**House Price Prediction Using Machine Learning**

**AN INTERNSHIP REPORT**

*Submitted by*

**SION RANI KONDEPUDI [EA2331201010063]**

**Under the Guidance of** **Dr. M. Muthuselvam**

(Assistant Professor, Directorate of Online Education)

*in partial fulfillment for the award of the degree of*

**BACHELOR OF COMPUTER APPLICATIONS**



DIRECTORATE OF ONLINE EDUCATION SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR- 603 203

DECEMBER 2024

DIRECTORATE OF ONLINE EDUCATION

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY KATTANKULATHUR – 603 203

BONAFIDE CERTIFICATE

This internship report titled **“Online Movies Job Recruitment Management System”** is the Bonafede work of **“Sion Rani Kondepudi [EA2331201010****063]”**, who carried out the internship work under my supervision along with the company mentor. Certified further, that to the best of my knowledge the work reported herein does not form any other internship report or dissertation based on which a degree or award was conferred on an earlier occasion on this or any other candidate

# INTERNSHIP OFFER LETTER



**ACKNOWLEDGEMENTS**

I express my humble gratitude to **Dr C. Muthamizhchelvan**, Vice-Chancellor, SRM Institute of Science and Technology, for the facilities extended for the project work and his continued support. I extend my sincere thanks to Director DOE, SRM Institute of Science and Technology, Prof. **Dr Manoranjan Pon Ram,** for his invaluable support. SRM Institute of Science and Technology, for her support throughout the project work. I want to convey my thanks to Programme Coordinator **Dr. M. Muthu Selvam**, Directorate of online Education, SRM Institute of Science and Technology, for their inputs during the project reviews and support. Our inexpressible respect and thanks to my guide, **Dr. M. Muthu Selvam,** Assistant Professor & Programme Coordinator Directorate of online Education, SRM Institute of Science and Technology, for providing me with an opportunity to pursue my project under her mentorship. He provided me with the freedom and support to explore the research topics of my interest. His passion for solving problems and making a difference in the world has always been inspiring. I sincerely thank the Directorate of online Education, staff and students, SRM Institute of Science and Technology, for their help during our project. Finally, I would like to thank parents, family members, and friends for their unconditional love, constant support, and encouragement.

**Sion Rani Kondepudi**

# TABLE OF CONTENTS

CHAPTER NO. TITLE PAGE NO.

1. [Abstract 5](#_TOC_250011)
2. [System Analysis 6](#_TOC_250010)
3. [Software requirements specifications 7](#_TOC_250009)
4. [Technology 8](#_TOC_250008)
   1. [JAVASCRIPT 8](#_TOC_250007)
   2. [HTML 9](#_TOC_250006)
   3. [Overview of CSS 11](#_TOC_250005)
   4. [Overview of PHP 13](#_TOC_250005)
   5. [Overview of an application using HTML, CSS and JavaScript 14](#_TOC_250004)
5. [Coding… 16](#_TOC_250003)
6. [Result 26](#_TOC_250002)
7. [Conclusion 32](#_TOC_250001)
8. [References 33](#_TOC_250000)

# ABSTRACT

House price prediction is a critical task in the real estate industry, as it helps homebuyers, sellers, and real estate agents make data-driven decisions. The goal of this project is to predict house prices based on key features such as location, size, and amenities using machine learning techniques. The project uses a dataset containing information on thousands of homes, such as their geographic location, the number of rooms, the population in the area, and the median income of households. These features are used as inputs to train machine learning models capable of predicting the median house price in each area.

In this project, we implemented two models: Gradient Boosting and Neural Networks. Both models were trained using a dataset that was carefully preprocessed to ensure accuracy. Data preprocessing involved handling missing values, normalizing numerical features, and encoding categorical variables. Feature engineering was also performed to create new, useful features, such as the bedroom-to-total-rooms ratio, which helped improve model performance.

Gradient Boosting is a powerful ensemble learning method that combines multiple weak models to create a strong model. It was chosen for this project because of its ability to handle complex datasets and its robustness in regression tasks. Neural Networks, on the other hand, are a type of deep learning model that mimics the human brain’s neural connections. The neural network model was designed with multiple layers, allowing it to learn complex patterns in the data.

The results of the project show that both models are effective in predicting house prices. The Gradient Boosting model achieved a higher R² score, indicating that it explained a greater proportion of the variance in house prices. The neural network model, while slightly less accurate, also performed well, especially for mid-range house prices. Error analysis revealed that both models struggled to predict extremely high or low prices, likely due to outliers in the data.

This project demonstrates the potential of machine learning in the real estate industry. By using data to predict house prices, real estate agents and homebuyers can make more informed decisions, leading to better pricing strategies and more efficient property transactions. The project also highlights the importance of data preprocessing and feature engineering in machine learning tasks, as these steps significantly improved model performance.

Future work could involve optimizing the models through hyperparameter tuning and exploring other machine learning algorithms, such as Random Forests or Support Vector Machines.

# SYSTEM ANALYSIS

## Requirement Analysis

* 1. **Existing System:**

## The existing methods of house price prediction are often manual or based on limited data. Traditionally, real estate agents rely on their experience and market knowledge to estimate house prices. This approach is subjective and can lead to inaccurate predictions, especially in dynamic markets where property values fluctuate based on numerous factors. Additionally, online tools available today often provide general estimates without considering all relevant factors like local amenities, population density, or recent trends in the housing market.

## In an existing system, predictions are typically based on historical sales data, which may not account for real-time changes in market conditions. Furthermore, these systems might not be flexible enough to handle new features, such as changes in infrastructure or emerging neighborhoods. As a result, there is a significant margin of error in existing house price predictions, which can lead to either overpricing or underpricing of properties.

## The manual approach is time-consuming and prone to human error. It also does not scale well when applied to large datasets, such as entire cities or regions. For real estate companies, this lack of scalability can lead to missed opportunities and suboptimal pricing strategies.

## Proposed System

The proposed system uses machine learning to predict house prices based on a variety of features that influence property values. By leveraging a large dataset containing information on thousands of homes, the system can learn the relationships between different features and house prices. This data-driven approach eliminates human bias and provides more accurate predictions by considering a wider range of factors.

The proposed system will use Gradient Boosting and Neural Networks, two of the most powerful machine learning algorithms for regression tasks. These models are capable of learning complex relationships in the data and can generalize well to new, unseen properties. By training the models on historical data, the system can make predictions for new properties based on their features, such as location, size, and nearby amenities.

The key advantage of the proposed system is its ability to handle large datasets and a wide range of features. It can easily incorporate new data, such as changes in local infrastructure or new housing developments, allowing it to adapt to changing market conditions. Additionally, the system can be integrated with real estate websites or apps, providing homebuyers and real estate agents with real-time price predictions.

# SOFTWARE REQUIREMENTS SPECIFICATIONS

## System configurations

The software requirement specification can produce at the culmination of the analysis task. The function and performance allocated to software as part of system engineering are refined by established a complete information description, a detailed functional description, a representation of system behavior, and indication of performance and design constrain, appropriate validate criteria, and other information pertinent to requirements.

## Software Requirements:

* Operating System: The project is platform-independent, and the code can run on Windows, macOS, or Linux. However, for the sake of consistency, we have used Windows 10 for the project development environment.
* Programming Language: Python is the primary language used for this project. Python’s extensive ecosystem of libraries for machine learning and data analysis, including Pandas, Scikit-learn, and TensorFlow, make it an ideal choice for this project. Its simplicity and readability also make it easier to write and maintain code.
* Libraries:
* Pandas: Used for data manipulation and analysis, Pandas allows us to handle large datasets efficiently. It provides functions for reading datasets from various formats (e.g., CSV, Excel), handling missing data, and performing data transformations.
* NumPy: Provides support for large multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays. NumPy is used primarily for numerical computations.
* Matplotlib and Seaborn: These libraries are used for data visualization. Matplotlib is a 2D plotting library, and Seaborn is a statistical data visualization library built on top of Matplotlib. These tools allow us to create various plots (scatter plots, histograms, box plots) to visualize the distribution and relationships of features.
* Scikit-learn: One of the most popular machine learning libraries for Python, Scikit-learn provides simple and efficient tools for data mining and data analysis. It includes numerous machine learning algorithms such as regression, classification, clustering, and more.
* TensorFlow: An open-source platform for machine learning, TensorFlow is used to build and train neural networks for predictive modeling. Its flexibility and scalability make it suitable for both research and production environments.
* XGBoost: A popular library for implementing Gradient Boosting, XGBoost is designed for speed and performance. It’s used in this project for the Gradient Boosting Regression model due to its ability to handle large datasets and deliver high accuracy.
* Integrated Development Environment (IDE): The project is developed using Jupyter Notebook, which is an open-source web application that allows you to create and share documents containing live code, equations, visualizations, and explanatory text. Jupyter Notebook is widely used in data science and provides an interactive environment for writing and testing Python code.
* Data Visualization Tools: Data visualization is a key part of understanding the dataset and evaluating model performance. We use tools like Matplotlib and Seaborn to create insightful charts and plots. These visualizations help in identifying trends and patterns in the data that may not be obvious from numerical analysis alone.

Expand each software requirement with detailed explanations of why the tool was chosen, its role in the project, and how it contributes to the overall workflow.

# TECHNOLOGY

## PYTHON

Python is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation.

Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly procedural), object-oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library.

Guido van Rossum began working on Python in the late 1980s as a successor to the ABC programming language and first released it in 1991 as Python 0.9.0. Python 2.0 was released in 2000. Python 3.0, released in 2008, was a major revision not completely backward-compatible with earlier versions. Python 2.7.18, released in 2020, was the last release of Python 2.

Python consistently ranks as one of the most popular programming languages, and has gained widespread use in the machine learning community.

Python is meant to be an easily readable language. Its formatting is visually uncluttered and often uses English keywords where other languages use punctuation. Unlike many other languages, it does not use curly brackets to delimit blocks, and semicolons after statements are allowed but rarely used. It has fewer syntactic exceptions and special cases than C or Pascal.

Python's large standard library provides tools suited to many tasks and is commonly cited as one of its greatest strengths. For Internet-facing applications, many standard formats and protocols such as MIME and HTTP are supported. It includes modules for creating graphical user interfaces, connecting to relational databases, generating pseudorandom numbers, arithmetic with arbitrary-precision decimals, manipulating regular expressions, and unit testing.

### Advantages of JavaScript

The merits of using JavaScript are −

* + Less server interaction − You can validate user input before sending the page off to the server. This saves server traffic, which means less load on your server.
  + Immediate feedback to the visitors − They don't have to wait for a page reload to see if they have forgotten to enter something.
  + Increased interactivity − You can create interfaces that react when the user hovers over them with a mouse or activates them via the keyboard.
  + Richer interfaces − You can use JavaScript to include such items as drag-and-drop components and sliders to give a Rich Interface to your site visitors.

### Limitations of JavaScript

We cannot treat JavaScript as a full-fledged programming language. It lacks the following important features-

* + Client-side JavaScript does not allow the reading or writing of files. This has been kept for security reason.
  + JavaScript cannot be used for networking applications because there is no such support available.
  + JavaScript doesn't have any multi-threading or multiprocessor capabilities.
  + Once again, JavaScript is a lightweight, interpreted programming language that allows you to build interactivity into otherwise static HTML pages.

### JavaScript Development Tools

* + One of major strengths of JavaScript is that it does not require expensive development tools. You can start with a simple text editor such as Notepad. Since it is an interpreted language inside the context of a web browser, you don't even need to buy a compiler.
  + To make our life simpler, various vendors have come up with very nice JavaScript editing tools. Some of them are listed here −
  + Microsoft FrontPage − Microsoft has developed a popular HTML editor called FrontPage. FrontPage also provides web developers with a number of JavaScript tools to assist in the creation of interactive websites.
  + Macromedia Dreamweaver MX − Macromedia Dreamweaver MX is a very popular HTML and JavaScript editor in the professional web development crowd. It provides several handy prebuilt JavaScript components, integrates well with databases, and conforms to new standards such as

XHTML and XML.

* + Macromedia HomeSite 5 − HomeSite 5 is a well-liked HTML and JavaScript editor from Macromedia that can be used to manage personal websites effectively.

## HTML

HTML stands for **Hypertext Markup Language**, and it is the most widely used language to write Web Pages.

Hypertext refers to the way in which Web pages (HTML documents) are linked together. Thus, the link available on a webpage is called **Hypertext**.

As its name suggests, HTML is a Markup Language which means you use HTML to simply "mark-up" a text document with tags that tell a Web browser how to structure it to display.

Originally, HTML was developed with the intent of defining the structure of documents like headings, paragraphs, lists, and so forth to facilitate the sharing of scientific information between researchers.

Now, HTML is being widely used to format web pages with the help of different tags available in HTML language.

### Advantages:

* + HTML helps to build structure of a website and is a widely used Markup language.
  + It is easy to learn.
  + Every browser supports HTML Language.
  + HTML is light weighted and fast to load.
  + Storage of big files are allowed because of the application cache feature.
  + Do not get to purchase any extra software because it’s by default in every window.
  + Loose syntax (although, being too flexible won’t suit standards).
  + HTML is simple to edit as being a plain text.
  + It integrates easily with other languages such as JavaScript, CSS etc.
  + HTML is that it is easy to code even for novice programmers.
  + HTML also allows the utilization of templates, which makes designing a webpage easy.
  + It is fast to download as the text is compressible.
  + Very useful for beginners in the web designing field.
  + HTML can be supported to each and every browser, if not supported to all the browsers.
  + HTML is built on almost every website, if not all websites.
  + HTML is increasingly used for data storage as like XML syntax.
  + HTML has many tag and attributes which can short your line of code.

### Disadvantages:

* + It cannot produce dynamic output alone, since it’s a static language.
  + Making the structure of HTML documents becomes tough to understand.
  + Errors can be costly.
  + It is the time consuming as the time it consume to maintain on the color scheme of a page and to make lists, tables and forms.
  + We need to write a lot of code for just creating a simple webpage.
  + We have to check up the deprecated tags and confirm not to use them to appear because another language that works with HTML has replaced the first work of the tag, and hence the opposite

language needs to be understood and learned.

* + Security features offered by HTML are limited.
  + If we need to write down long code for creating a webpage then it produces some complexity.
  + HTML can create only static and plain pages so if we would like dynamic pages then HTML isn’t useful.
  + Editing of web page need to be done separately, they are not centralized.

HTML5 is the next major revision of the HTML standard superseding HTML 4.01, XHTML 1.0, and XHTML 1.1. HTML5 is a standard for structuring and presenting content on the World Wide Web.

HTML5 is a cooperation between the World Wide Web Consortium (W3C) and the Web Hypertext Application Technology Working Group (WHATWG).

The new standard incorporates features like video playback and drag-and-drop that have been previously dependent on third-party browser plug-ins such as Adobe Flash, Microsoft Silverlight, and Google Gears.

### Browser Support:

The latest versions of Apple Safari, Google Chrome, Mozilla Firefox, and Opera all support many HTML5 features and Internet Explorer 9.0 will also have support for some HTML5 functionality.

The mobile web browsers that come pre-installed on iPhones, iPads, and Android phones all have excellent

support for HTML5.

### New Features:

HTML5 introduces a number of new elements and attributes that can help you in building modern websites. Here is a set of some of the most prominent features introduced in HTML5.

**New Semantic Elements** − These are like <header>, <footer>, and <section>.

**Forms 2.0** − Improvements to HTML web forms where new attributes have been introduced for <input> tag.

**Persistent Local Storage** − To achieve without resorting to third-party plugins.

**WebSocket** − A next-generation bidirectional communication technology for web applications.

**Server-Sent Events** − HTML5 introduces events which flow from web server to the web browsers and they are called Server-Sent Events (SSE).

**Canvas** − This supports a two-dimensional drawing surface that you can program with JavaScript.

**Audio & Video** − You can embed audio or video on your webpages without resorting to third-party plugins.

**Geolocation** − Now visitors can choose to share their physical location with your web application.

**Microdata** − This lets you create your own vocabularies beyond HTML5 and extend your web pages with custom semantics.

**Drag and drop** − Drag and drop the items from one location to another location on the same webpage.

### Backward Compatibility:

HTML5 is designed, as much as possible, to be backward compatible with existing web browsers. Its new features have been built on existing features and allow you to provide fallback content for older browsers.

It is suggested to detect support for individual HTML5 features using a few lines of JavaScript.

If you are not familiar with any previous version of HTML, I would recommend that you go through our HTML Tutorial before exploring the features of HTML5.

## Overview of CSS

**Cascading Style Sheets (CSS):** CSS is defined as a method sheet language that provides web designers control over how an internet site communicates with web browsers including the formatting and display of their HTML documents.

CSS or cascading sheet may be a text-based coding language that specifies the website formats and the way a site communicates with web browsers. The language allows web developers to regulate various style elements and functionalities, like layout, color, fonts, and therefore the formatting and display of HTML documents.

The main goal (as a method sheet language) was to separate document content from document presentation, which incorporates style elements, like color, layout, and fonts. CSS handles the design and feel a part of an internet page. Using CSS, you will control the color of the text, the design of fonts, the spacing between paragraphs, how columns are sized and laid out, etc.

CSS instructs the display of the HTML on how the web site will display at the user’s end. Let us have a glance on the benefits and drawbacks of CSS.

### Why CSS?

**CSS saves time:** You can write CSS once and reuse the same sheet in multiple HTML pages.

**Easy Maintenance:** To make a global change simply change the style, and all elements in all the webpages will be updated automatically.

**Search Engines:** CSS is considered a clean coding technique, which means search engines won’t have to struggle to “read” its content.

**Superior styles to HTML:** CSS has a much wider array of attributes than HTML, so you can give a far better look to your HTML page in comparison to HTML attributes.

**Offline Browsing:** CSS can store web applications locally with the help of an offline cache. Using this we can view offline websites.

### Advantages of CSS:

* + 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.
  + The main advantage of CSS is that style is applied consistently across variety of sites. One instruction can control several areas which is advantageous.
  + 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.

### 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.

## Overview of PHP

## PHP is a general-purpose scripting language geared towards web development. It was originally created by Danish-Canadian programmer Rasmus Lerdorf in 1993 and released in 1995. The PHP reference implementation is now produced by the PHP Group. PHP was originally an abbreviation of Personal Home Page, but it now stands for the recursive acronym PHP: Hypertext Preprocessor.

PHP code is usually processed on a web server by a PHP interpreter implemented as a module, a daemon or a Common Gateway Interface (CGI) executable. On a web server, the result of the interpreted and executed PHP code—which may be any type of data, such as generated HTML or binary image data—would form the whole or part of an HTTP response. Various web template systems, web content management systems, and web frameworks exist that can be employed to orchestrate or facilitate the generation of that response. Additionally, PHP can be used for many programming tasks outside the web context, such as standalone graphical applications[16] and drone control.[17] PHP code can also be directly executed from the command line.

### Advantages of PHP:

* + Ease of Learning and Use: PHP has a straightforward syntax, making it easy for beginners to learn and understand, especially for those familiar with HTML and JavaScript.
  + Open Source and Free: PHP is free to use and widely supported, with an active community that contributes to a wealth of libraries, frameworks, and resources.
  + Platform Independence: PHP can run on various operating systems (Windows, Linux, macOS) and is compatible with most web servers, making it highly versatile.
  + Performance: For small to medium-sized applications, PHP provides decent performance. It has also evolved over the years with significant improvements in speed and memory management in newer versions.
  + Wide Range of Frameworks: PHP has popular frameworks like Laravel, Symfony, CodeIgniter, and Zend, which speed up development by providing pre-built modules, security features, and MVC architecture.

**Disadvantages of CSS:**

* + Security Concerns: PHP's open-source nature and its extensive use make it a target for malicious attacks. If not coded carefully, PHP applications can be vulnerable to security issues like SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF).
  + Inconsistent Syntax: PHP’s syntax has evolved over the years, resulting in inconsistencies. This can make the language confusing, especially for new developers or those migrating from other languages.
  + Limited Modular Design: While PHP supports object-oriented programming, it’s not as robust as some modern languages (e.g., Java, C#). This can make managing larger applications challenging without strict architectural practices.
  + Performance Issues in Large Applications: For highly complex, large-scale applications, PHP can be slower compared to languages like Java or Python. High-concurrency applications may require more optimizations or alternatives.

## Overview of an application using HTML, CSS and JavaScript

There are many web languages available, however we’re just going to look at three of them. They are HTML, CSS and JavaScript and they are considered to be the backbone of the web. When it comes to web development there is front-end web development and back end-web development. These three languages are for front-end web development and are responsible for what you can see and do on a website. They are referred to as client side languages as they run in the browser (Google Chrome, Firefox etc.) of your computer. The browser translates these languages and the result of this translation is the visual web page.

It’s important to note HTML and CSS are not considered to be programming languages. HTML is a markup language and CSS is a styling language. JavaScript, however, is a programming language. Hence, they are all web languages, but they perform different jobs.

For those that are new to web design the easiest way to understand this relationship is to look at an analogy that will break it down into bite-size chunks. One way to look at front-end web development is to see these three languages interact with one another the same way a human body interacts with its surroundings.

### The Website as a Human Body

To make this easy to understand, I’m going to refer to these three web languages as different aspects of the human body. We’re going to look at the body itself as a sort of command centre; the accessories that are worn on the body as a way of representing personal style; and then the actions that the body is capable of as a way of animating ourselves. So for the purpose of this discussion, HTML will be referred to as the physical body, CSS will be the body’s accessories and JavaScript is the body’s ability to talk or move. All of these ‘bodily aspects’ need to work together to form a functional, visually appealing, interactive website.

### HTML

* + The Body
  + HyperText Markup Language (HTML)
  + Content and basic structure
  + Describes and defines
  + Made up of tags
  + Tells the browser what to display

**HyperText Markup Language** (HTML) can be broken down into HyperText, which is what grants access

to other texts through links, and Markup which outlines the basic structure and appearance of raw text. What this means is that HTML describes and defines the content and basic structure of the website. It does this through a means of special tags or codes which tell the browser what to do. HTML is the bare basics of a website.

An HTML only website can be compared to a functioning human body. Note, I didn’t say fully-functional. An HTML only website has all of its body parts, although it doesn’t offer much to look at because it doesn’t have any accessories or personal style. At this stage, it’s also a body which is not capable of moving or speaking. A website which consists of only HTML would probably look a little like this:

This is where CSS comes in.

### CSS

* + The Accessories
  + Cascading Style Sheet (CSS)
  + Gives style and structure to the content
  + Link the CSS file to the HTML
  + Tells the browser how to display

A **Cascading Style Sheet** is the website’s accessories. It’s responsible for outlining the colours, font and positioning of the content on a website. It adds some style and structure to the content. In order to make use of the CSS capabilities it needs to be linked within the HTML content so that style can be added to the website. CSS will tell the browser how to display the existing HTML.

CSS can be compared to adding personal style to the body. When you link CSS to HTML, it’s like dressing up the body. For example, you can choose a specific colour shirt and match it with a specific colour pair of trousers. On a website, you can choose the colour of the background or the font size of a heading and much more. It’s important to note that CSS cannot live without HTML as there would be nothing to style. Just like clothes or shoes would be pointless without someone to wear them.

So by now you should have an understanding of how structure and style are constructed on a website. A website that consists of HTML and CSS might looks like this:

However, you can’t help but notice that something is missing. The web page is lacking certain functions like a search box or options to comment. Right now the body, with all its accessories, looks more like a mannequin in a store window than a real human being. That’s where JavaScript comes in.

### JavaScript

* + The body’s ability to perform actions
  + JavaScript is not Java
  + Behaviour of the website
  + Used for interactive functionality
  + Allows for the user to interact with the browser

JavaScript controls the behaviour of the website. It’s important to note that JavaScript and Java are two different things. JavaScript was designed to manipulate web pages and it is used to create interactive functionality. Without JavaScript a website will still be functional, but in a limited way. JavaScript is what animates HTML and CSS, and it’s what brings your website to life.

JavaScript can be compared to the body’s ability to perform actions such as walking or talking. So when you add JavaScript to HTML and CSS, it transforms the body from being a beautifully dressed mannequin into a real-life walking talking human being. It animates the body, giving it lifelike qualities. JavaScript can also be compared to a fully functional body that has the ability to interact. As we all know, having an interactive website is critical, otherwise its just a boring page filled with information. Here we see a website which consists of HTML, CSS and JavaScript:

If you look at this example of twitter, JavaScript allows you to expand the tweet to see re-tweets, to set a tweet as a favourite and more. A popular JavaScript App is Google Maps.

# CODING

### This is House Price Prediction Using Machine Learning using python and Machine learning

### Python Code:

**import pandas as pd**

**import matplotlib.pyplot as plt**

**import seaborn as sns**

**dataset = pd.read\_excel("HousePricePrediction.xlsx")**

**# Printing first 5 records of the dataset**

**print(dataset.head(5))**

**dataset.shape**

**obj = (dataset.dtypes == 'object')**

**object\_cols = list(obj[obj].index)**

**print("Categorical variables:",len(object\_cols))**

**int\_ = (dataset.dtypes == 'int')**

**num\_cols = list(int\_[int\_].index)**

**print("Integer variables:",len(num\_cols))**

**fl = (dataset.dtypes == 'float')**

**fl\_cols = list(fl[fl].index)**

**print("Float variables:",len(fl\_cols))**

**numerical\_dataset = dataset.select\_dtypes(include=['number'])**

**plt.figure(figsize=(12, 6))**

**sns.heatmap(numerical\_dataset.corr(),**

**cmap = 'BrBG',**

**fmt = '.2f',**

**linewidths = 2,**

**annot = True)**

**unique\_values = []**

**for col in object\_cols:**

**unique\_values.append(dataset[col].unique().size)**

**plt.figure(figsize=(10,6))**

**plt.title('No. Unique values of Categorical Features')**

**plt.xticks(rotation=90)**

**sns.barplot(x=object\_cols,y=unique\_values)**

**plt.figure(figsize=(18, 36))**

**plt.title('Categorical Features: Distribution')**

**plt.xticks(rotation=90)**

**index = 1**

**for col in object\_cols:**

**y = dataset[col].value\_counts()**

**plt.subplot(11, 4, index)**

**plt.xticks(rotation=90)**

**sns.barplot(x=list(y.index), y=y)**

**index += 1**

**dataset.drop(['Id'], axis=1, inplace=True)**

**dataset['SalePrice'] = dataset['SalePrice'].fillna( dataset['SalePrice'].mean())**

**new\_dataset = dataset.dropna()**

**new\_dataset.isnull().sum()**

**from sklearn.preprocessing import OneHotEncoder**

**s = (new\_dataset.dtypes == 'object')**

**object\_cols = list(s[s].index)**

**print("Categorical variables:")**

**print(object\_cols)**

**print('No. of. categorical features: ',**

**len(object\_cols))**

**from sklearn.preprocessing import OneHotEncoder**

**import pandas as pd**

**# Assuming 'new\_dataset' is already defined and 'object\_cols' is a list of categorical column names**

**OH\_encoder = OneHotEncoder(sparse\_output=False, handle\_unknown='ignore')**

**OH\_cols = pd.DataFrame(OH\_encoder.fit\_transform(new\_dataset[object\_cols]))**

**# Setting index and column names to match 'new\_dataset'**

**OH\_cols.index = new\_dataset.index**

**OH\_cols.columns = OH\_encoder.get\_feature\_names\_out(object\_cols)**

**# Dropping original categorical columns from 'new\_dataset'**

**df\_final = new\_dataset.drop(object\_cols, axis=1)**

**# Concatenating the new one-hot encoded columns**

**df\_final = pd.concat([df\_final, OH\_cols], axis=1)**

**from sklearn.metrics import mean\_absolute\_error**

**from sklearn.model\_selection import train\_test\_split**

**X = df\_final.drop(['SalePrice'], axis=1)**

**Y = df\_final['SalePrice']**

**# Split the training set into**

**# training and validation set**

**X\_train, X\_valid, Y\_train, Y\_valid = train\_test\_split(**

**X, Y, train\_size=0.8, test\_size=0.2, random\_state=0)**

**from sklearn import svm**

**from sklearn.svm import SVC**

**from sklearn.metrics import mean\_absolute\_percentage\_error**

**model\_SVR = svm.SVR()**

**model\_SVR.fit(X\_train,Y\_train)**

**Y\_pred = model\_SVR.predict(X\_valid)**

**print(mean\_absolute\_percentage\_error(Y\_valid, Y\_pred))**

**from sklearn.ensemble import RandomForestRegressor**

**model\_RFR = RandomForestRegressor(n\_estimators=10)**

**model\_RFR.fit(X\_train, Y\_train)**

**Y\_pred = model\_RFR.predict(X\_valid)**

**mean\_absolute\_percentage\_error(Y\_valid, Y\_pred)**

**from sklearn.linear\_model import LinearRegression**

**model\_LR = LinearRegression()**

**model\_LR.fit(X\_train, Y\_train)**

**Y\_pred = model\_LR.predict(X\_valid)**

**print(mean\_absolute\_percentage\_error(Y\_valid, Y\_pred))**

**!pip install catboost**

**from catboost import CatBoostRegressor**

**from sklearn.metrics import r2\_score**

**cb\_model = CatBoostRegressor()**

**cb\_model.fit(X\_train, Y\_train)**

**preds = cb\_model.predict(X\_valid)**

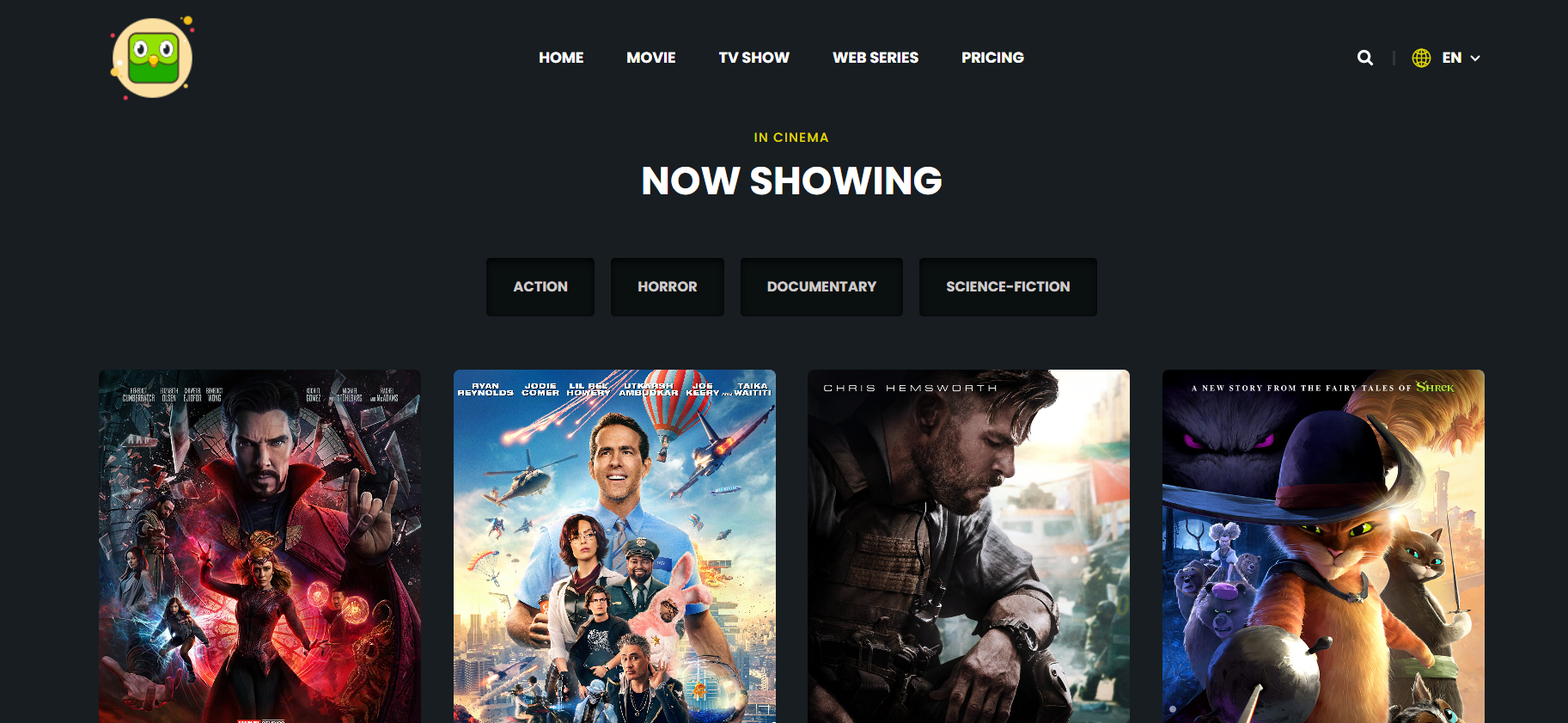
**cb\_r2\_score=r2\_score(Y\_valid, preds)**

**cb\_r2\_score**

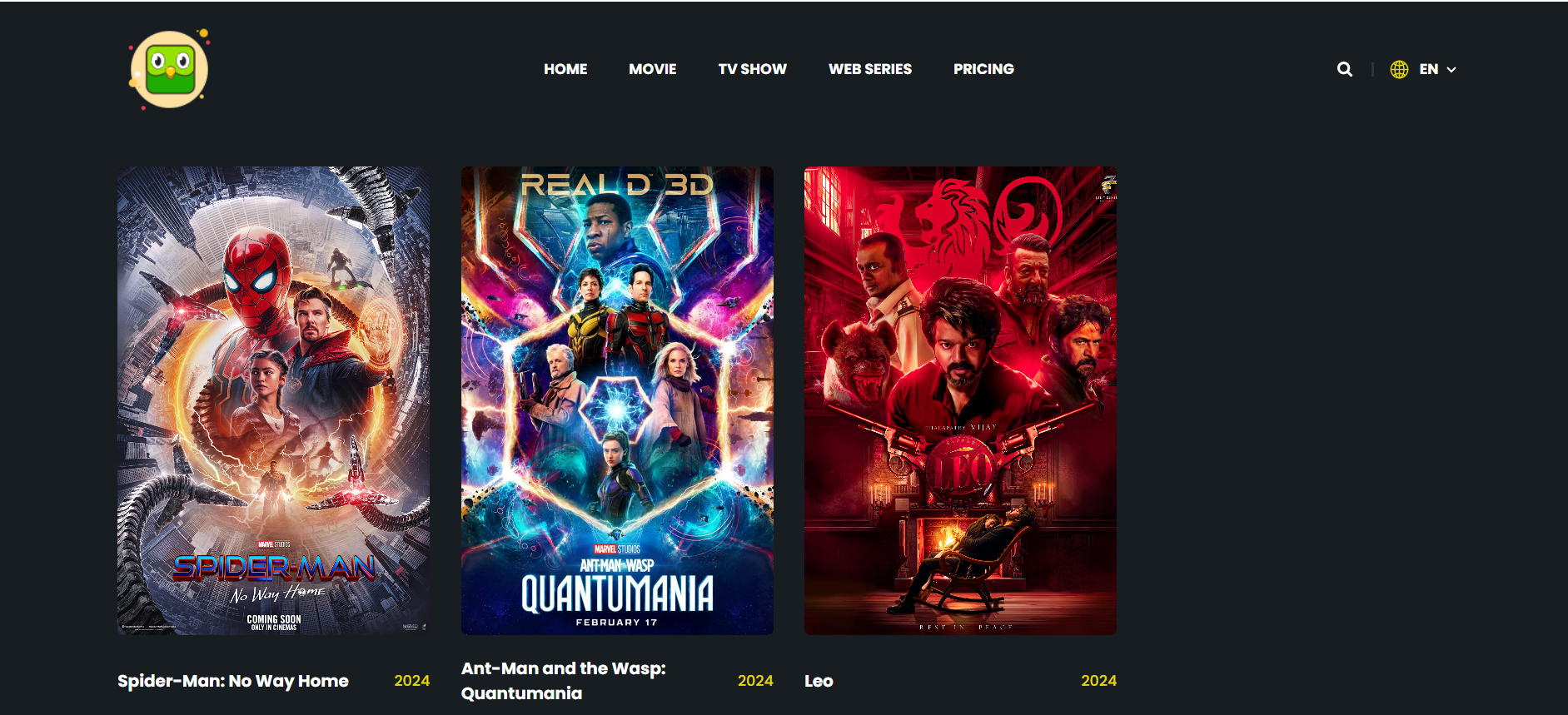
# RESULT

# User Mode

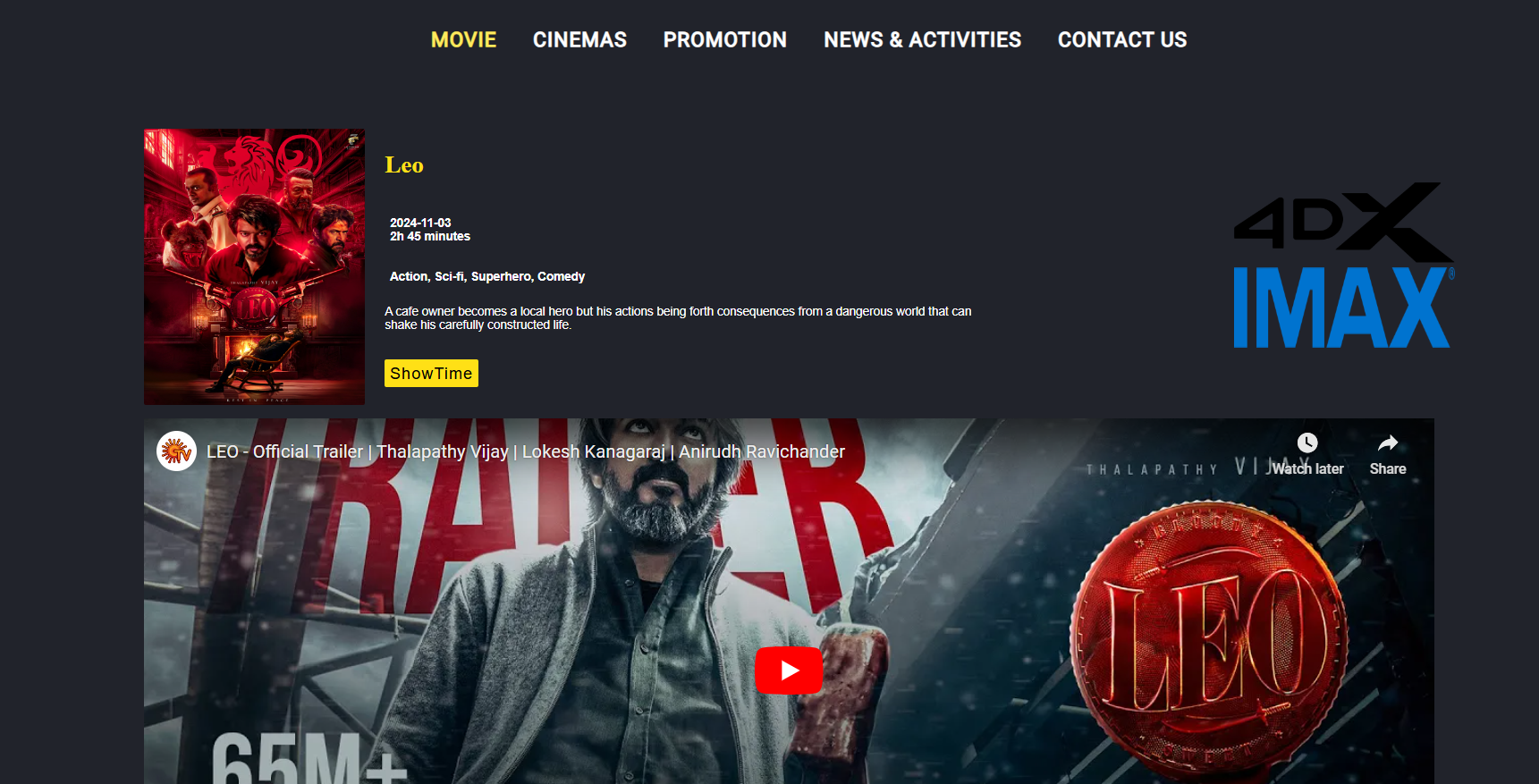
* 1. **Home page:**

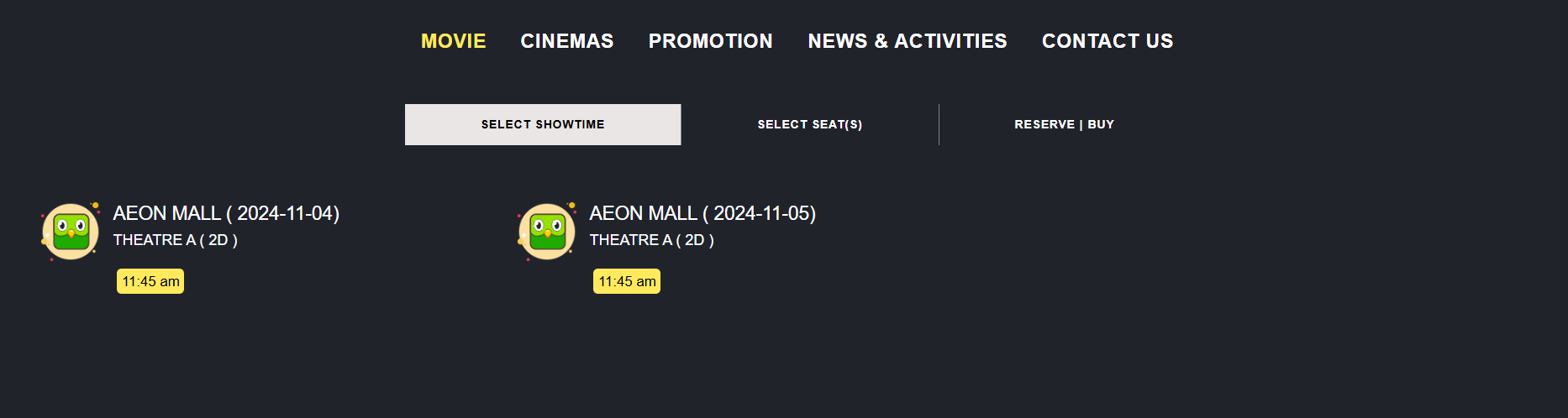


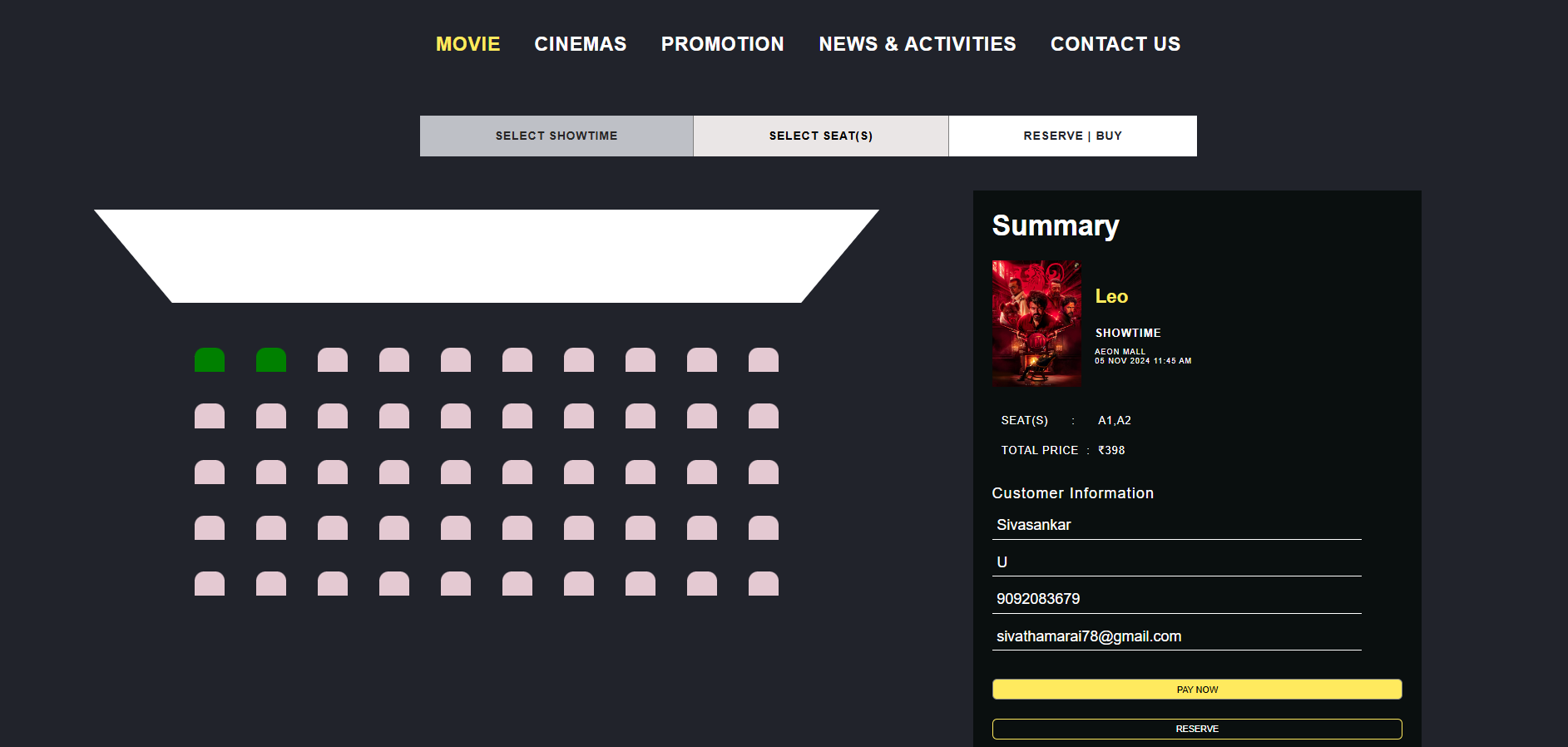
* 1. **Home page:**

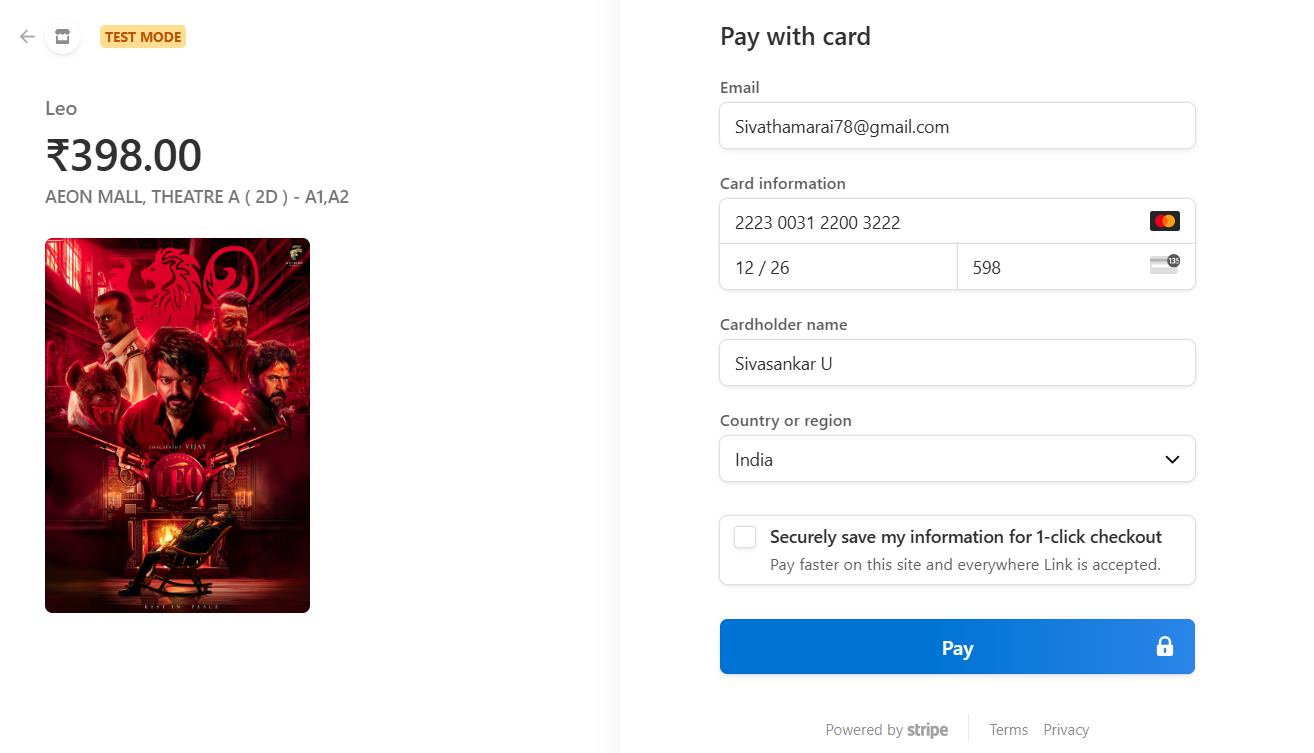


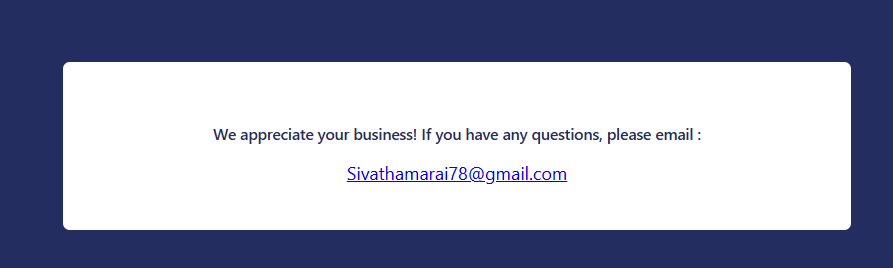
* 1. **Booking:**

****

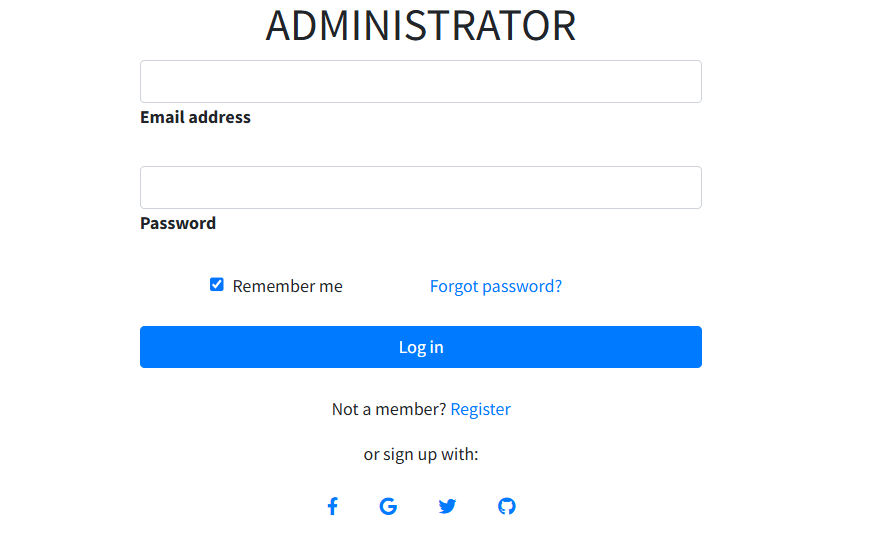
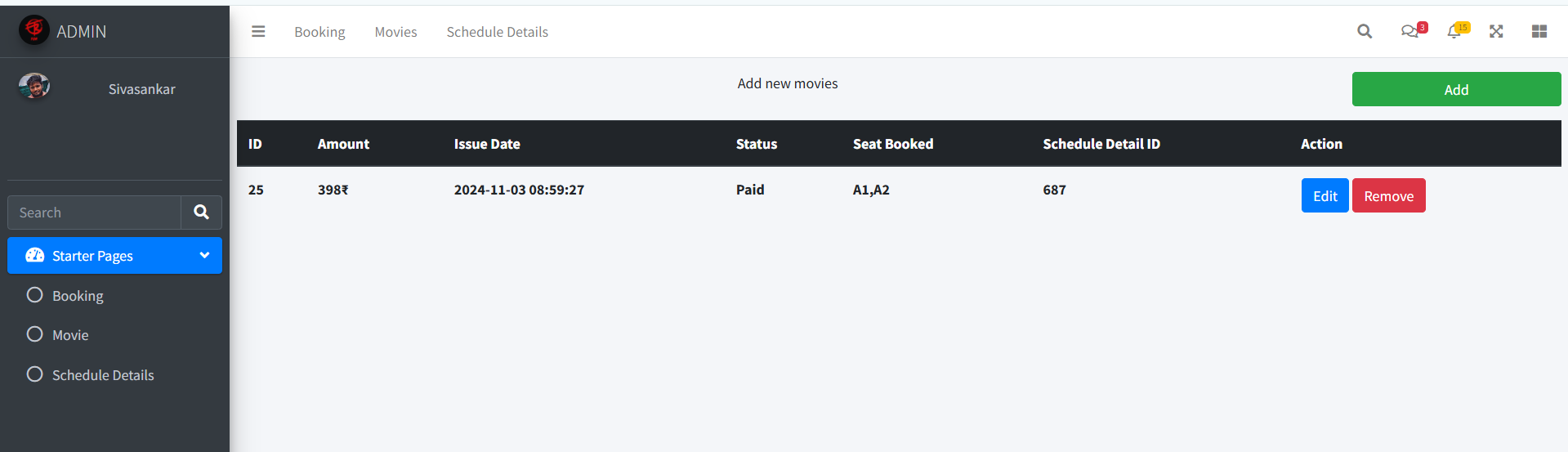
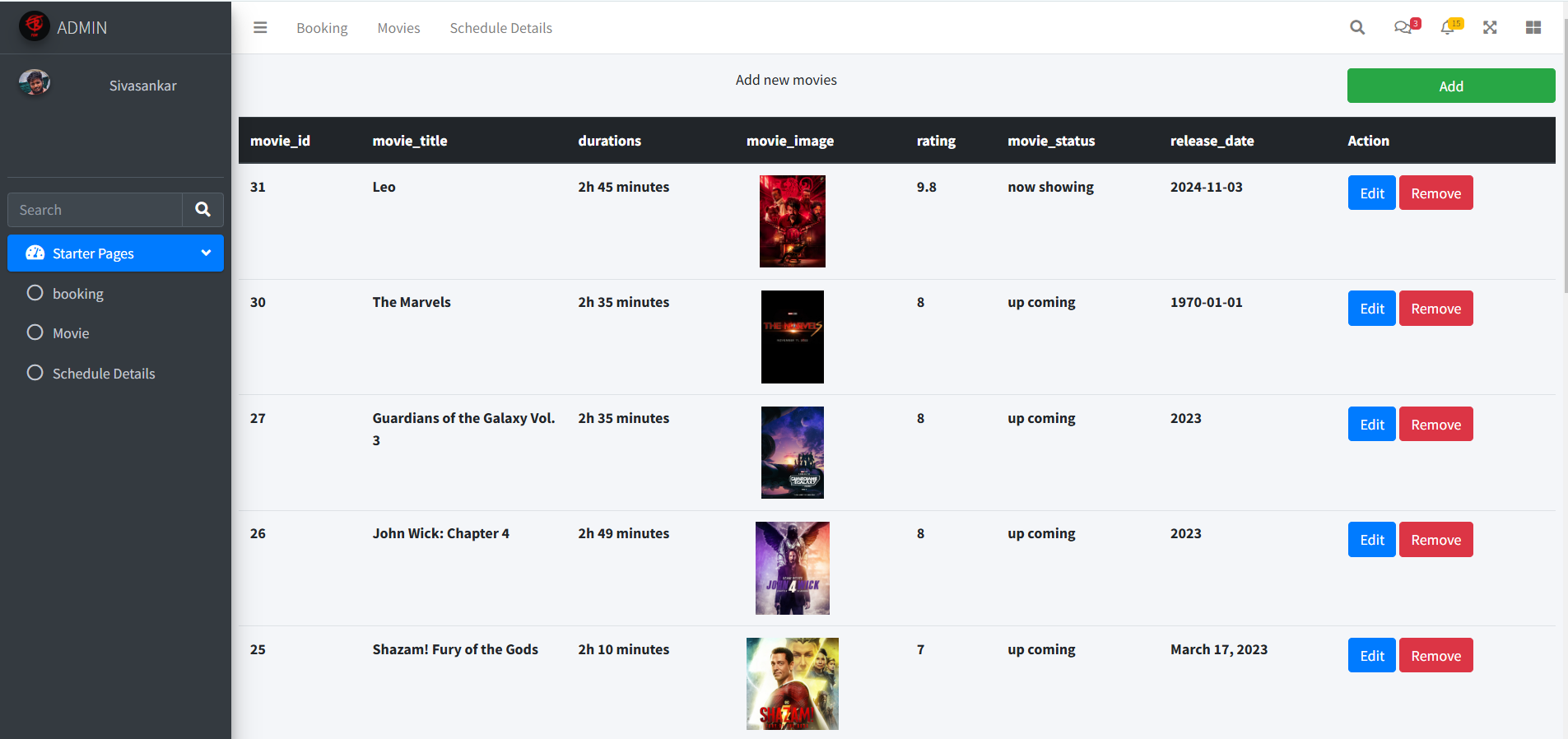
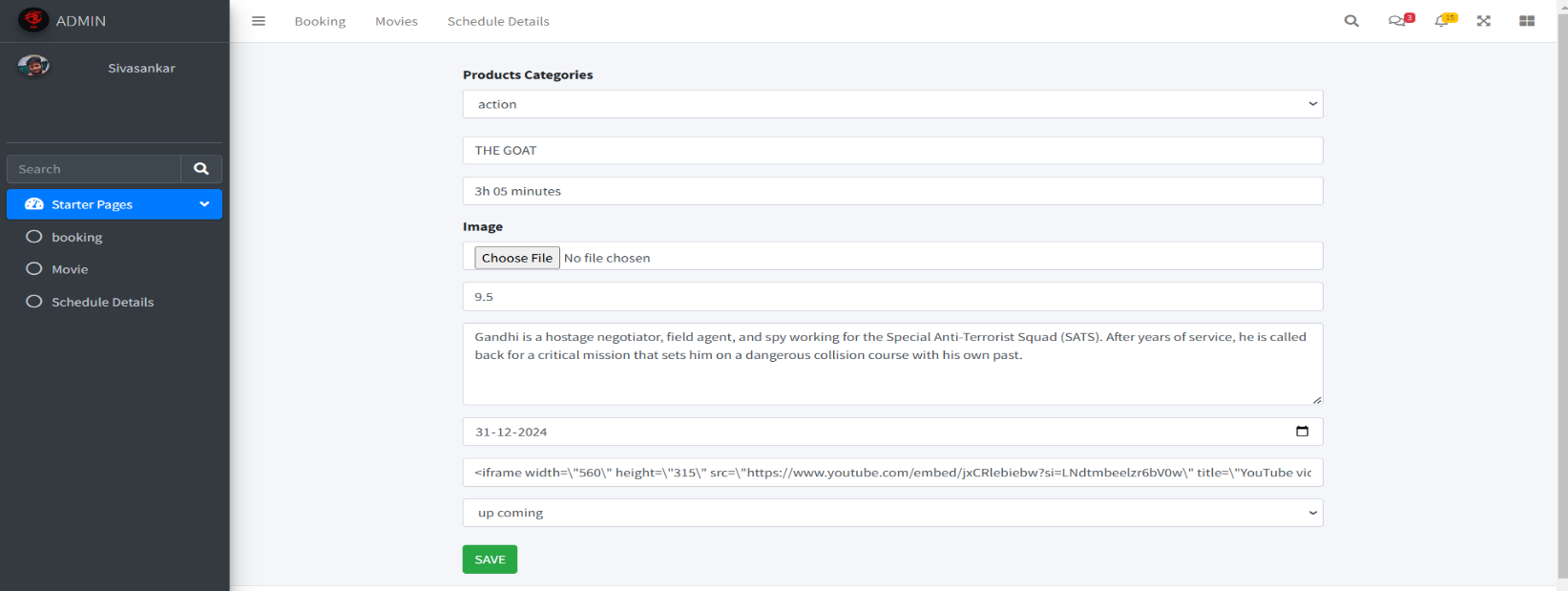
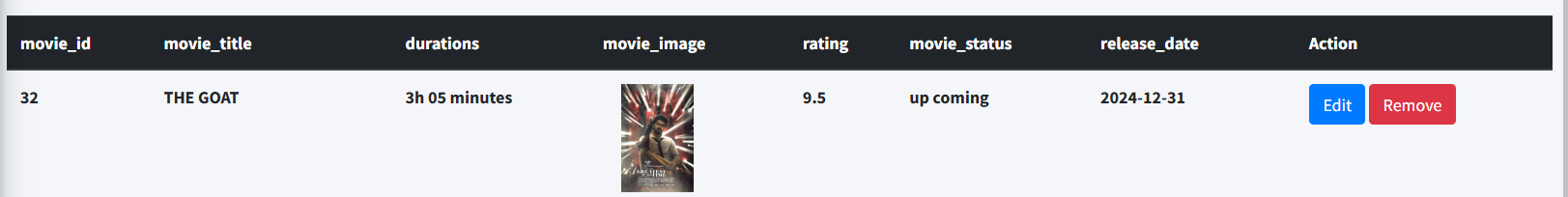
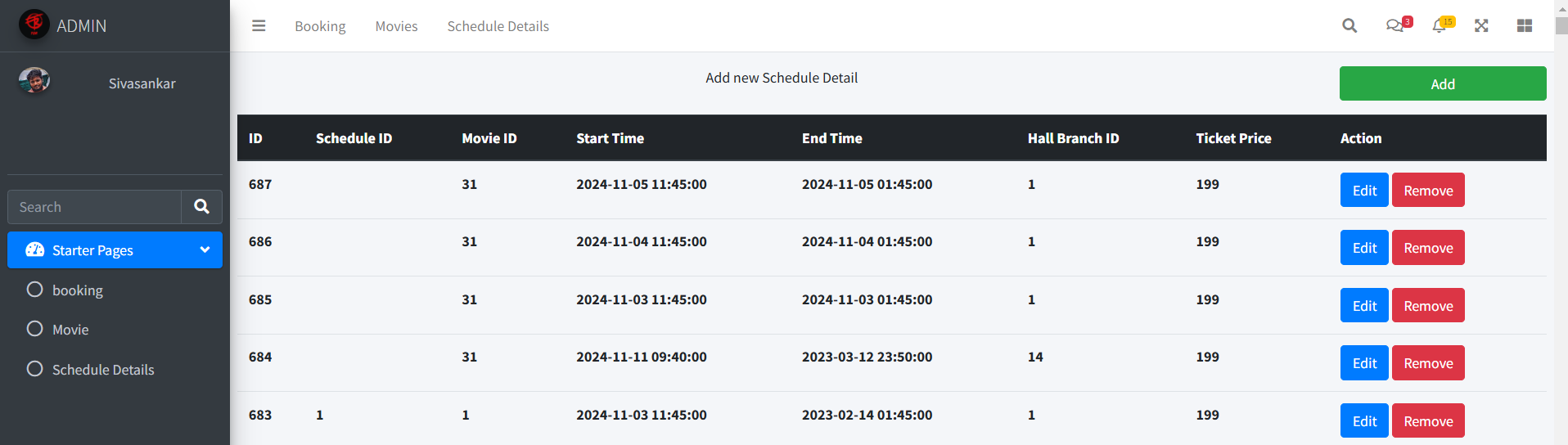
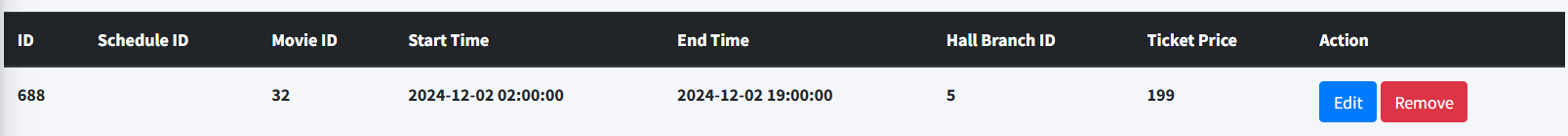
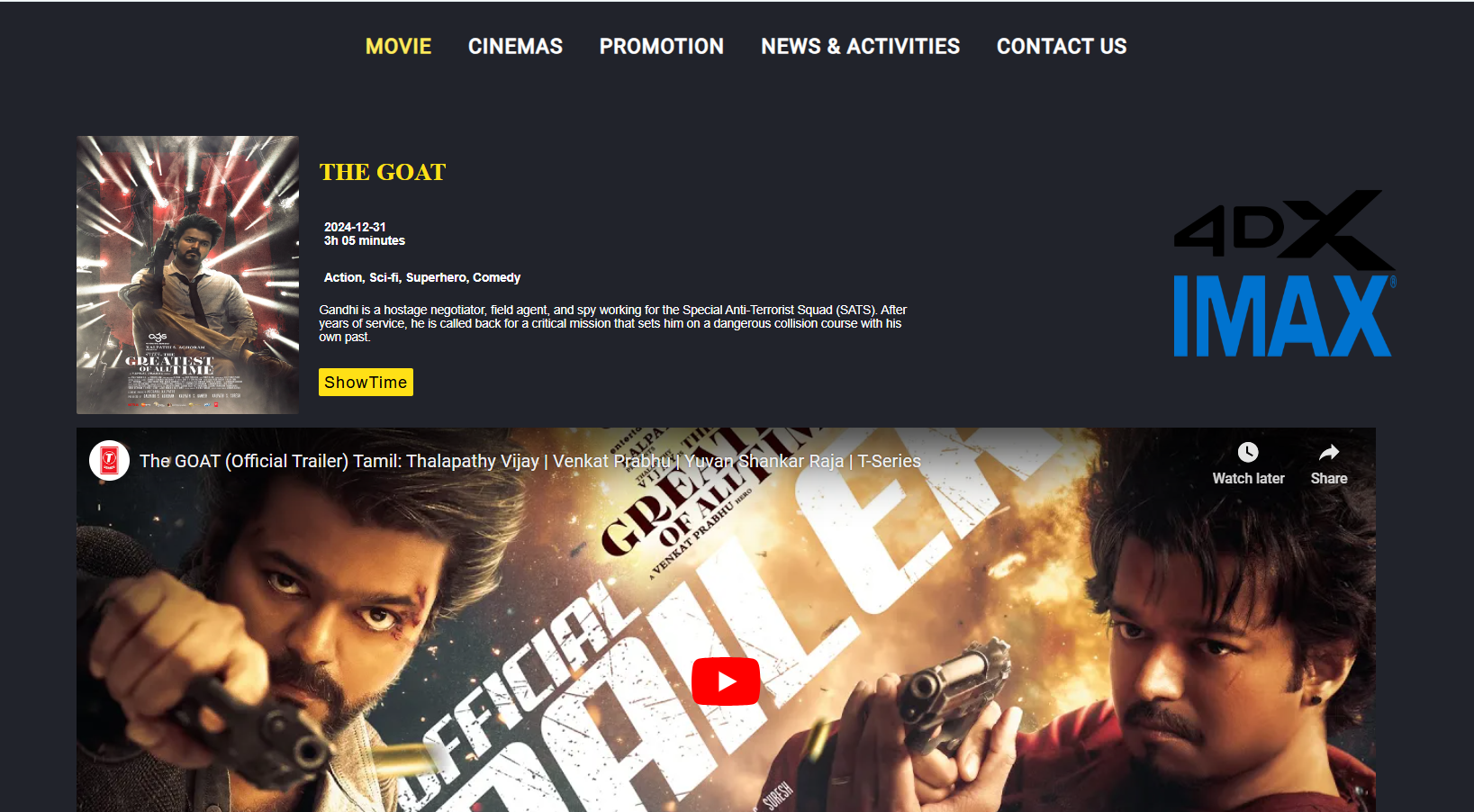


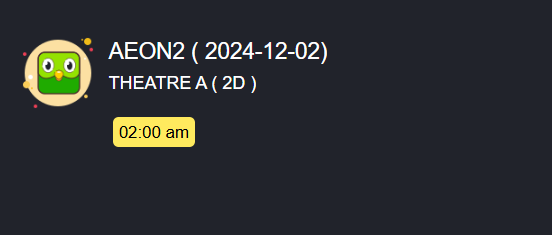






# Admin Mode

* 1. **Login:**
  2. **Once login you can see the booked details**
  3. **Click Movies to see the movies in cinema.**
  4. **Click Add to add the new movie.**
  5. **In Schedule details we can see the current movie schedule hall and time**
  6. **Click add to update the new add movie show and timing.**
  7. **Finally the Movie and the show time displayed.**



# CONCLUSION

In an Existing Team we address these challenges and present an approach to efficient, incremental consolidation of the requirement. Following common practice, our method iterates over information requirements to create the final design. we show how to efficiently accommodate a new information requirement to an existing design and also, how to update a design in life of an evolving information requirement. we came up with an idea to create web application where we can view the book and update the new movies with cinema hall and timings. For this app creation I have taken the initiative a proposed a blueprint of the output how the Movies list will be shown and search for the movies. The final design satisfying all requirements is created. The team was highly impressed with the plan i was solely assigned to develop the blueprint into final outcome. Developed a single Movie App booking management application using HTML, CSS, JavaScript, PHP and the team was highly impressed with the outcome.

# REFERENCES

Best reference sites for HTML, CSS, PHP and JavaScript programming:

* + 1. W3schools: [PHP](http://www.w3schools.com/htmldom/dom_reference.asp), [JavaScript](http://www.w3schools.com/jsref/default.asp), [HTML, CSS, XML, Multimedia, Server-side](http://www.w3schools.com/sitemap/sitemap_references.asp)
    2. MDN Web Docs : comprehensive resources for [HTML](https://developer.mozilla.org/en-US/docs/Web/HTML), [CSS](https://developer.mozilla.org/en-US/docs/Web/CSS), and [JavaScript](https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/JavaScript_basics)
    3. CSS-Tricks: guides, and examples specifically for [CSS](https://css-tricks.com/category/notes/)
    4. JavaScript.info: [JavaScript](https://javascript.info/) guide
    5. [MySQL](https://dev.mysql.com/doc/) Official Documentation
    6. [MySQL](https://www.tutorialspoint.com/mysql/index.htm) concepts from beginner to advanced levels
    7. Practical examples on [MySQL](https://www.geeksforgeeks.org/mysql-subquery/?ref=gcse)
    8. [PHP](https://www.php.net/docs.php).net Official Documentation
    9. [PHP](https://phptherightway.com/) The Right Way
    10. Tuts+ [PHP](https://code.tutsplus.com/)
    11. Stripe [API](https://stripe.com/docs/api/?lang=php#intro) for payments
    12. [Stripe](https://docs.stripe.com/testing) card documentation
    13. [Google](https://fonts.googleapis.com) Font API
    14. W3 schools:MYSQL DATABASE
    15. ZVON: [HTML, XML, CSS](http://www.zvon.org/)
    16. SitePoint: [JavaScript, HTML, CSS, PHP](https://www.sitepoint.com/)