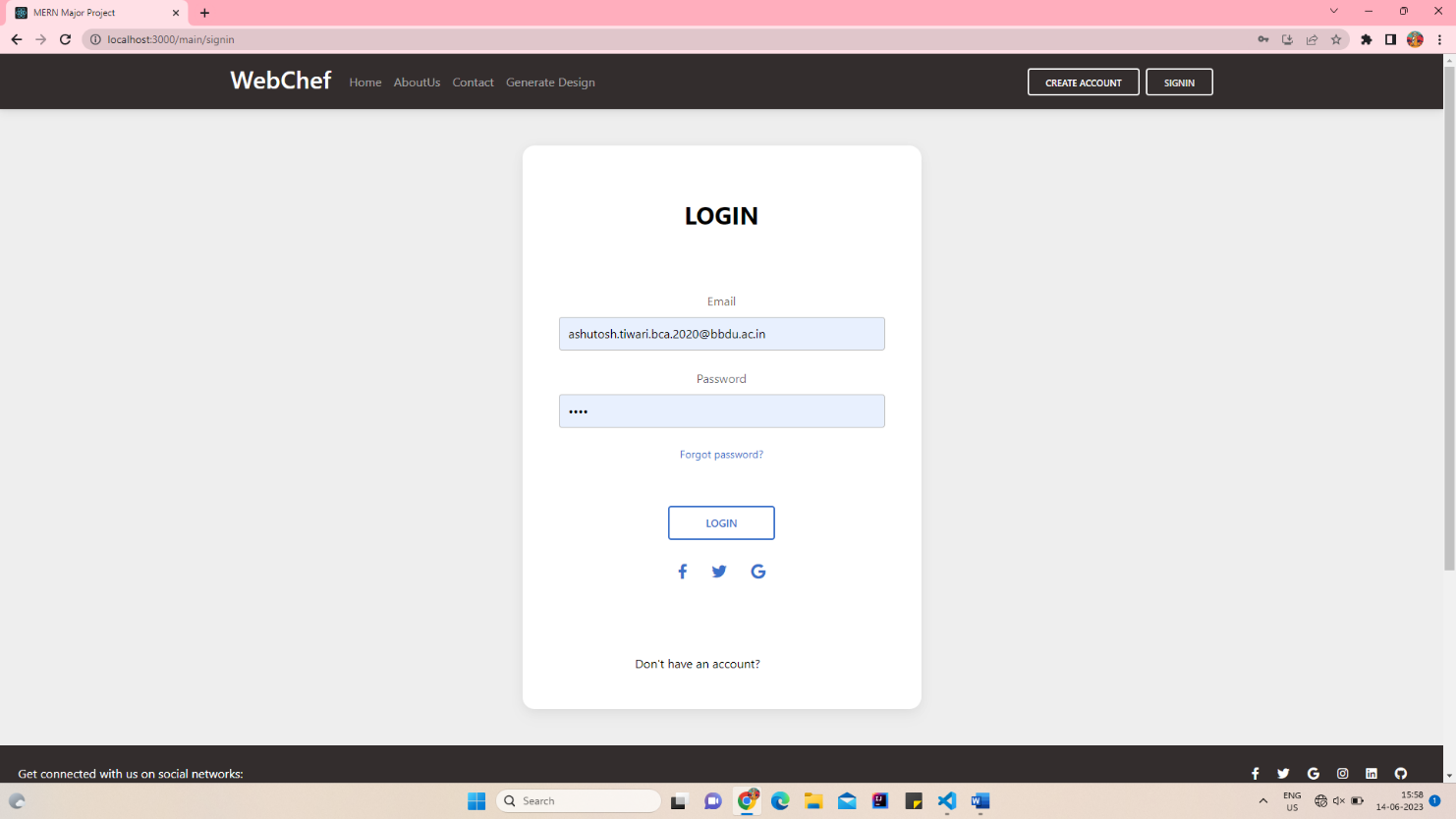
## Home Page –

******

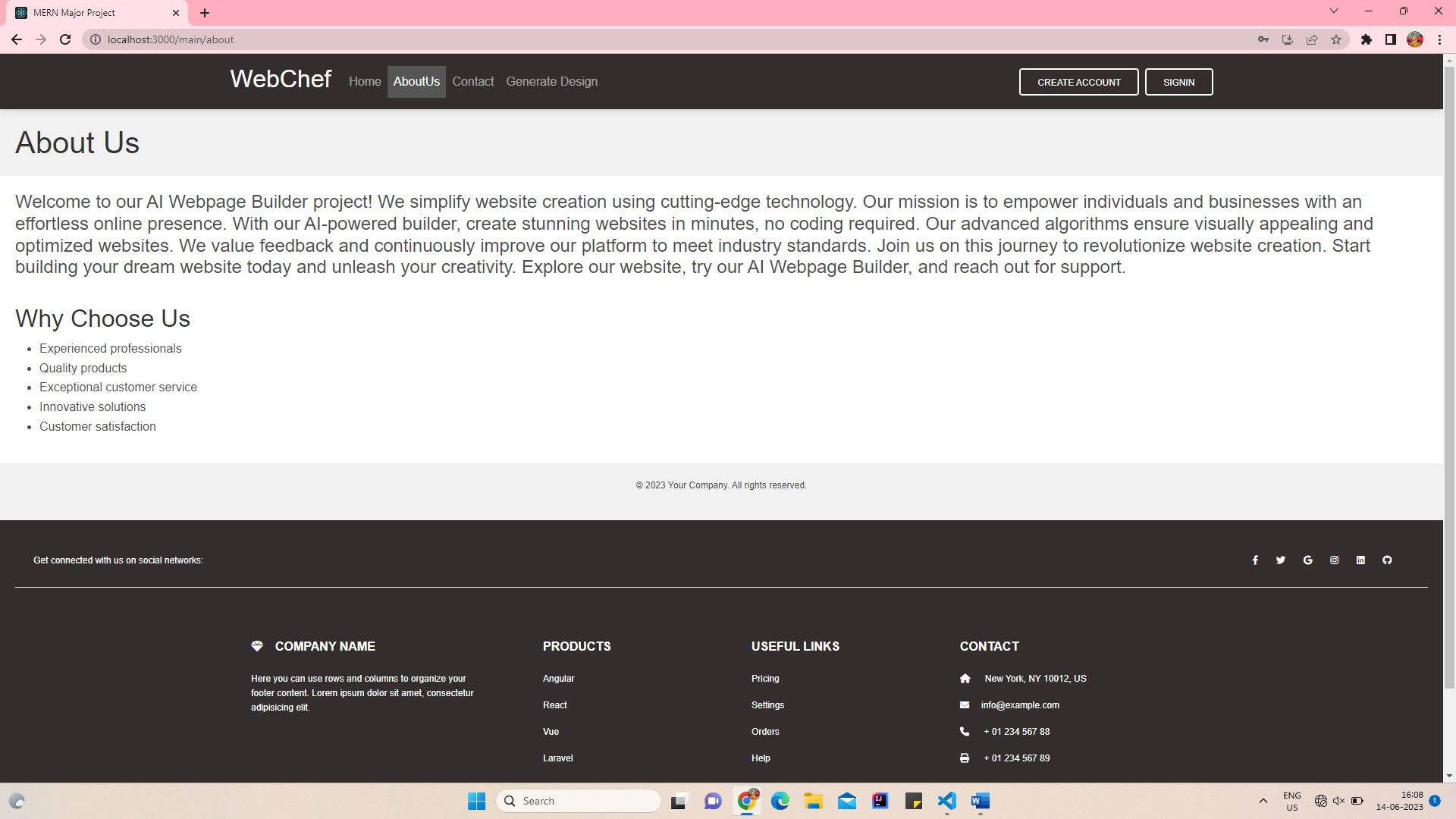
**Sign Up Page –**



## Login Page –



## About Us –



## Contact Us –

## 

## Manage Mockups –

# 

## 

## 

## Upload Mockup –

# 

## Generated Code –

# 

# CONCLUSION

An AI code generator for web design from mockups is an innovative technology that can help businesses automate their web development processes and improve the overall quality of their web applications. With an AI code generator, web developers can quickly generate code and iterate designs faster, resulting in faster development cycles, cost-effectiveness, improved accuracy, scalability, and improved user experience.

.

# BIBLIOGRAPHY

### Web References: -

* + https[://w](http://www.google.com/)ww[.google.com](http://www.google.com/)
  + [**https://www.geeksforgeeks.org/**](https://www.geeksforgeeks.org/)
  + [**https://www.tutorialspoint.com/index.htm**](https://www.tutorialspoint.com/index.htm)
  + https://en.wikipedia.org/wiki/Main\_Page