DESIGN AND DEVELOPMENT OF MOVIETIC

A MAJOR PROJECT-I REPORT

Submitted in partial fulfillment of the requirements

for the degree of

BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE & ENGINEERING

By

GROUP NO. 05

Daksh Raj Singh	**0536CS201012**
Dikshant Kumar Bhatiya	**0536CS201014**
Deepak Kumar	**0187CS201048**
Naina Soni	**0187CS201103**
Shreya Singh	**0187CS201161**

Under the guidance of

Prof. Bhavna Soni
(Assistant Professor)

December - 2023

Department of COMPUTER SCIENCE & ENGINEERING
Sagar Institute of Science & Technology (SISTec)
Bhopal (M.P.)
Approved by AICTE, New Delhi & Govt. of M.P.
Affiliated to Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal (M.P.)

Sagar Institute of Science Technology (SISTec), Bhopal
Department of COMPUTER SCIENCE ENGINEERING
Bhopal (M.P.)

December - 2023

CERTIFICATE

I hereby certify that the work which is being presented in the B.Tech Major Project-I Report entitled **MOVIETIC** in partial fulfillment of the requirements for the award of the degree of ***Bachelor of Technology*** in ***Computer Science & Engineering*** and submitted to the Department of Computer Science & Engineering, Sagar Institute of Science & Technology (*SISTec*), Bhopal (M.P.) is an authentic record of my own work carried out during the period from July-2023 to December-2023 under the supervision of **Prof. Bhavna Soni (Project guide)**. The content presented in this project has not been submitted by me for the award of any other degree elsewhere.

Signature

Daksh Raj Singh	**0536CS201012**
Dikshant Kumar Bhatiya	**0536CS201014**
Deepak Kumar	**0187CS201048**
Naina Soni	**0187CS201103**
Shreya Singh	**0187CS201161**

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

Date:

Prof. Bhavna Soni	**Dr. Rahul Dubey**	**Dr. Dinesh Kumar Rajoriya**
Project Guide	**HOD**	**Principal**

ABSTRACT

To define, design, and implement a Database to aid in comprehensively managing all aspects of a theatre. This system deals with a single branch of a theatre. It keeps track of the various halls, currently showing movies, show timings, booked tickets, and price listings in the theatre as well as the associated attributes. It is possible for the management of the theatre to use the database to fully understand all parts of the system without use of additional software. The system allows a cashier to add tickets for a given show, and for a manager to add movies and shows to the database. The system automatically checks the validity of the new entries and disallows incorrect data from being entered. This is done automatically without user intervention

ACKNOWLEDGEMENT

We take this occasion to thank God, almighty for blessing us with his grace and taking our endeavor to a successful culmination. We extend our sincere and heartfelt thanks to our esteemed guide,

Prof. Bhavna Soni , for providing us with the right guidance and advice at the crucial junctures and for showing us the right way. We also take this opportunity to express a deep sense of gratitudeto **Dr.Rahul dubey (HOD)** and **Prof. Deepti Jain (Project Coordinator) .** We also wish to expressour gratitude to **Dr. Swati Saxena (Vice Principal)** and **Dr Dinesh Kumar Rajoriya (Principal)**.

First of all, I would like to thanks all those people who helped me directly or indirectly to complete my project whenever I found my self in problems. Our all faculties encourages me and due to their kindness and helpful nature and help I got very much confidence to complete this project.

I am deeply inherited who devoted his precious time in giving me the information about the various aspect and gave support and guidance at every point of time. I am really thankful to their kind and supportive nature. His inspiring nature has always made my work easy.

It’s a very difficult task to build a project from scratch and take it up to the expectation of our teachers, I would like to thank my teachers my friends who helped us in this crucial time and encouraged us to build a project up to the mark so that we can present it to our teachers. We would like to thank our friends and family for the support and encouragement they have given us during the course of our work.

LIST OF FIGURES

LIST OF ABBREVIATIONS

ACRONYM	FULL FORM
HTML	Hyper Text Mark-up Language
CSS	Cascading Style Sheet
VSCODE	Visual Studio Code
AJAX	Asynchronous JavaScript And XML
XML	Extensible Markup Language
SQL	Structured Query Language
HTTP	HyperText Transfer Protocol
PIP	Python Package Manager
Distro	Distribution

TABLE OF CONTENTS

Chapter 1
Introduction

CHAPTER 1
INTRODUCTION

1.1 Introduction

This project is a Theatre Management System. It is a website made in HTML connected to a MySQL Database using Python with JavaScript and CSS enhancements. The database has tables dealing with all aspects of a theatre: halls, movies available, current showing, price lists, and booked tickets. These are accessed by the webpage to show relevant details for a user. Thus, by use of this project, the core aspects of a theatre are easily maintained and manipulated as needed by the manager.

1.1.1 Purpose of the project

This project is aimed at small theatres. It is a tool to be used to simplify the process of adding, storing, viewingand manipulating the movies available at the theatre, the various showings of the movie, the tickets booked for the shows and the pricing of the tickets. It is a robust and comprehensive system that is minimalistic and simplefor use by cashiers and managers who have no prior training.

1.1.2 Scope

The scope of this project is at the ticket booking kiosk of the cashier and at the manager's desk. It is only concerned with active use of resources of the theatre and does not have long term storage of information. It only deals with movies, movie showings and booked tickets, and does not handle other aspects, such as onlinebooking, concessions, etc.

1.1.3 Existing System

The currently used system in theatre management is highly primitive and/or inefficient and/or slow. It is, in some cases, worked on in ledgers and with a manual booking system. Even in cases where it is on computers instead of a manual system, it is not new or often upgraded/maintained as required to make it work properly. Thus, our system is brought in to replace it.

1.1.4 Proposed System

The proposed system is a fully functional system that takes care of all aspects related to movies on-site. It allows cashiers to book tickets and the manager to add movies and shows. The system is designed so as to be used by anyone with a basic understanding of english and no other training at all.

Chapter 2
Software & Hardware Requirements

CHAPTER 2
SOFTWARE AND HARDWARE REQUIREMENT

2.1 RECOMMENDED OPERATING SYSTEM

- **Windows:** 7 or newer
- **MAC:** OS X v10.7 or higher
- **Linux:** Ubuntu

2.2 SOFTWARE REQUIREMENTS

People often ask what browser they should use. There is no single answer for this. Use whichever browser works best on your computer. However, we recommend downloading Firefox and/or Chrome in addition to having Internet Explorer or Safari.

- Firefox
- Chrome

2.3 HARDWARE REQUIREMENTS

We strongly recommend a computer fewer than 5 years old.

- Processor: Minimum 1 GHz; Recommended 2GHz or more
- Ethernet connection (LAN) OR a wireless adapter (Wi-Fi)
- Hard Drive: Minimum 32 GB; Recommended 64 GB or more
- Memory (RAM): Minimum 1 GB; Recommended 4 GB or above

Chapter 3
Problem Description

CHAPTER 3
PROBLEM DESCRIPTION

Small theatres generally have old or primitive management systems which they do not replace or upgrade as it is perceived to be not worth the cost compared to revenues. However, old systems can be very slow and inefficient and as such, slow down the process. Furthermore, they are difficult to maintain and the cost of doing so can build up over time and eventually exceed costs of not replacing the system. This Project is designed for use by such cinemas, who can use this highly minimalistic design to replace current systems at low cost and with no additional training requirements. The code behind the project is designed to put maximum functionality within it so as reduce cost and difficulty of maintenance and future upgrades.

Chapter 4
Literature Survey

CHAPTER 4
LITRATURE SURVEY

Theatre Management and Ticket Booking Systems (TMTBS) have evolved significantly in recent years due to advancements in technology and changes in consumer expectations. This literature survey provides an overview of key research and developments in the field of TMTBS, highlighting trends, challenges, and innovative solutions.

1. Ticket Booking Trends:

Online Ticket Booking and Mobile Apps: Research has shown a notable shift from traditional box office sales to online ticket booking platforms and mobile apps. Customers prefer the convenience of booking tickets from their smartphones and computers (Guttentag, 2015).

Dynamic Pricing: Studies suggest that dynamic pricing strategies, which adjust ticket prices based on demand and availability, have gained prominence in the industry. These strategies can optimize revenue for theatres (Chen et al., 2019).

2. User Experience and Customer Engagement:

Personalization: Researchers emphasize the importance of personalization in TMTBS. Personalized recommendations, targeted promotions, and tailored communication enhance the customer experience (Vural et al., 2018).

Customer Feedback Analysis: Literature highlights the significance of analyzing customer feedback and reviews to improve services and shows. Natural language processing techniques are employed for sentiment analysis (He et al., 2017).

3. Show Scheduling and Management:

Optimization Algorithms: Operations research and optimization algorithms are used to efficiently schedule shows, allocate resources, and minimize conflicts in multi-screen and multi-event venues (Agrawal et al., 2016).

4. Security and Privacy:

Data Security: Ensuring data security in TMTBS is critical. Research explores encryption methods, access controls, and compliance with data protection regulations (Shetty et al., 2019).

5. Integration and Scalability:

Integration with Third-party Systems: Researchers emphasize the importance of integrating TMTBS with external systems, such as payment gateways, marketing tools, and analytics platforms, to enhance functionality (Kim et al., 2020).

Scalability: Scalability is a key concern in TMTBS. Studies investigate scalable architecture design and cloud-based solutions to handle peak traffic loads (Ghani et al., 2017).

6. Emerging Technologies:

Blockchain: Some studies explore the potential of blockchain technology for secure ticketing, reducing fraud, and enhancing transparency in the ticketing process (Hassan et al., 2019).

Artificial Intelligence: AI and machine learning are employed for predictive analytics in ticket sales and customer behavior analysis (Fan et al., 2018).

7. COVID-19 Impact:

The COVID-19 pandemic significantly impacted the theatre industry. Research discusses how TMTBS adapted to the pandemic, including the implementation of social distancing measures, contactless ticketing, and virtual shows (Ku et al., 2021).

The literature survey reveals a dynamic and evolving landscape in Theatre Management and Ticket Booking Systems. Technology-driven innovations, customer-centric approaches, and adaptability to changing circumstances, such as the pandemic, are reshaping the theatre industry. Future research in this field will likely continue to explore emerging technologies and novel strategies to enhance the theatre experience while ensuring efficiency and security in ticketing and management processes.

Chapter 5
Software Requirement Specification

CHAPTER 5
SOFTWARE REQUIREMENTS SPECIFICATION

5.1 FUNCTIONAL REQUIREMENTS:

5.1.1 Manager :

1. The ability to insert new Movies. Movie name, language, length, showing start date and end date are inputted. In case of incorrect or invalid input, entry is nullified. If correctly entered, then a movie ID is randomly generated and the data is passed to MySQL via JavaScript, Flask and Python.
2. Similarly, Shows can be entered by entering movie, hall, time and date. If correctly entered, a show ID is randomly generated and ask data is put into database. Then, the price ID is fetched and stored after. Shows cannot be added if an invalid type is chosen or if the hall in which it is to be shown is already occupied.
3. The manager can view tickets that have been booked by the cashier, such as tickets numbers and seat numbers for a given show that have been booked.
4. The manager can alter prices. The list of prices based on day and your is brought up. Then, a price is then selected and altered.
5. The cashier had one functionality. The cashier books tickets for the customer. The cashier selects a movie, and based on that, a showing is selected for a time and date. Then a seating chart is brought up and a seat is chosen. Then a unique ticket ID is generated and the data is stored in the database.
6. All data is stored on a common database. All systems in the theatre (cashier’s terminals and manager’s computer) access the same database for their bookings. Multiple branches of the same theatre may share the same database, but different theatres use different database.

5.1.2 Cashier :

1. The cashier had one functionality. The cashier books tickets for the customer. The cashier selects a movie, and based on that, a showing is selected for a time and date. Then a seating chart is brought up and a seat is chosen. Then a unique ticket ID is generated and the data is stored in the database.
2. All data is stored on a common database. All systems in the theatre (cashier’s terminals and manager’s computer) access the same database for their bookings. Multiple branches of the same theatre may share the same database, but different theatres use different database.

5.2 NON-FUNCTIONAL REQUIREMENTS:

5.2.1 PERFORMANCE AND SCALABILITY.

- How fast does the system return results? How much will this performance change with higher workloads?

Performance Deals with the measure of the system's response time under different load conditions.

Example of performance requirements:

- The landing page supporting 5,000 users per hour must provide 6 second or less response time in a Chrome desktop browser, including the rendering of text and images and over an LTE connection.

Scalability assesses the highest workloads under which the system will still meet the performance requirements.

Example of scalability requirements:

- The system must be scalable enough to support 1,000,000 visits at the same time while maintaining optimal performance

5.2.2 PORTABILITY AND COMPATIBILITY.

- Which hardware, operating systems, and browsers, along with their versions does the software run on? Does it conflict with other applications and processes within these environments?

Portability determines how a system or its element can be launched within one environment or another

Example of portability requirements:

- A program running on Windows 10 must be able to run on Windows 11 without any change in its behavior and performance

Compatibility, as an additional aspect of portability, defines how a system can coexist with another system in the same environment.

Example of compatibility requirements:

- The iOS application must support iPhone devices running on OS versions,3.63.3,3.4

5.2.3 RELIABILITY,MAINTAINABILITY AND AVAILABILITY.

• How often does the system experience critical failures? How much time does it take to fix the issue when it arises? And how is user availability time compared to downtime

Reliability specifies how likely the system or its element would run without a failure for a given periodof time under predefined conditions.

Example of reliability requirements:

• The system must perform without failure in 95 percent of use cases during a month.

Maintainability defines the time required for a solution or its component to be fixed, changed to increase performance or other qualities, or adapted to a changing environment.

Example of maintainability requirements:

• The mean time to restore the system (MTTRS) following a system failure must not be greater than 10 minutes. MTTRS includes all corrective maintenance time and delay time

Availability describes how likely the system is accessible to a user at a given point in time.

Example of availability requirements:

• The web dashboard must be available to US users 99.98 percent of the time every month during business hours EST.

5.2.4 SECURITY.

• How well are the system and its data protected against attacks?

Security is a non-functional requirement assuring all data inside the system or its part will be protected against malware attacks or unauthorized access

Example of security requirement:

• The user password saved in hash form in database that protect from attacker

5.2.5 USABILITY.

- How easy is it for a customer to use the system?

Usability indicates how effectively and easy user can learn and use a system

Example of Usability requirement:

- User easily navigate by simplicity and easy user interface of our website

Chapter 6
Software Design

CHAPTER 6
SOFTWARE DESIGN

6.1 USE CASE DIAGRAM

The purpose of use case diagram is to capture the dynamic aspect of a system. However, this definition is too generic to describe the purpose, as other four diagrams (activity, sequence, collaboration, and State chart) also have the same purpose. We will look into some specific purpose, which will distinguish it from other four diagrams.

Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. Hence, when a system is analyzed to gather its functionalities, use cases are prepared and actors are identified.

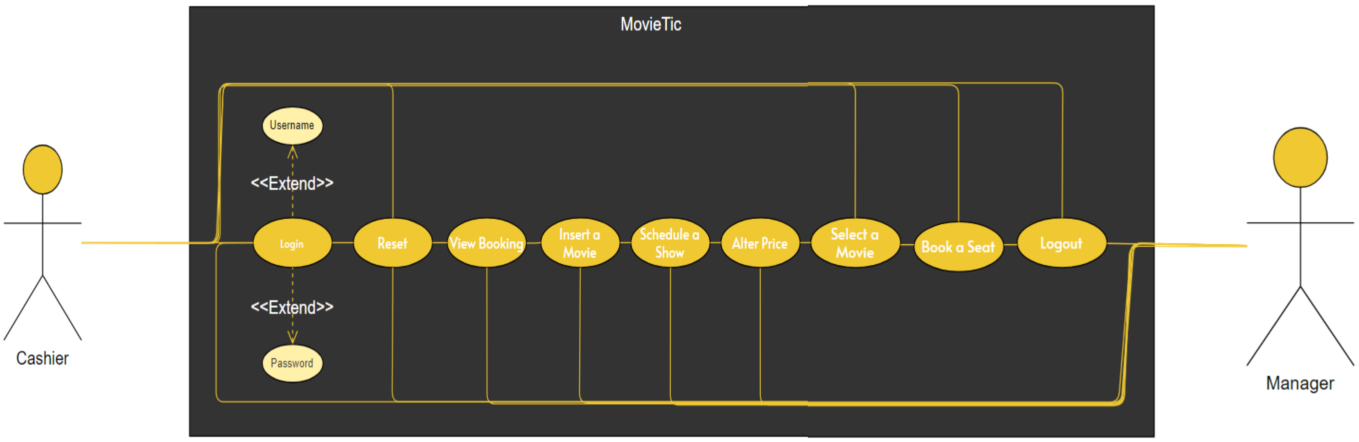


Figure 6.1 : MovieTic Use Case Diagram

6.2 ER DIAGRAM

ER Diagram stands for Entity Relationship Diagram, also known as ERD is a diagram that displays the relationship of entity sets stored in a database. In other words, ER diagrams help to explain the logical structure of databases. ER diagrams are created based on three basic concepts: entities, attributes and relationships.

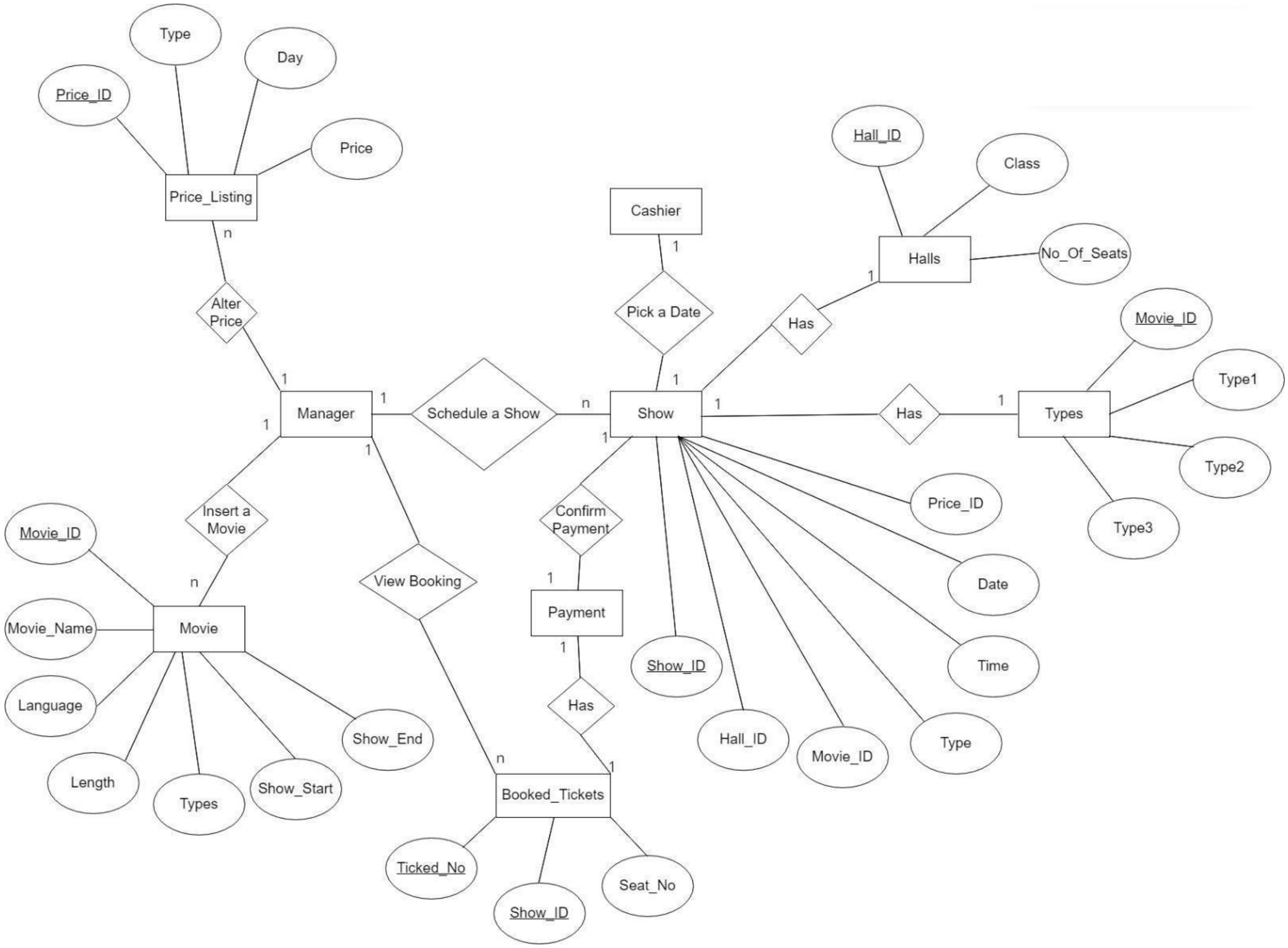


Figure 6.2 : MovieTic ER Diagram

Chapter 7
Output Screens

CHAPTER 7
OUTPUT SCREEN

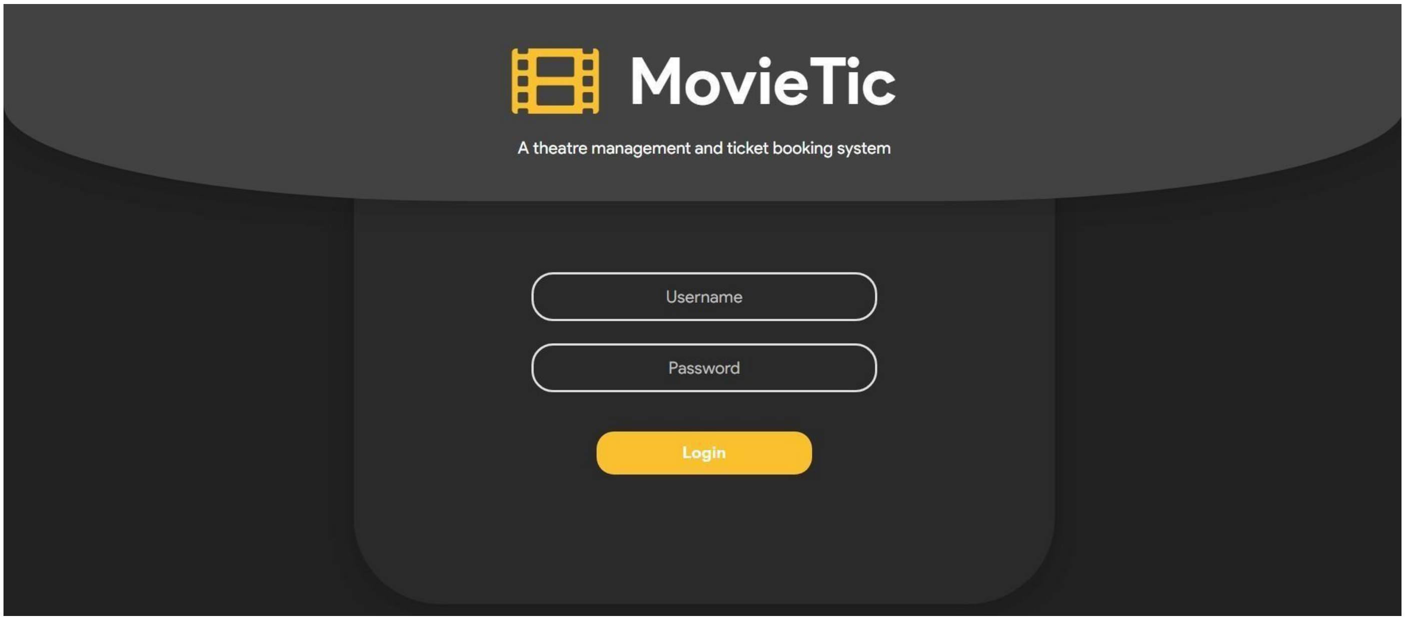


Figure 7.1 : Login page

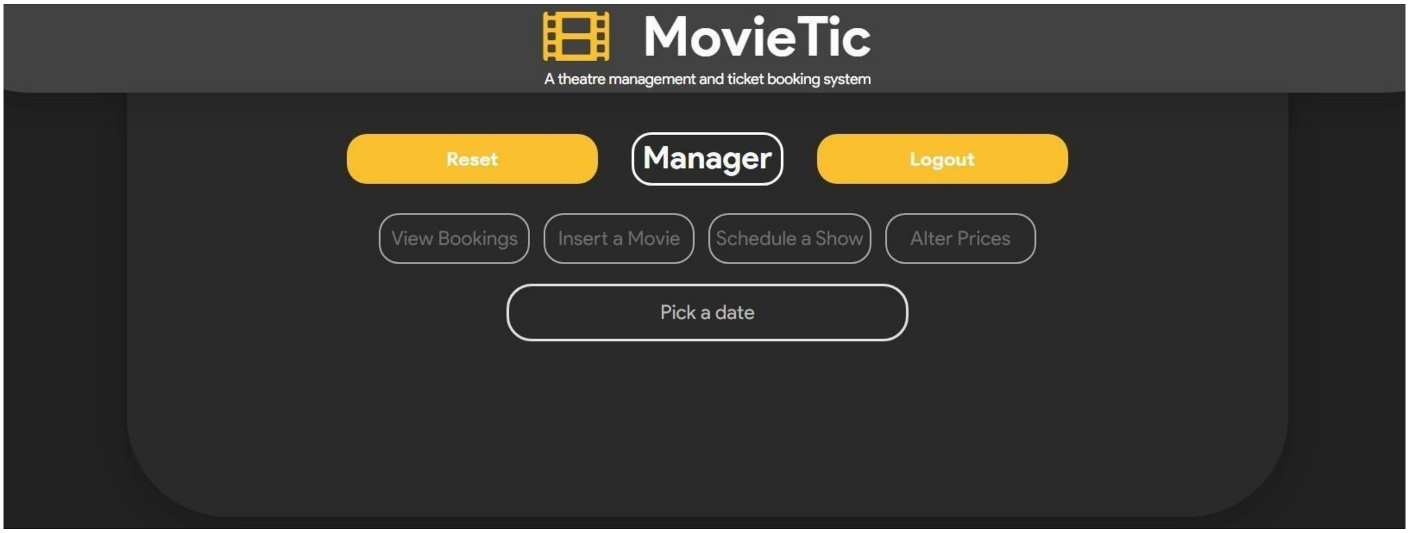
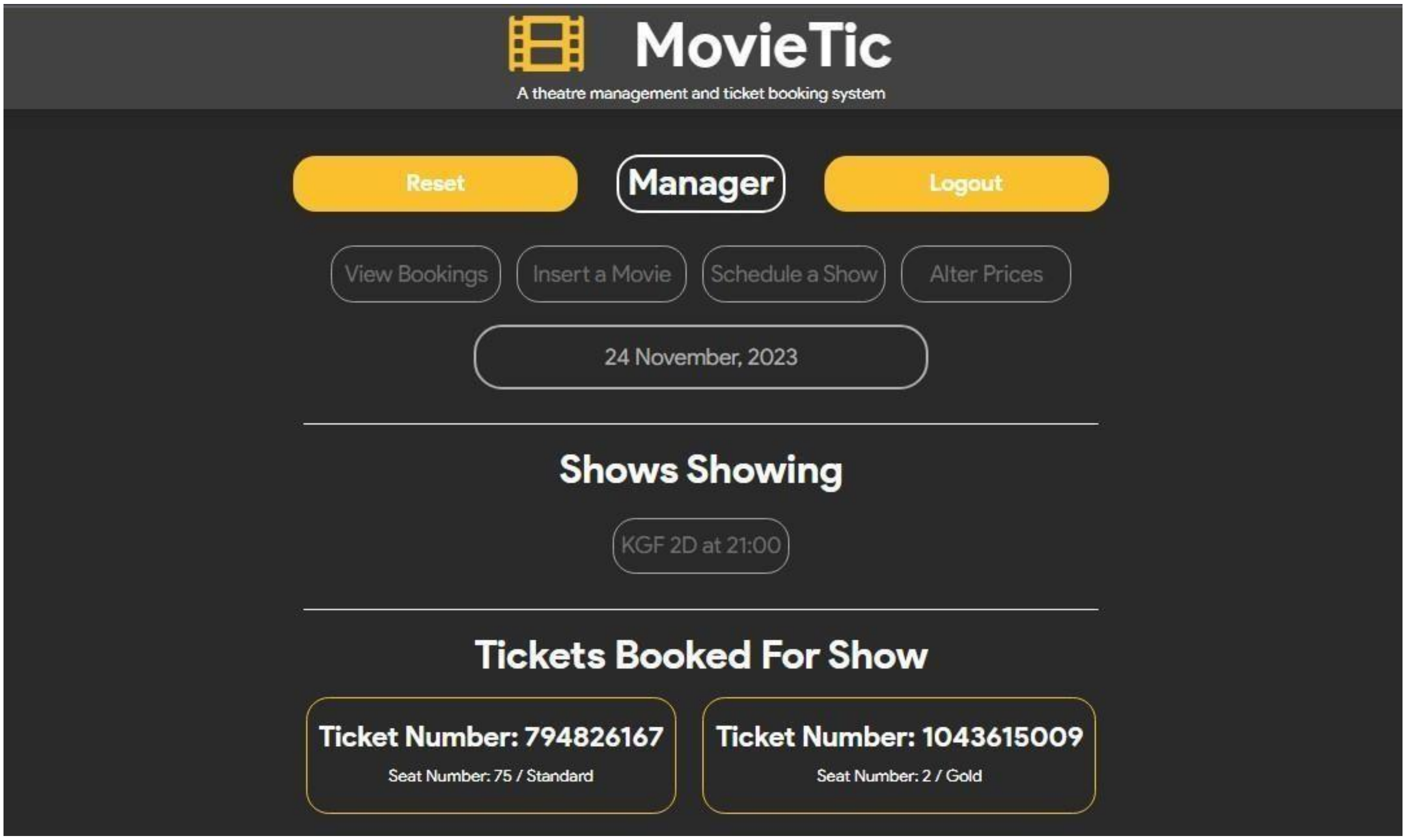


Figure 7.2 : Manager page



Shows Showing

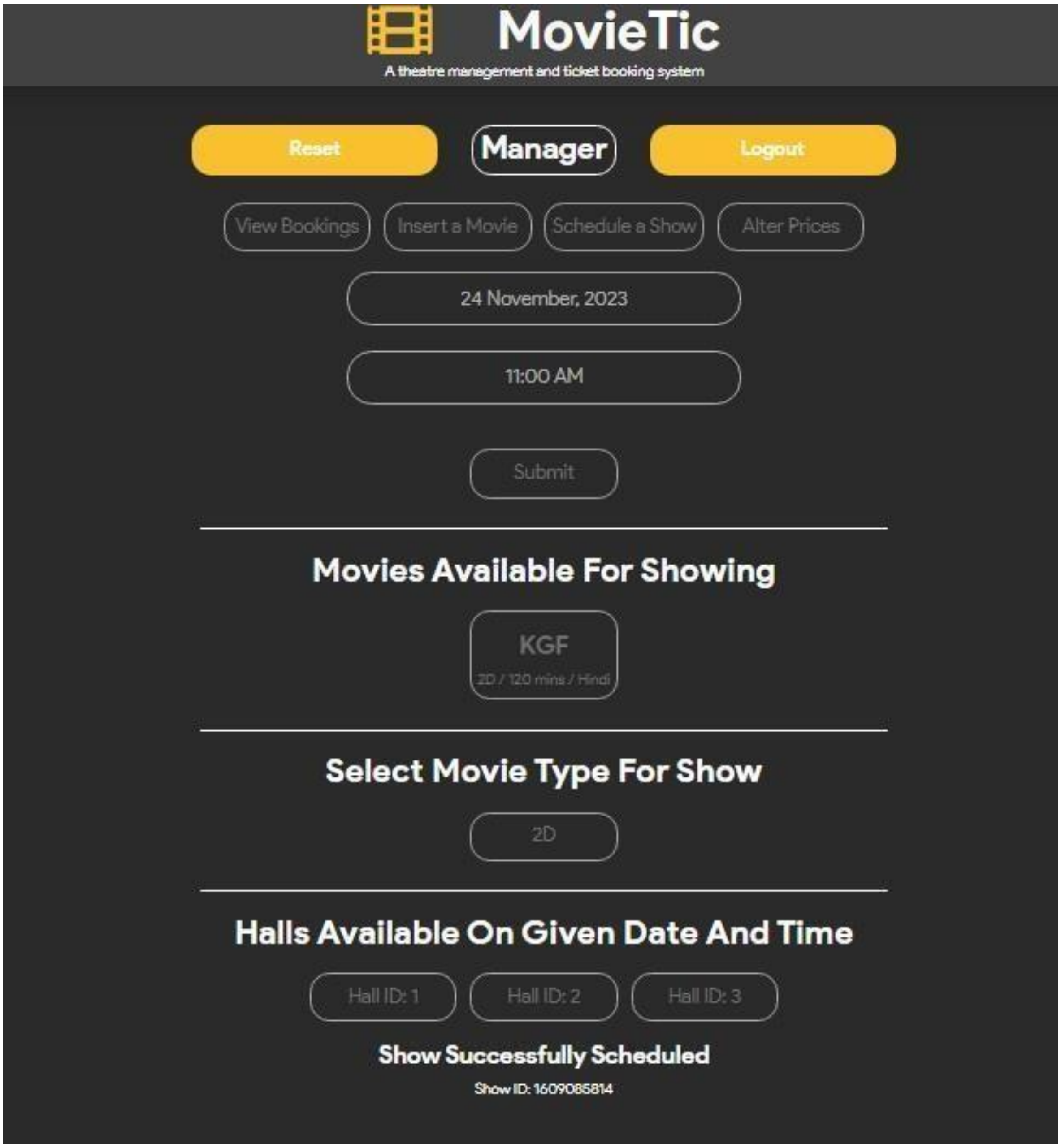
KGF 2D at 21:00

Tickets Booked For Show

Ticket Number: 794826167
Seat Number: 75 / Standard

Ticket Number: 1043615009
Seat Number: 2 / Gold

Figure 7.2.1 : Manager page (View Booking)



24 November, 2023

11:00 AM

Movies Available For Showing

KGF
2D / 120 mins / Hindi

Select Movie Type For Show

2D

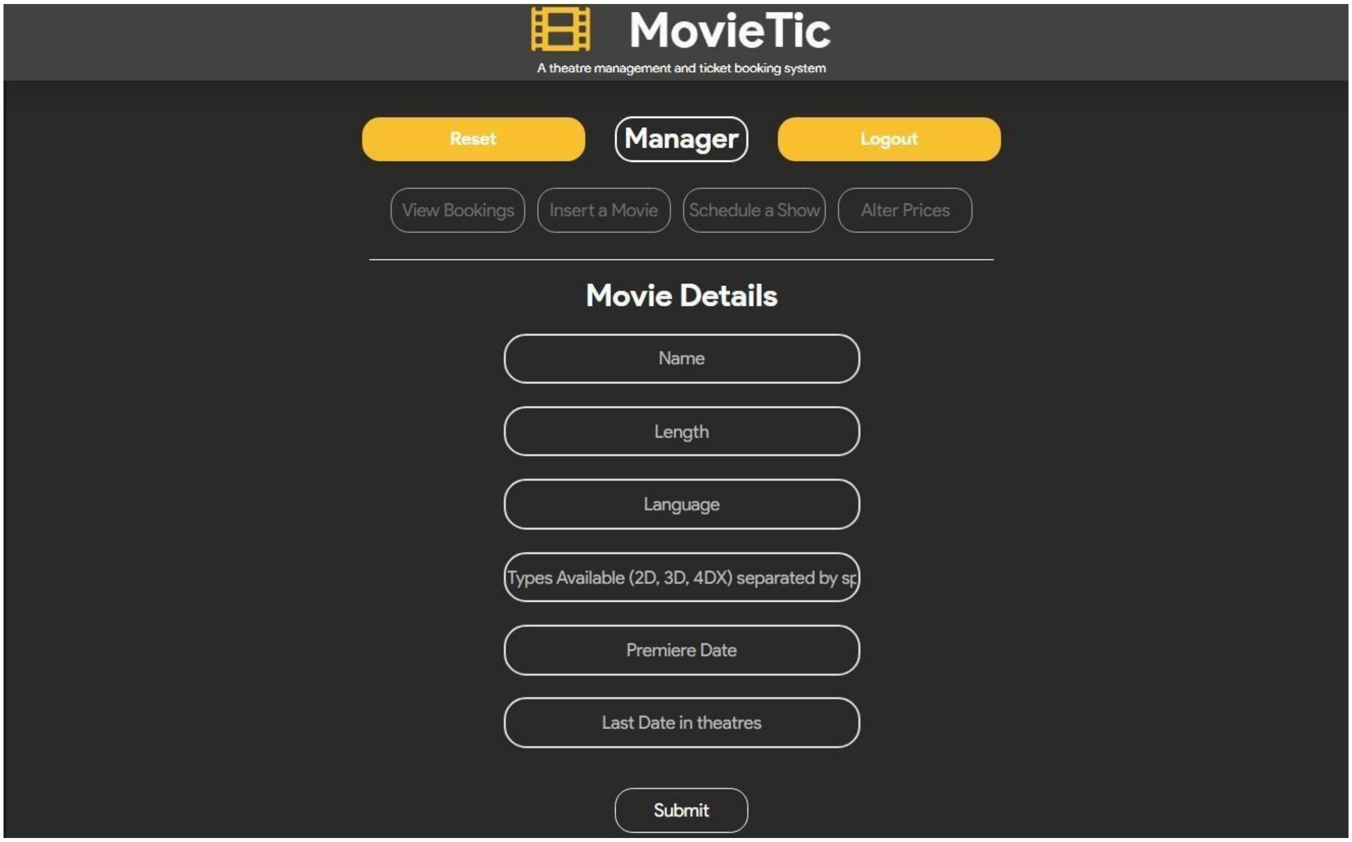
Halls Available On Given Date And Time

Hall ID: 1
Hall ID: 2
Hall ID: 3

Show Successfully Scheduled

Show ID: 1609085814

Figure 7.2.2 : Manager page (Schedule a Show)



Movie Details

Name

Length

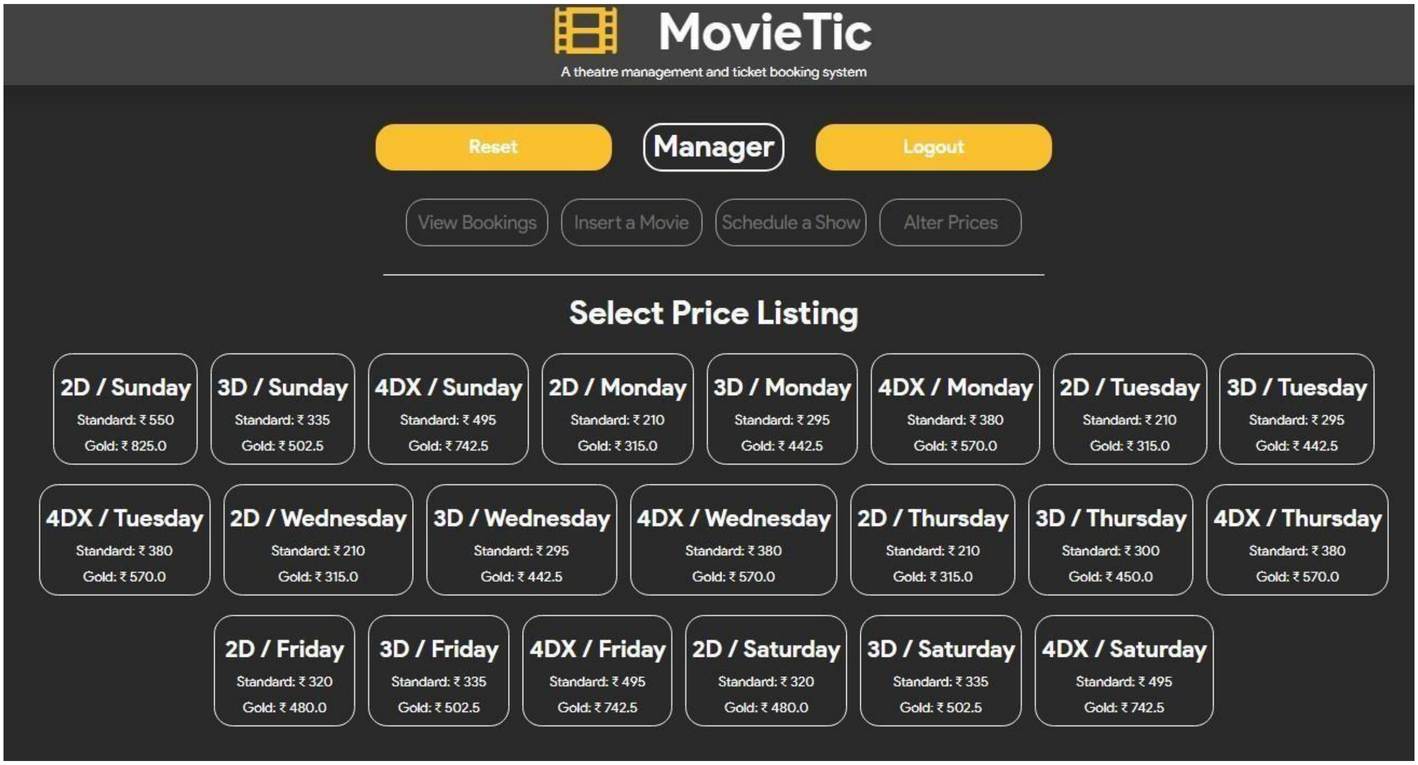
Language

Types Available (2D, 3D, 4DX) separated by sp

Premiere Date

Last Date in theatres

Figure 7.2.3 Manager page (Insert a Movie)



Select Price Listing

2D / Sunday
Standard: ₹ 550
Gold: ₹ 825.0

3D / Sunday
Standard: ₹ 335
Gold: ₹ 502.5

4DX / Sunday
Standard: ₹ 495
Gold: ₹ 742.5

2D / Monday
Standard: ₹ 210
Gold: ₹ 315.0

3D / Monday
Standard: ₹ 295
Gold: ₹ 442.5

4DX / Monday
Standard: ₹ 380
Gold: ₹ 570.0

2D / Tuesday
Standard: ₹ 210
Gold: ₹ 315.0

3D / Tuesday
Standard: ₹ 295
Gold: ₹ 442.5

4DX / Tuesday
Standard: ₹ 380
Gold: ₹ 570.0

2D / Wednesday
Standard: ₹ 210
Gold: ₹ 315.0

3D / Wednesday
Standard: ₹ 295
Gold: ₹ 442.5

4DX / Wednesday
Standard: ₹ 380
Gold: ₹ 570.0

2D / Thursday
Standard: ₹ 210
Gold: ₹ 315.0

3D / Thursday
Standard: ₹ 300
Gold: ₹ 450.0

4DX / Thursday
Standard: ₹ 380
Gold: ₹ 570.0

2D / Friday
Standard: ₹ 320
Gold: ₹ 480.0

3D / Friday
Standard: ₹ 335
Gold: ₹ 502.5

4DX / Friday
Standard: ₹ 495
Gold: ₹ 742.5

2D / Saturday
Standard: ₹ 320
Gold: ₹ 480.0

3D / Saturday
Standard: ₹ 335
Gold: ₹ 502.5

4DX / Saturday
Standard: ₹ 495
Gold: ₹ 742.5

Figure 7.2.4 : Manager page (Alter Prices)

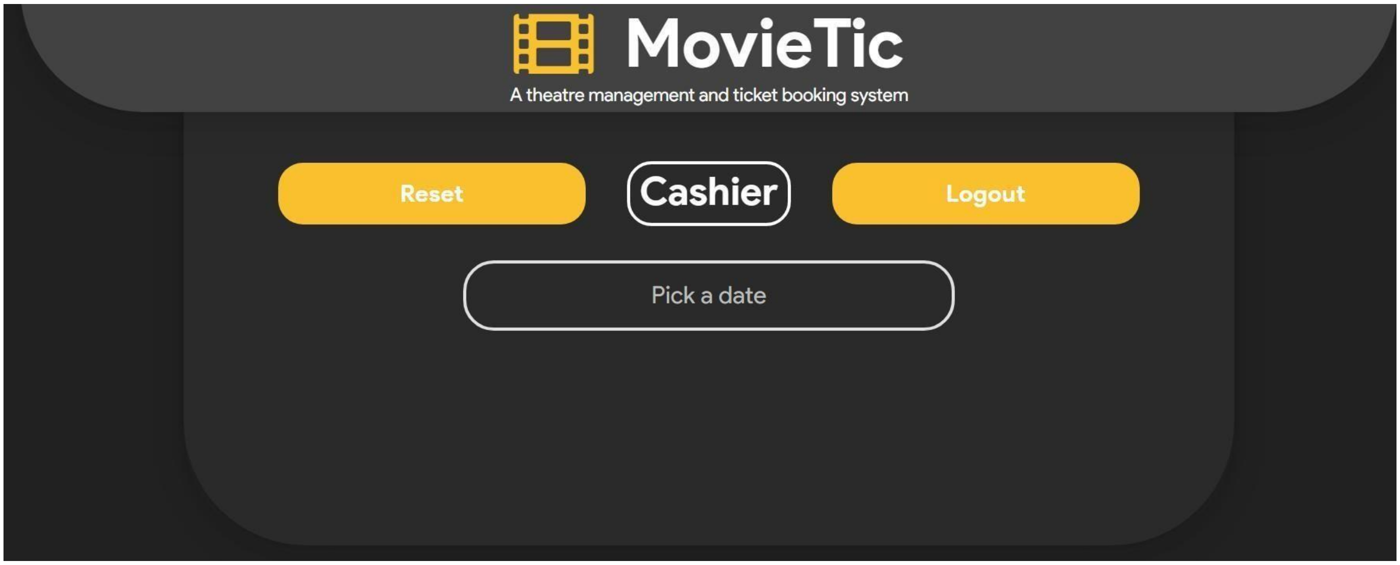
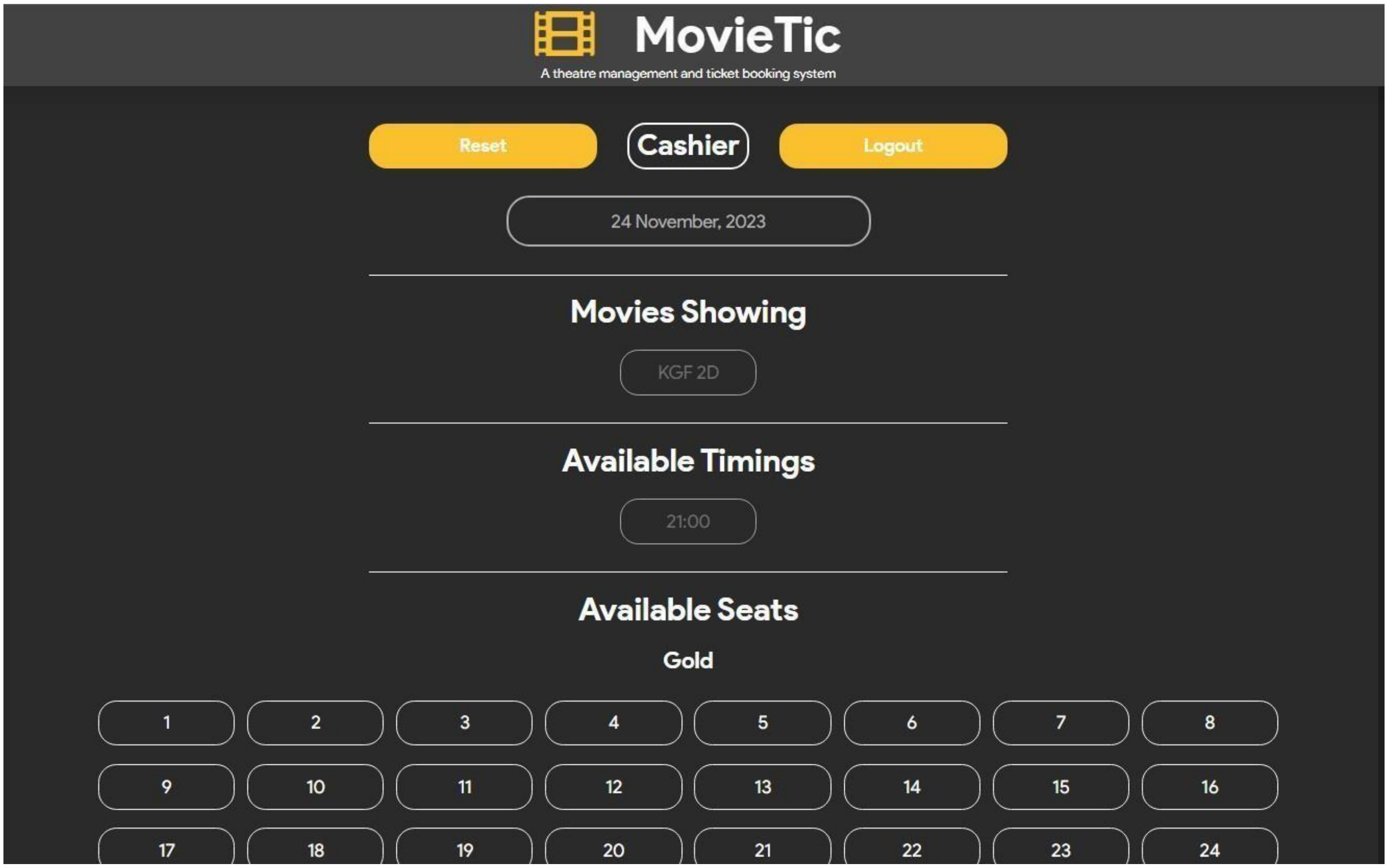


Figure 7.3 : Cashier page



24 November, 2023

KGF 2D

21:00

Figure 7.3.1 : Cashier page (Pick a date)

Chapter 8
Deployment

CHAPTER 8
DEPLOYMENT

8.1 Introduction (VS CODE IDE) :

Visual Studio Code is the most popular code editor and the IDEs provided by Microsoft for writing different programs and languages. It allows the users to develop new code bases for their applications and allow them to successfully optimize them and debug them properly. It is a very user-friendly code editor and it is supported on all the different types of operating systems like Windows, macOS, and Linux. It has the support for all the languages like C, C++, Java, Python, JavaScript, React, Node JS, etc.

It is the most popular code editor in India also. It allows users with different types of in-app installed extensions for the different types of their supported languages. It allows the programmers to write the code with ease with the help of these extensions. Also, Visual Studio Code has a great vibrant software UI with amazing night mode features. It suggests auto-complete code to the users which suggests the users complete their code with full ease.

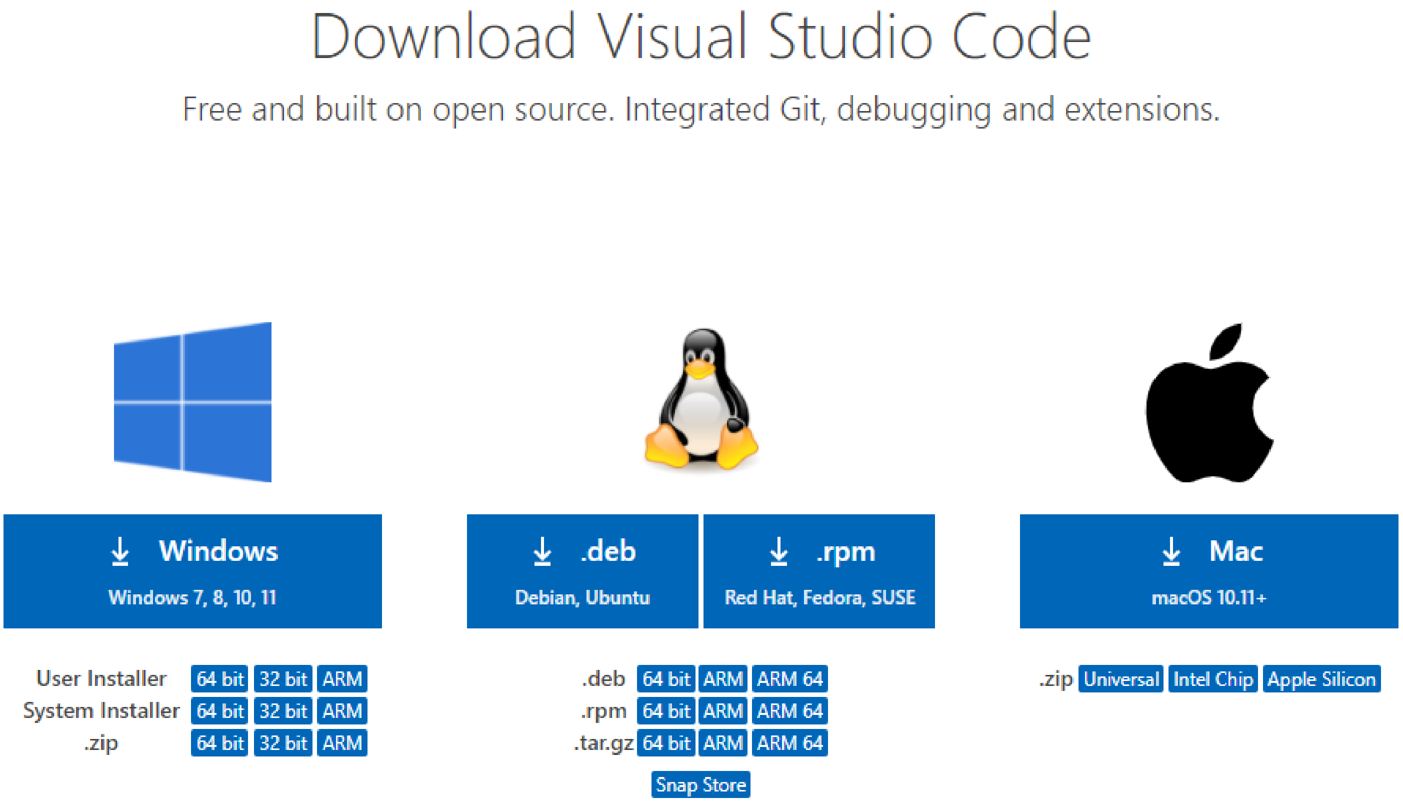
8.2 Installation (VS CODE IDE) :

Follow the below steps to install Visual Studio Code on Windows:

Step 1 : Visit the official website of the Visual Studio Code using any web browser like Google Chrome, Microsoft Edge, etc.

Step 2 : Press the “Download for Windows” button on the website to start the download of the Visual Studio Code Application.

Step 3 : When the download finishes, then the Visual Studio Code icon appears in the downloads folder.



Download Visual Studio Code

Free and built on open source. Integrated Git, debugging and extensions.

Figure 8.1 : Download (VS CODE IDE)

Step 4 : Click on the installer icon to start the installation process of the Visual Studio Code.

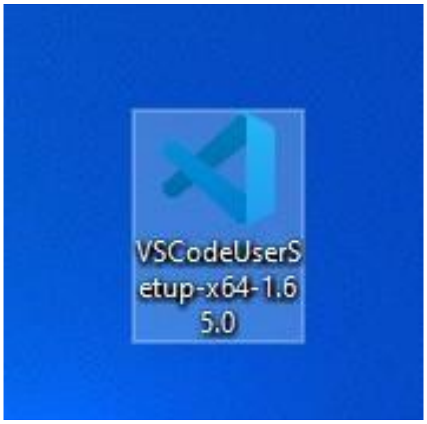
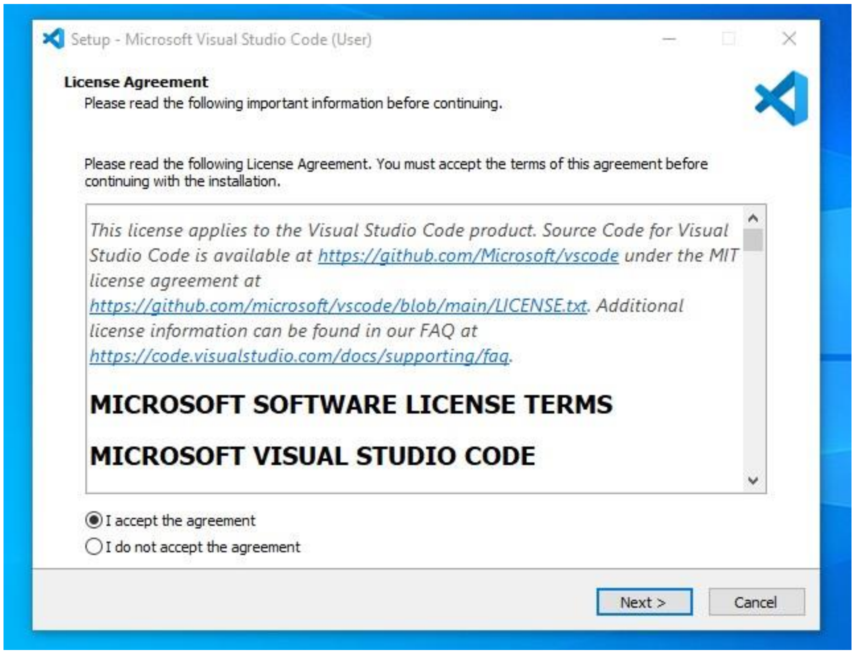


Figure 8.2 : Installer Icon (VS CODE IDE)

Step 5 : After the Installer opens, it will ask you for accepting the terms and conditions of the Visual Studio Code. Click on I accept the agreement and then click the Next button.



License Agreement

Please read the following important information before continuing.

Please read the following License Agreement. You must accept the terms of this agreement before continuing with the installation.

This license applies to the Visual Studio Code product. Source Code for Visual Studio Code is available at https://github.com/Microsoft/vscode under the MIT license agreement at https://github.com/microsoft/vscode/blob/main/LICENSE.txt. Additional license information can be found in our FAQ at https://code.visualstudio.com/docs/supporting/faq.

MICROSOFT SOFTWARE LICENSE TERMS

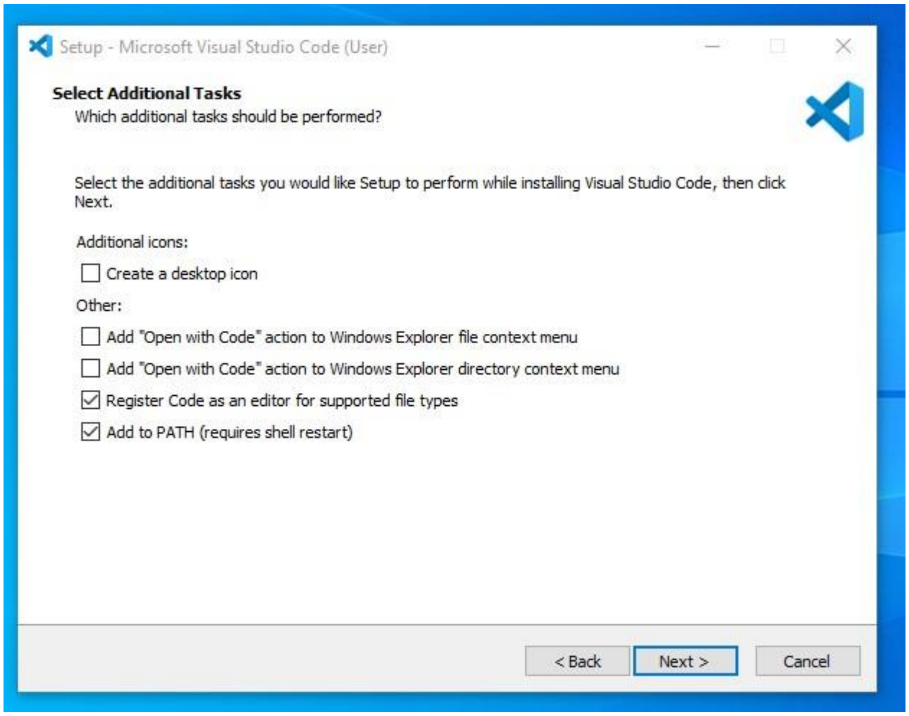
MICROSOFT VISUAL STUDIO CODE

● I accept the agreement

○ I do not accept the agreement

Figure 8.3 : License Agreement (VS CODE IDE)

Step 6 : Choose the location data for running the Visual Studio Code. It will then ask you for browsing the location. Then click on Next button.



Select Additional Tasks

Which additional tasks should be performed?

Select the additional tasks you would like Setup to perform while installing Visual Studio Code, then click Next.

Additional icons:

□ Create a desktop icon

Other:

□ Add "Open with Code" action to Windows Explorer file context menu

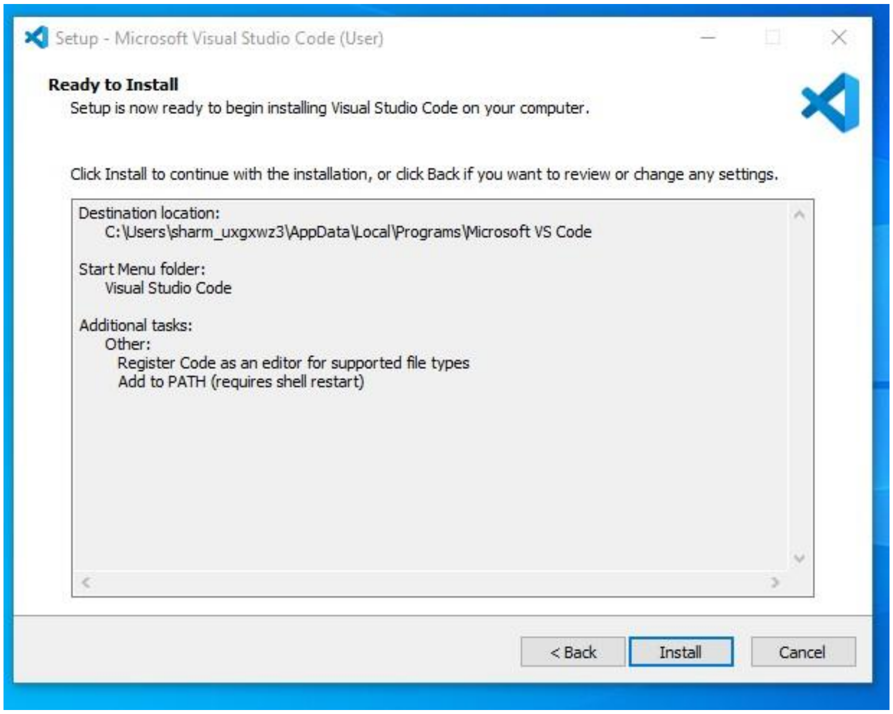
□ Add "Open with Code" action to Windows Explorer directory context menu

☑ Register Code as an editor for supported file types

☑ Add to PATH (requires shell restart)

Figure 8.4 : Additional Tasks (VS CODE IDE)

Step 7 : Then it will ask for beginning the installing setup. Click on the Install button.



Setup - Microsoft Visual Studio Code (User)

Ready to Install

Setup is now ready to begin installing Visual Studio Code on your computer.

Click Install to continue with the installation, or click Back if you want to review or change any settings.

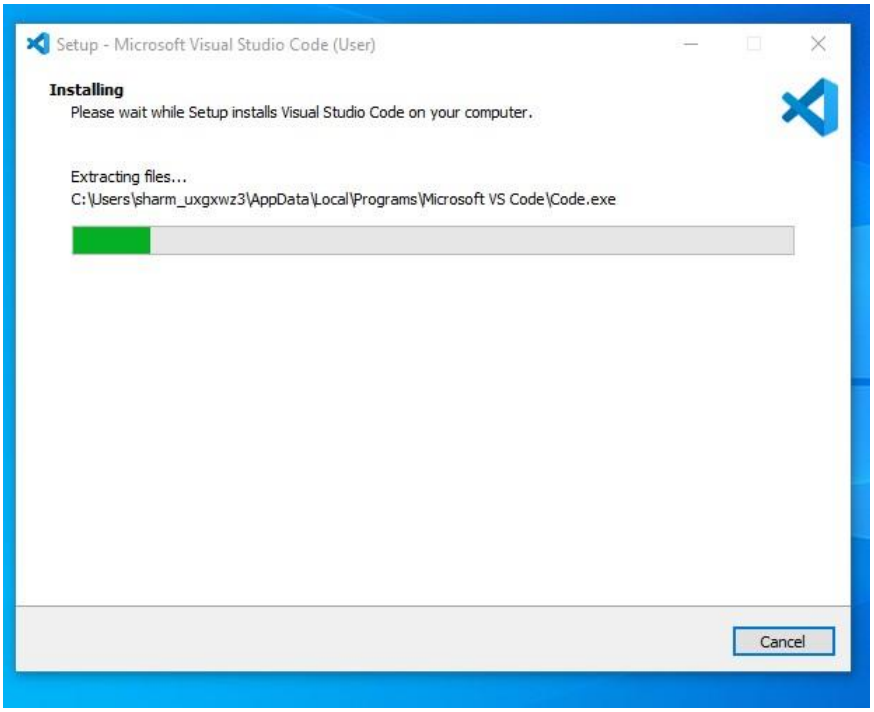
Destination location:
C:\Users\sharm_uxgxwz3\AppData\Local\Programs\Microsoft VS Code

Start Menu folder:
Visual Studio Code

Additional tasks:
Other:
Register Code as an editor for supported file types
Add to PATH (requires shell restart)

Figure 8.5 : Ready To Install (VS CODE IDE)

Step 8 : After clicking on Install, it will take about 1 minute to install the Visual Studio Code on your device.



Setup - Microsoft Visual Studio Code (User)

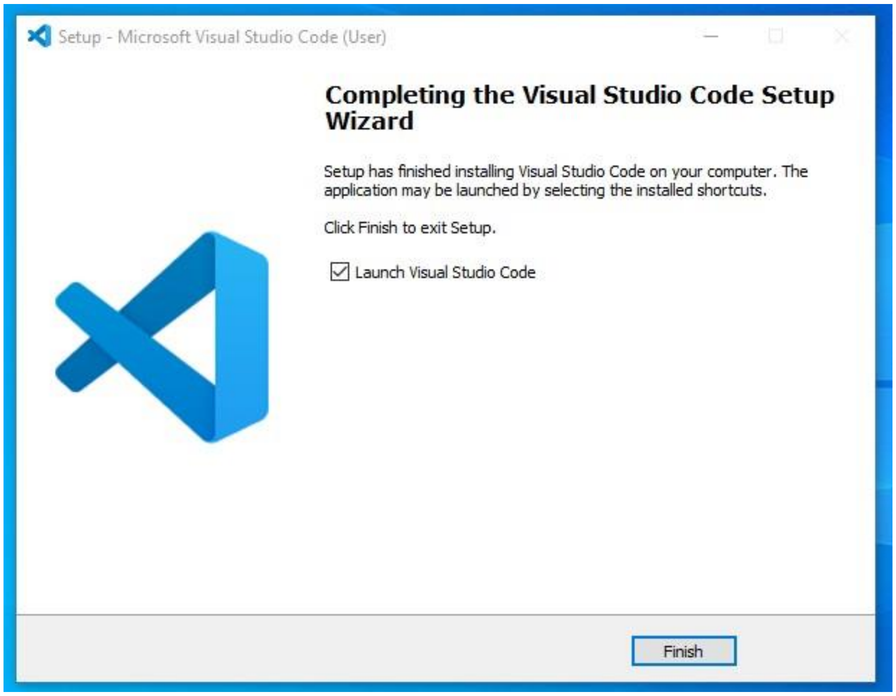
Installing

Please wait while Setup installs Visual Studio Code on your computer.

Extracting files...
C:\Users\sharm_uxgxwz3\AppData\Local\Programs\Microsoft VS Code\Code.exe

Figure 8.6 : Installing (VS CODE IDE)

Step 9 : After the Installation setup for Visual Studio Code is finished, it will show a window like this below. Tick the “Launch Visual Studio Code” checkbox and then click Next.



Completing the Visual Studio Code Setup Wizard

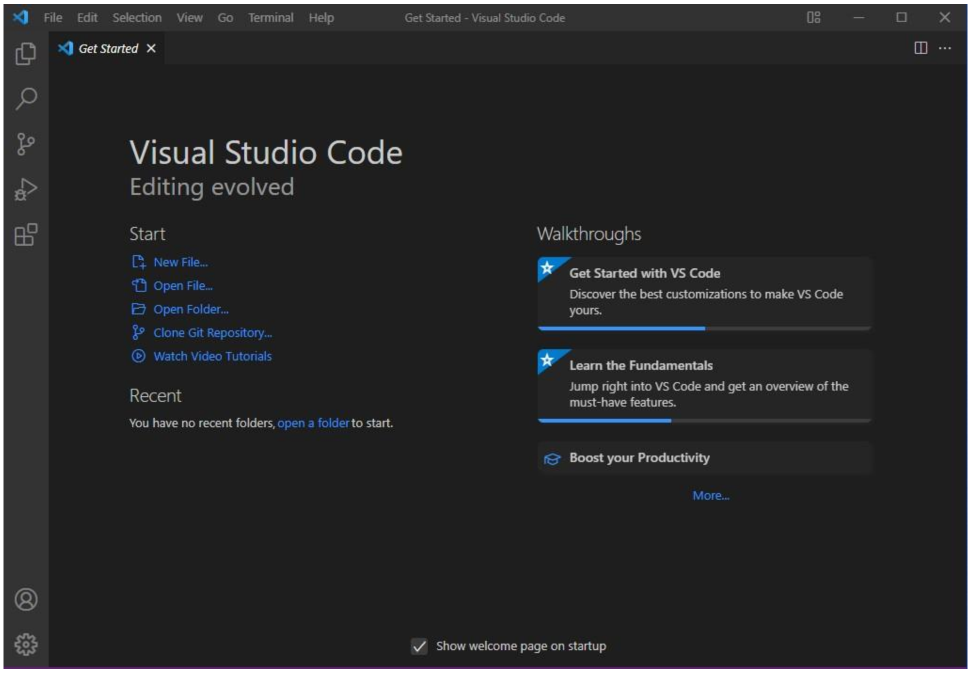
Setup has finished installing Visual Studio Code on your computer. The application may be launched by selecting the installed shortcuts.

Click Finish to exit Setup.

☑ Launch Visual Studio Code

Figure 8.7 : Finish(VS CODE IDE)

Step 10 : After the previous step, the Visual Studio Code window opens successfully. Now you can create a new file in the Visual Studio Code window and choose a language of yours to begin your programming journey!



You have no recent folders, open a folder to start.

Get Started with VS Code: Discover the best customizations to make VS Code yours.

Learn the Fundamentals: Jump right into VS Code and get an overview of the must-have features.

☑ Show welcome page on startup

Figure 8.8 : VS CODE IDE

8.3 Introduction (XAMMP) :

XAMPP is the most popular software package which is used to set up a PHP development environment for web services by providing all the required software components. During the process of software deployment, most of the web servers use almost similar components, so use of XAMPP provides easy transition from local server to live server. XAMPP is a AMP stack which stands for Cross platform, Apache, MySQL, PHP, perl with some additional administrative software tools such as PHPMyAdmin (for database access), FileZilla FTP server, Mercury mail server and JSP Tomcat server.

Other commonly known software packages like XAMPP are WAMP, LAMP, and others.

The XAMPP server is used to test PHP pages. It works as local server. It contains a MySQL database to manage or save data on a local server.

Advantages of XAMPP:

- It is free and easy to use and easily available for Windows, Linux and Mac OS .
- It is a beginners friendly solution package for full stack web development.
- It is a open source software package which gives a easy installation experience.
- It is very simple and lightweight to create set up for development, testing and deployment.
- It is a time-saver and provides several ways for managing configuration changes.
- It handles many administrative tasks like checking the status and security.

Software components of XAMPP:

- Apache plays the role of processing the HTTP request. It is the actual default web server application. It is the most popular web servers maintained by Apache Software Foundation.

- MySQL The role of database management system in XAMPP is played by MySQL. It helps to store and manage collected data very efficiently. It is an open-source and most popular.

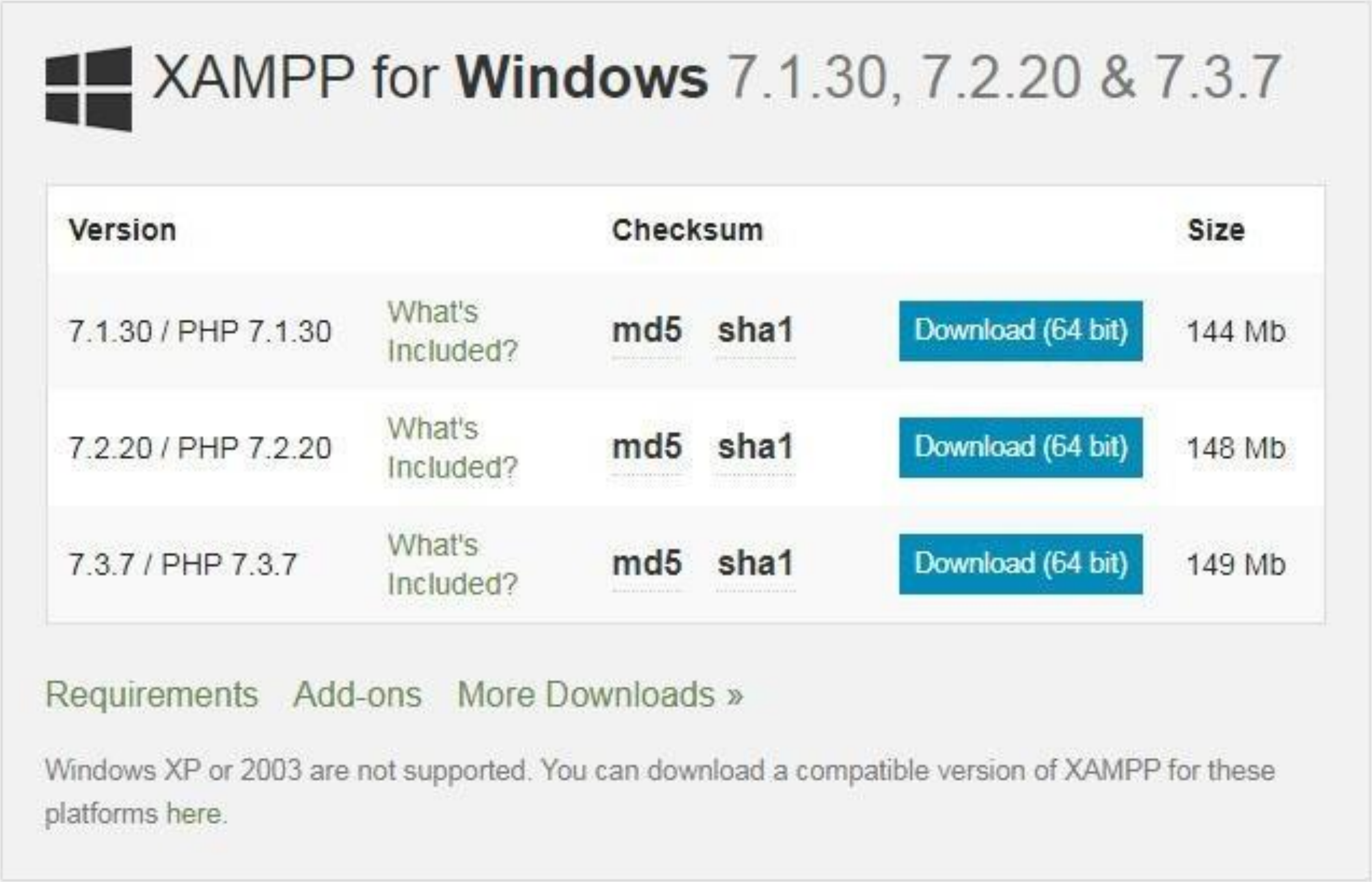
- PHP is the server-side scripting language which stand for Hypertext Preprocessor. It is embedded with HTML code which interacts with the webserver. It is an open-source and work well with MySQL and has become a common choice for web developers.

- Perl is the high-level programming language designed for text editing which serves purpose like web development and network programming.

8.4 Intallation (XAMMP) :

Steps to install XAMPP on Windows:

- In the web browser, visit Apache Friends and download XAMPP installer.



XAMPP for **Windows** 7.1.30, 7.2.20 & 7.3.7

Version		Checksum			Size
7.1.30 / PHP 7.1.30	What's Included?	md5	sha1	Download (64 bit)	144 Mb
7.2.20 / PHP 7.2.20	What's Included?	md5	sha1	Download (64 bit)	148 Mb
7.3.7 / PHP 7.3.7	What's Included?	md5	sha1	Download (64 bit)	149 Mb

Requirements Add-ons More Downloads »

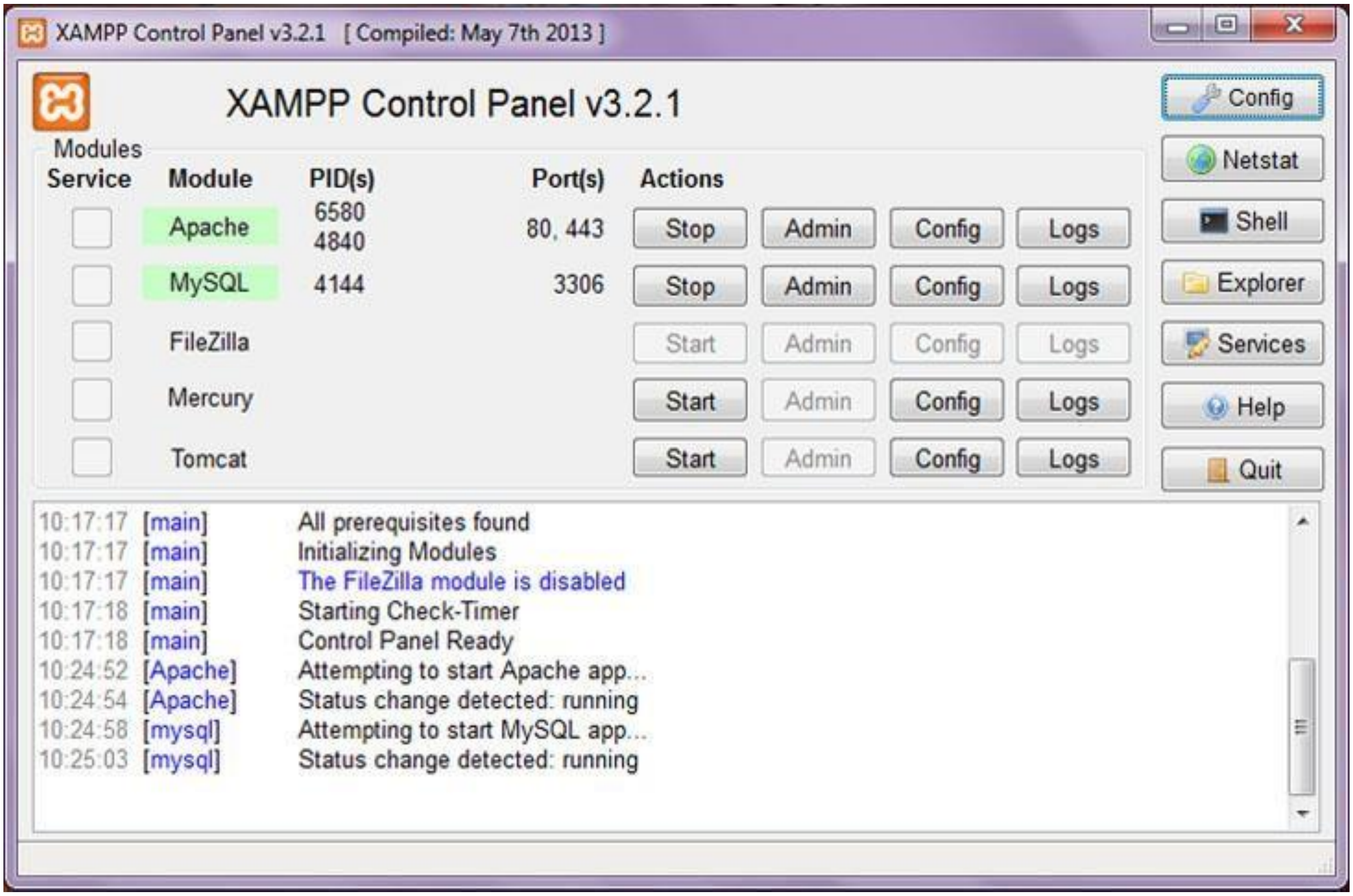
Windows XP or 2003 are not supported. You can download a compatible version of XAMPP for these platforms here.

Figure 8.9 : Download (XAMMP)

- During the installation process, select the required components like MySQL, FileZilla ftp server, PHP, phpMyAdmin or leave the default options and click the Next button.

Figure 8.10 : Installation process (XAMMP)

- Uncheck the Learn more about bitnami option and click Next button.
- Choose the root directory path to set up the htdocs folder for our applications. For example ‘C:\xampp’.
- Click the Allow access button to allow the XAMPP modules from the Windows firewall.
- After the installation process, click the Finish button of the XAMPP Setup wizard.
- Now the XAMPP icon is clearly visible on the right side of start menu. Show or Hide can be set by using the control panel by clicking on the icon.
- To start Apache and MySql, just click on the Start button on the control panel.


XAMPP Control Panel v3.2.1 [Compiled: May 7th 2013]
XAMPP Control Panel v3.2.1
Modules
Config
Netstat
Shell
Explorer
Services
Help
Quit

Service	Module	PID(s)	Port(s)	Actions
□	Apache	6580 4840	80, 443	Stop Admin Config Logs
□	MySQL	4144	3306	Stop Admin Config Logs
□	FileZilla			Start Admin Config Logs
□	Mercury			Start Admin Config Logs
□	Tomcat			Start Admin Config Logs

```
10:17:17  [main]     All prerequisites found
10:17:17  [main]     Initializing Modules
10:17:17  [main]     The FileZilla module is disabled
10:17:18  [main]     Starting Check-Timer
10:17:18  [main]     Control Panel Ready
10:24:52  [Apache]   Attempting to start Apache app...
10:24:54  [Apache]   Status change detected: running
10:24:58  [mysql]    Attempting to start MySQL app...
10:25:03  [mysql]    Status change detected: running
```

Figure 8.11 : Control Panel (XAMMP)

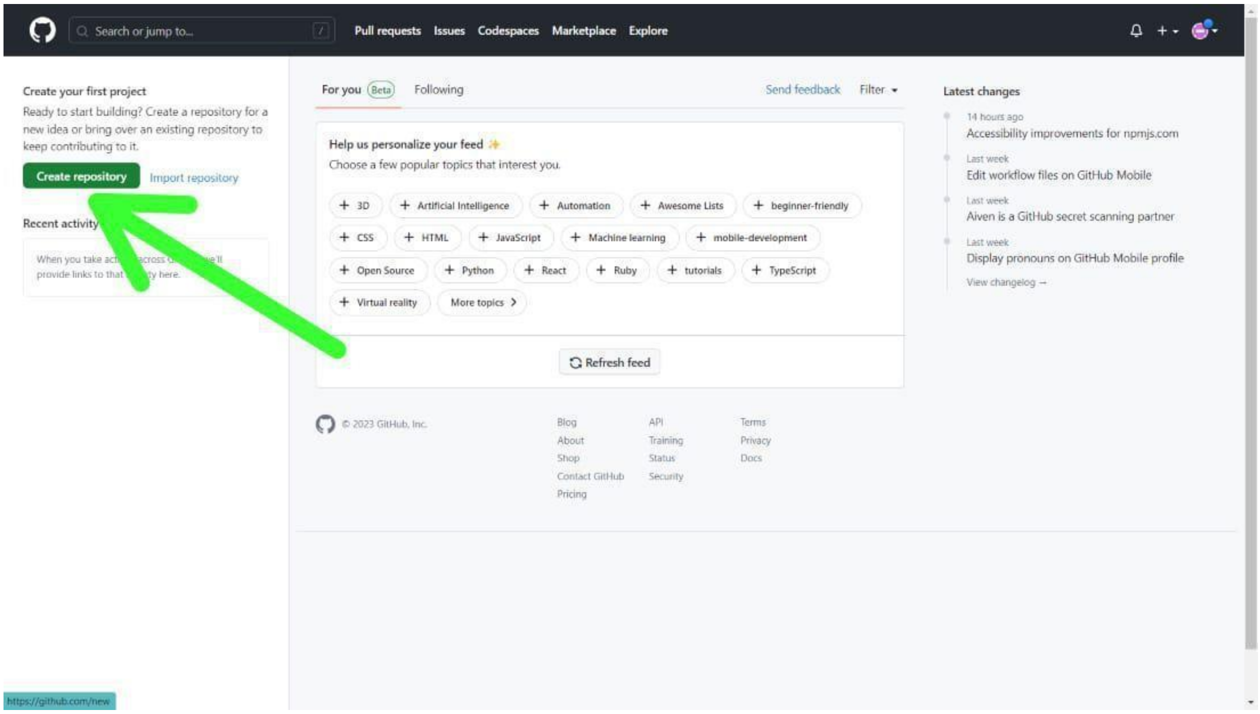
8.5 Introduction (GitHub) :

GitHub is an web based platform which hosts software development projects and uses Git for version management. Git is a distributed version control system that helps developers to work together on same software projects and keep track of changes made to their code by on another. GitHub offers a user-friendly interface, which is very collaborative tools, and more project management tools, GitHub will enhance the potential of the Git.

GitHub allows developers to create and manage the code in the repository in the remote location where others can aceses the code or Github is an collection repositories which contains the files of the project.

8.6 Hosting On GitHub :

Step 1 : You need to create a new repository to host your website. To create a new repository, click on the "Create repository" button on your left section after the new registration. You can also do it by clicking on the "new" button on your GitHub Dashboard. Give your repository a name that corresponds to your username. For example, if your website is called "username", name your repository "username.github.io".



Create your first project

Ready to start building? Create a repository for a new idea or bring over an existing repository to keep contributing to it.

Help us personalize your feed ✨

Choose a few popular topics that interest you.

Latest changes

14 hours ago
Accessibility improvements for npmjs.com

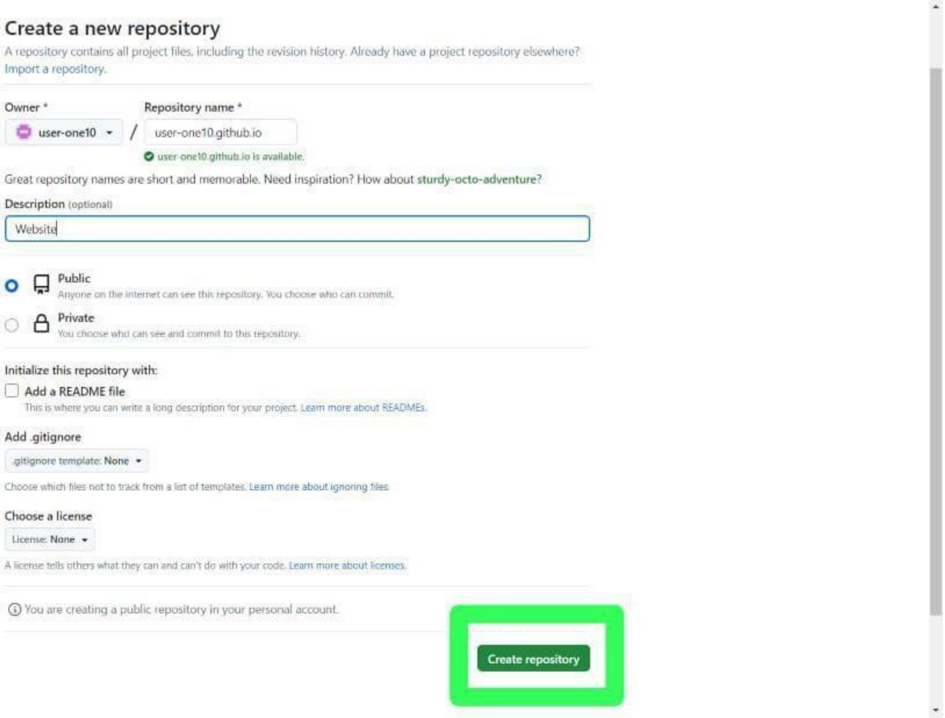
Last week
Edit workflow files on GitHub Mobile

Last week
Aiven is a GitHub secret scanning partner

Last week
Display pronouns on GitHub Mobile profile

Figure 8.12 : Create Repository (GitHub)

Give the description of the website you are creating and then click on “create repository” button.



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Owner * user-one10 / Repository name * user-one10.github.io

user-one10.github.io is available.

Great repository names are short and memorable. Need inspiration? How about sturdy-octo-adventure?

Description (optional): Website

● Public — Anyone on the internet can see this repository. You choose who can commit.

○ Private — You choose who can see and commit to this repository.

Initialize this repository with:

□ Add a README file — This is where you can write a long description for your project. Learn more about READMEs.

Add .gitignore — .gitignore template: None

Choose which files not to track from a list of templates. Learn more about ignoring files.

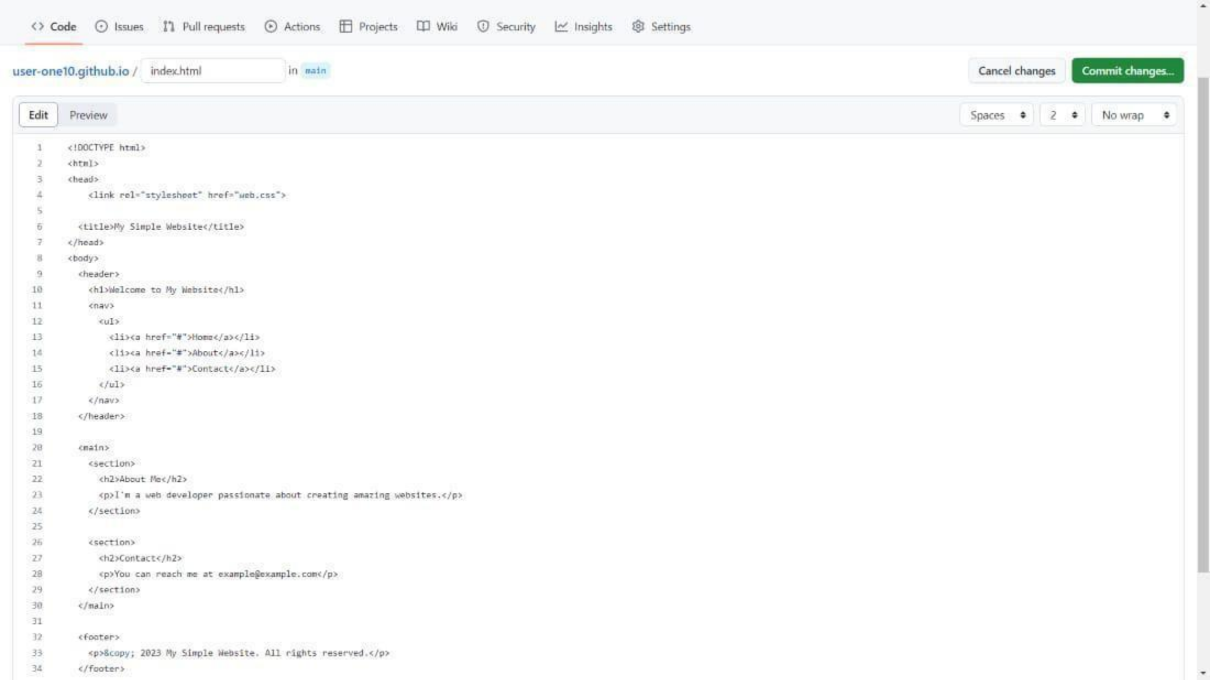
Choose a license — License: None

A license tells others what they can and can't do with your code. Learn more about licenses.

You are creating a public repository in your personal account.

Figure 8.13 : Give the description (GitHub)

Step 2 : Create an index.html file and a web.css file to create a website, you need an HTML file that contains the content of your website. You can create a new HTML file using a text editor like Visual Studio Code or Notepad. Save the file as “index.html” in your local repository.

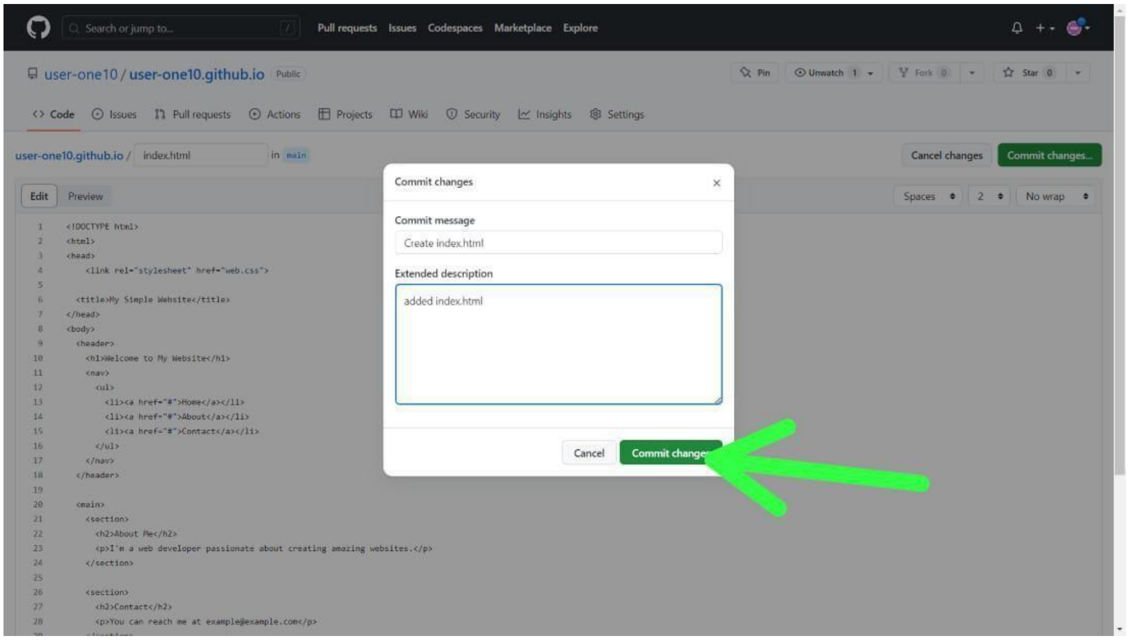


user-one10.github.io / index.html in main

```html
<!DOCTYPE html>
<html>
<head>
    <link rel="stylesheet" href="web.css">

  <title>My Simple Website</title>
</head>
<body>
  <header>
    <h1>Welcome to My Website</h1>
    <nav>
      <ul>
        <li><a href="#">Home</a></li>
        <li><a href="#">About</a></li>
        <li><a href="#">Contact</a></li>
      </ul>
    </nav>
  </header>

  <main>
    <section>
      <h2>About Me</h2>
      <p>I'm a web developer passionate about creating amazing websites.</p>
    </section>

    <section>
      <h2>Contact</h2>
      <p>You can reach me at example@example.com</p>
    </section>
  </main>

  <footer>
    <p>&copy; 2023 My Simple Website. All rights reserved.</p>
  </footer>
```

Figure 8.14 : Add HTML file (GitHub)

Add a description and “Commit changes” button so that your code gets added and saved on GitHub.



user-one10.github.io / index.html in main

```
<!DOCTYPE html>
<html>
<head>
    <link rel="stylesheet" href="web.css">

  <title>My Simple Website</title>
</head>
<body>
  <header>
    <h1>Welcome to My Website</h1>
    <nav>
      <ul>
        <li><a href="#">Home</a></li>
        <li><a href="#">About</a></li>
        <li><a href="#">Contact</a></li>
      </ul>
    </nav>
  </header>

  <main>
    <section>
      <h2>About Me</h2>
      <p>I'm a web developer passionate about creating amazing websites.</p>
    </section>

    <section>
      <h2>Contact</h2>
      <p>You can reach me at example@example.com</p>
```

Commit changes

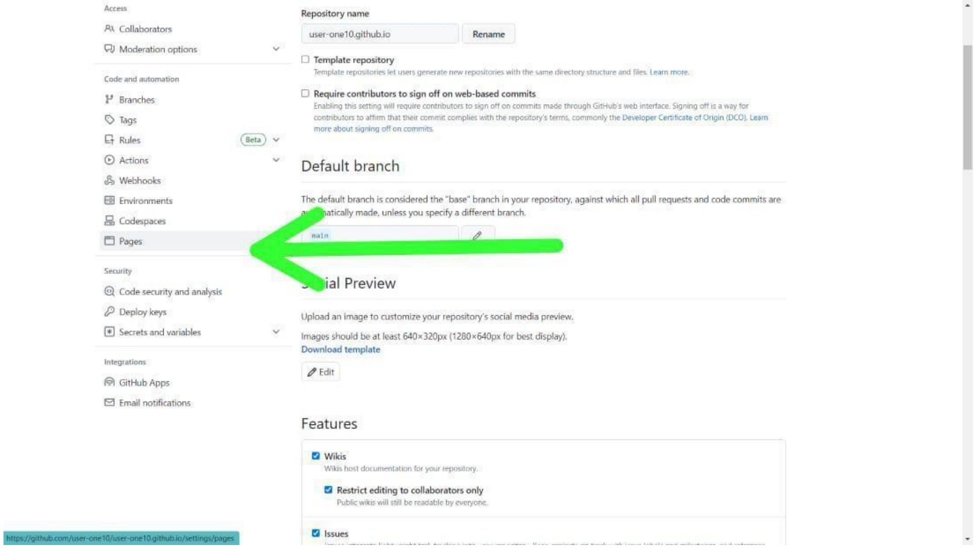
Commit message: Create index.html

Extended description: added index.html

Figure 8.15 : Commit changes (GitHub)

Add the CSS code for your website by doing the same process as above and then add a description and click on the “Commit changes” button.

Step 3 : Enable GitHub Pages After pushing your code to your repository, you need to enable GitHub Pages for your repository. To do this, navigate to your repository on GitHub and click on the “Settings” tab. Scroll down to the “Pages” section from the left section.



Repository name: user-one10.github.io

☐ Template repository
Template repositories let users generate new repositories with the same directory structure and files. Learn more.

☐ Require contributors to sign off on web-based commits
Enabling this setting will require contributors to sign off on commits made through GitHub's web interface. Signing off is a way for contributors to affirm that their commit complies with the repository's terms, commonly the Developer Certificate of Origin (DCO). Learn more about signing off on commits.

Default branch

The default branch is considered the "base" branch in your repository, against which all pull requests and code commits are automatically made, unless you specify a different branch.

main

Social Preview

Upload an image to customize your repository's social media preview.

Images should be at least 640×320px (1280×640px for best display).

Features

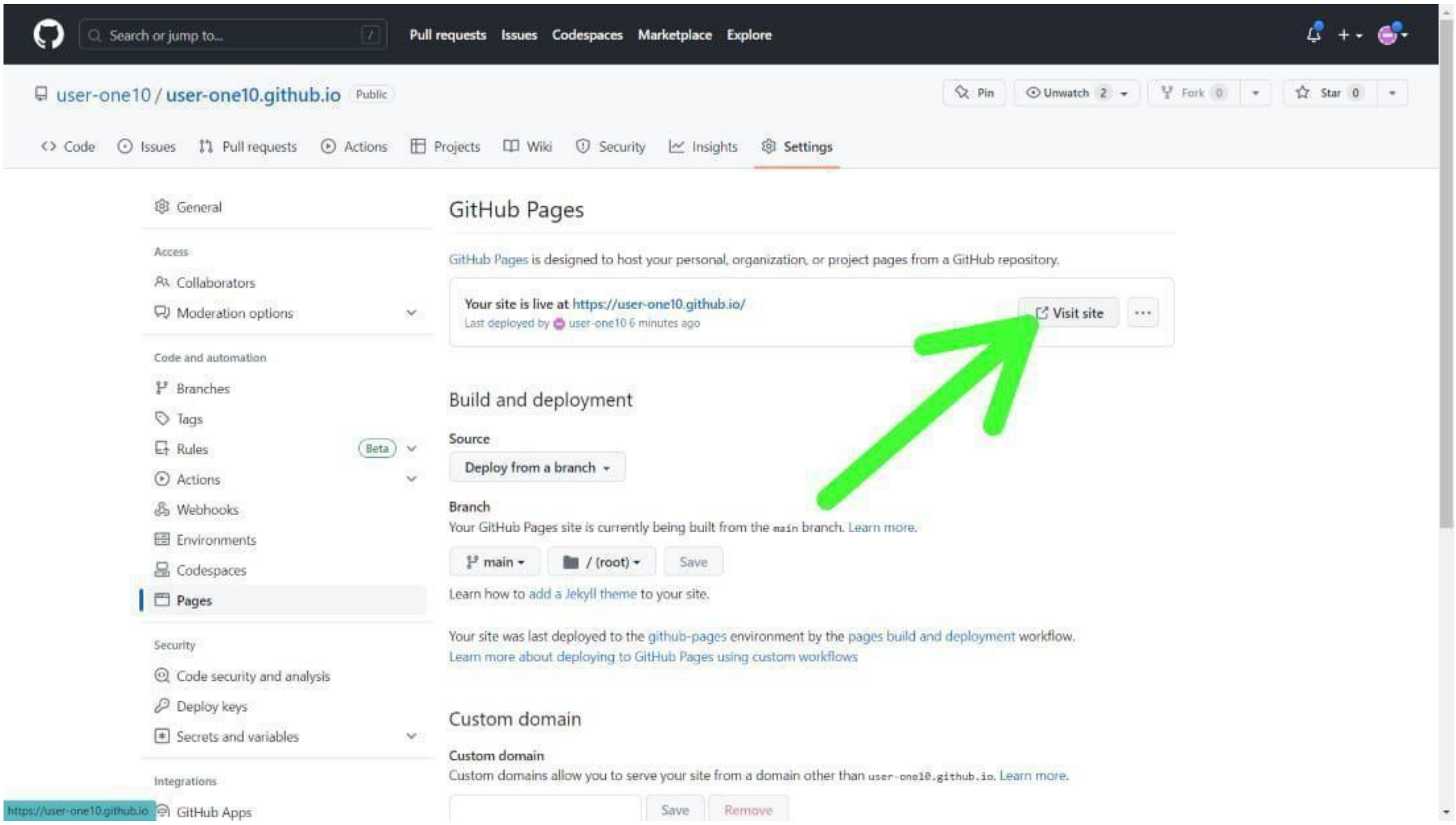
☑ Wikis
Wikis host documentation for your repository.

☑ Restrict editing to collaborators only
Public wikis will still be readable by everyone.

☑ Issues

Figure 8.16 : Pages (GitHub)

Step 4 : Visit your websiteOnce GitHub Pages are enabled for your repository, your website is now live! You can visit your website by navigating to https://your-username.github.io/ in your web browser. It might take a few moments for the changes to propagate and for your website to be accessible.



GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

Your site is live at https://user-one10.github.io/
Last deployed by user-one10 6 minutes ago

Build and deployment

Source
Deploy from a branch

Branch
Your GitHub Pages site is currently being built from the `main` branch. Learn more.

main / (root) Save

Learn how to add a Jekyll theme to your site.

Your site was last deployed to the github-pages environment by the pages build and deployment workflow.
Learn more about deploying to GitHub Pages using custom workflows

Custom domain

Custom domain
Custom domains allow you to serve your site from a domain other than `user-one10.github.io`. Learn more.

Figure 8.17 : Visit site (GitHub)

Chapter 9
Conclusion & Future Works

CHAPTER 9
CONCLUSION AND FUTURE WORK

9.1 Conclusion

As per the Requirements of the project, here is implemented a System that enables Theatre staff to fully minimize all aspects of a theatre movie management to an efficient and fast computer system. It allows movies, and shows to be added to the system, and tickets booked for shows. As such, need for bulky ledgers drops sharply, and all relevant is stored and can be printed as needed. It automatically determines prices and deletes old shows, movies and tickets reducing the amount of data to traverse for manager.
Thus, we have implemented a fully comprehensive and minimalistic efficient system for use by managers and cashiers without any additional training.

9.2 Limitations of the project

- This project only considers theatre functions relating to movies and showings. Other functions such as employees, concessions, cleaning schedules, etc. have not been considered in this release.
- Halls cannot be inserted or updated from the front end in any way.

9.3 Future Enchanements

The future enhancements that may be brought about in the project may pertain to the limitations of the project. This includes:

- Factoring in additional functionalities relating to the theatre not covered under movies:
  - Concession Stands: Stalls selling food and drinks for movie-goers. This includes items sold, pre-orders etc.
  - Cleaning schedules: Halls need to be cleaned between movies, and theatre needs to be cleaned before opening and after closing.
  - Employees: Keeping track of employees hired, salaries, bonuses, duties, remarks etc.
- Account for expansion: Allow a high-ranking executive (with a new login ID) to add new halls to a location, or add new locations, with manager able to add shows only for their location, etc.

References

- Google (www.google.com)
- ChatGpt (https://chat.openai.com)
- YouTube (https://www.youtube.com)
- http://amsul.ca/pickadate.js/date/
- ttps://teamtreehouse.com/community/nested-loops-in-flask-how-to-iterate-and-make-nested-lists
- https://www.w3resource.com/mysql/date-and-time-functions/date-and-time-functions.php
- https://dev.mysql.com/doc/refman/8.0/en/date-and-time-functions.html
- https://dev.mysql.com/doc/connector-python/en/connector-python-example-connecting.html
- https://getbootstrap.com/docs/4.1/getting-started/introduction/
- https://w3resource.com/mysql/date-and-time-functions/mysql-dayname-function.php
- https://w3resource.com/mysql/date-and-time-functions/mysql-curdate-function.php
- https://w3resource.com/mysql/date-and-time-functions/mysql-datediff-function.php

About Project

Title of the project	MoiveTic
Semester	7
Members	5
Team Leader	Daksh Raj Singh
Describe role of every member in the project	Daksh Raj Singh Frontend and Graphics Dikshant Kumar Bhatiya Frontend and Backend Deepak Kumar Frontend ,Backend and Database Naina Soni Database Shreya Singh Frontend
What is the motivation for selecting this project?	Industry Demand
Project Type (Desktop Application, Web Application, Mobile App, Web)	Web – Application

Tools & Technologies:

Programming language used	HTML, CSS, JAVASCRIPT,PYTHON
Compiler used (with version)	VSCODE (1.73.1)
IDE used (with version)	VSCODE
Front End Technologies (with version, wherever Applicable)	HTML5, CSS3 , JAVASCRIPT ES6
Back End Technologies (with version, wherever applicable)	FLASK (2.2.2)
Database used (with version)	MYSQL (5.0.37)

Software Design Coding:

Is prototype of the software developed?	No
SDLC model followed (Waterfall, Agile, Spiral etc.)	Agile
Why above SDLC model is followed?	Everyone worked on different components
Justify that the SDLC model mentioned above is followed in the project.	Project was divided into different components and each component was made by different member.
Software Design approach followed (Functional or Object Oriented)	Object Oriented
In case Object Oriented approach is followed, which of the OOPS principles are covered in design?	Encapsulation, polymorphism
No. of Tiers (example 3-tier)	3-tier
Total no. of front end pages	0
Total no. of tables in database	6
Database is in which Normal Form?	First
Are the entries in database encrypted?	No
Front end validations applied (Yes / No)	Yes
Session management done (in case of web applications)	Yes
Is application browser compatible (in case of web applications)	Yes
Exception handling done (Yes / No)	Yes
Commenting done in code (Yes / No)	Yes
Naming convention followed (Yes / No)	Yes
What difficulties faced during deployment of project?	Difficulty in connecting database to backend, difficulty in applying exception handling
Total no. of Use-cases	1
Give titles of Use-cases	MovieTic Use Case Diagram

Project Requirements

MVC architecture followed (Yes / No)	Yes
If yes, write the name of MVC architecture followed (MVC-1,MVC-2)	MVC – 1
Design Pattern used (Yes / No)	Yes
If yes, write the name of Design Pattern used	MVC
Interface type (CLI / GUI)	GUI
No. of Actors	2
Name of Actors	Cashier,Manager
Total no. of Functional Requirements	8
List few important non Functional Requirements	Performance & Scalability,Portability & Compatibility,Reliability,Maintainability, Availability,Security,Usability

Testing

Which testing is performed ?(Manual or Automation)	Manual
Is Beta testing done for this project?	Yes

Write project narrative covering above mentioned points

We have developed a highly comprehensive and easy to use system for any small Theatre. It is easy to implement and requires no training to use. It provides options for managers and cashiers. It is error-proof and does large amount of work in the background. Thus, the system aids to simplify the processes used by cashier and manager as well as reduce operational costs, the primary concerns of all businesses.

Daksh Raj Singh	**0536CS201012**
Dikshant Kumar Bhatiya	**0536CS201014**
Deepak Kumar	**0187CS201048**
Naina Soni	**0187CS201103**
Shreya Singh	**0187CS201161**

Signature

Project guide

(Prof. Bhavna Soni)