**A**

**Project Report**

**on**

## Movie Information System

### BTech-IT, Sem VIII

**Prepared By:**

**Dedaniya Parth (IT023)**

**Guided By:**

**Prof. Harshad B. Prajapati**

**Dept. of Information Technology**

****

**Department of Information Technology Faculty of technology,**

**Dharmsinh Desai University**

**College road, Nadiad- 387001 April, 2023**

## CANDIDATE’S DECLARATION

I declare that 8th semester report entitled “**MovieBuzz**” is our own work conducted under the supervision of the guide **Prof. Harshad B. Prajapati.**

I further declare that to the best of our knowledge the report for B.Tech. VIII semester does not contain part of the work which has been submitted either in this or any other university without proper citation.

Candidate’s Signature

Candidate’s Name: Dedaniya Parthkumar A.

Student ID: 19ITUOS074

Submitted To:

Prof. Harshad B. Prajapati

Department of Information Technology,

Faculty of Technology,

Dharmsinh Desai University, Nadiad

Gujarat.

# DHARMSINH DESAI UNIVERSITY

**NADIAD-387001, GUJARAT**



# CERTIFICATE

**This is to certify that the project carried out in the subject of System Design Practice, entitled “MovieBuzz(Movie Information System)”and recorded in this report is a bonafide report of work of**

1. **Dedaniya Parth Roll No. IT023 ID No: 19ITUOS074**

**of Department of Information Technology, semester VIII. He was involved in Project work during academic year 2022 -2023.**

**Prof. Harshad B. Prajapati**

(Project Guide),

Department of Information Technology, Faculty of Technology,

Dharmsinh Desai University, Nadiad Date:

#### Prof. (Dr.) V. K. Dabhi

Head , Department of Information Technology, Faculty of Technology,

Dharmsinh Desai University, Nadiad Date:

### ACKNOWLEDGEMENT

It is indeed a great pleasure to express our thanks and gratitude to all those who helped us during this project. This project has given us a great opportunity to think, implement and interact with various aspects of the Software Development Life Cycle. We would like to acknowledge all the people who have helped us at one stage or another by providing the much-needed support, encouragement and groundwork to complete our project.

I express a deep sense of gratitude towards our project guide **Prof. Harshad B. Prajapati** towards his innovative ideas and earnest effort to make our project a success. It is his sincerity that prompted us throughout the project to do hard work using the industry adopted technologies. Our commitment to the application is the sole result of patience, hard work and dedication being inspired by him.

A blend of gratitude, pleasure and great satisfaction is what we feel to convey our indebtedness to all those who all have directly or indirectly contributed towards completion of the project.

with sincere regards,

**Dedaniya Parth**

## TABLE OF CONTENTS

[ACKNOWLEDGMENT](https://docs.google.com/document/d/1i6BY2TAP3SKSyLyGDMvS47WhnKS0SLvyrpe2QbJy_jQ/edit#heading%3Dh.44sinio) 4

ABSTRACT 7

LIST OF FIGURES 8

LIST OF TABLES 9

1. [Introduction 1](https://docs.google.com/document/d/1i6BY2TAP3SKSyLyGDMvS47WhnKS0SLvyrpe2QbJy_jQ/edit#heading%3Dh.2jxsxqh)0
   1. Purpose 10
   2. Document Conventions 10
   3. Intended audience and reading suggestions 10
   4. Product Scope 10
   5. [Technology and Literature Review](https://docs.google.com/document/d/1i6BY2TAP3SKSyLyGDMvS47WhnKS0SLvyrpe2QbJy_jQ/edit#heading%3Dh.4i7ojhp) 10
2. [Project Management](https://docs.google.com/document/d/1i6BY2TAP3SKSyLyGDMvS47WhnKS0SLvyrpe2QbJy_jQ/edit#heading%3Dh.2xcytpi) 11
   1. Feasibility Study 11
      1. [Technical feasibility](https://docs.google.com/document/d/1i6BY2TAP3SKSyLyGDMvS47WhnKS0SLvyrpe2QbJy_jQ/edit#heading%3Dh.1ci93xb) 11
      2. [Time Schedule feasibility](https://docs.google.com/document/d/1i6BY2TAP3SKSyLyGDMvS47WhnKS0SLvyrpe2QbJy_jQ/edit#heading%3Dh.3whwml4) 11
      3. [Operational feasibility](https://docs.google.com/document/d/1i6BY2TAP3SKSyLyGDMvS47WhnKS0SLvyrpe2QbJy_jQ/edit#heading%3Dh.2bn6wsx) 11
      4. [Implementation feasibility](https://docs.google.com/document/d/1i6BY2TAP3SKSyLyGDMvS47WhnKS0SLvyrpe2QbJy_jQ/edit#heading%3Dh.qsh70q) 12
   2. [Project Planning](https://docs.google.com/document/d/1i6BY2TAP3SKSyLyGDMvS47WhnKS0SLvyrpe2QbJy_jQ/edit#heading%3Dh.3as4poj) 12
      1. [Project Development Approach and Justification](https://docs.google.com/document/d/1i6BY2TAP3SKSyLyGDMvS47WhnKS0SLvyrpe2QbJy_jQ/edit#heading%3Dh.1pxezwc) 12
      2. [Project Plan](https://docs.google.com/document/d/1i6BY2TAP3SKSyLyGDMvS47WhnKS0SLvyrpe2QbJy_jQ/edit#heading%3Dh.49x2ik5) 14
      3. [Roles and Responsibilities](https://docs.google.com/document/d/1i6BY2TAP3SKSyLyGDMvS47WhnKS0SLvyrpe2QbJy_jQ/edit#heading%3Dh.2p2csry) 14
3. System Requirements Study 15
   1. Problems and Weaknesses of Current System 15
   2. User Characteristics 15
   3. Hardware and Software Requirements 15
   4. [Constraints](https://docs.google.com/document/d/1i6BY2TAP3SKSyLyGDMvS47WhnKS0SLvyrpe2QbJy_jQ/edit#heading%3Dh.147n2zr) 16
      1. Hardware Limitations 16
      2. Reliability Requirements 16
   5. Assumptions and Dependencies 16
4. [System Analysis](https://docs.google.com/document/d/1i6BY2TAP3SKSyLyGDMvS47WhnKS0SLvyrpe2QbJy_jQ/edit#heading%3Dh.3o7alnk) 17
   1. [Requirements of New System](https://docs.google.com/document/d/1i6BY2TAP3SKSyLyGDMvS47WhnKS0SLvyrpe2QbJy_jQ/edit#heading%3Dh.23ckvvd) 17
      1. User Requirements 17
      2. System Requirements 18
5. [System Design](https://docs.google.com/document/d/1i6BY2TAP3SKSyLyGDMvS47WhnKS0SLvyrpe2QbJy_jQ/edit#heading%3Dh.32hioqz) 22
   1. [Use Case Diagram](https://docs.google.com/document/d/1i6BY2TAP3SKSyLyGDMvS47WhnKS0SLvyrpe2QbJy_jQ/edit#heading%3Dh.1hmsyys) 22
   2. [Class Diagram](https://docs.google.com/document/d/1i6BY2TAP3SKSyLyGDMvS47WhnKS0SLvyrpe2QbJy_jQ/edit#heading%3Dh.41mghml) 23
   3. [Sequence Diagram](https://docs.google.com/document/d/1i6BY2TAP3SKSyLyGDMvS47WhnKS0SLvyrpe2QbJy_jQ/edit#heading%3Dh.2grqrue) 24
   4. Activity Diagram 26
6. [Implementation Planning](https://docs.google.com/document/d/1i6BY2TAP3SKSyLyGDMvS47WhnKS0SLvyrpe2QbJy_jQ/edit#heading%3Dh.1v1yuxt) 27
   1. [Implementation Environment](https://docs.google.com/document/d/1i6BY2TAP3SKSyLyGDMvS47WhnKS0SLvyrpe2QbJy_jQ/edit#heading%3Dh.4f1mdlm) 27
   2. Program/Modules Specification 27
   3. [Coding Standards](https://docs.google.com/document/d/1i6BY2TAP3SKSyLyGDMvS47WhnKS0SLvyrpe2QbJy_jQ/edit#heading%3Dh.2u6wntf) 27
7. Testing 28
   1. [Testing Plan](https://docs.google.com/document/d/1i6BY2TAP3SKSyLyGDMvS47WhnKS0SLvyrpe2QbJy_jQ/edit#heading%3Dh.19c6y18) 28
   2. [Testing Strategy](https://docs.google.com/document/d/1i6BY2TAP3SKSyLyGDMvS47WhnKS0SLvyrpe2QbJy_jQ/edit#heading%3Dh.3tbugp1) 28
   3. [Testing Methods](https://docs.google.com/document/d/1i6BY2TAP3SKSyLyGDMvS47WhnKS0SLvyrpe2QbJy_jQ/edit#heading%3Dh.28h4qwu) 28
   4. [Test Cases](https://docs.google.com/document/d/1i6BY2TAP3SKSyLyGDMvS47WhnKS0SLvyrpe2QbJy_jQ/edit#heading%3Dh.28h4qwu) 29
8. User Manual 32
9. Limitation and Future Enhancement 46
10. [Conclusion and Discussion](https://docs.google.com/document/d/1i6BY2TAP3SKSyLyGDMvS47WhnKS0SLvyrpe2QbJy_jQ/edit#heading%3Dh.nmf14n) 47
    1. Conclusions 47
    2. [Discussion](https://docs.google.com/document/d/1i6BY2TAP3SKSyLyGDMvS47WhnKS0SLvyrpe2QbJy_jQ/edit#heading%3Dh.37m2jsg) 47
       1. [Self Analysis of Project Viabilities](https://docs.google.com/document/d/1i6BY2TAP3SKSyLyGDMvS47WhnKS0SLvyrpe2QbJy_jQ/edit#heading%3Dh.1ksv4uv) 47

10.2.3 Problem Encountered and Possible Solutions 47

10.3.3 Summary of Project work 47

1. [References](https://docs.google.com/document/d/1i6BY2TAP3SKSyLyGDMvS47WhnKS0SLvyrpe2QbJy_jQ/edit#heading%3Dh.1mrcu09) 48

## ABSTRACT

The “MovieBuzz(movie information system)” project aims to provide a comprehensive and user-friendly platform for movie enthusiasts to explore, discover and keep track of information related to their favorite movies. The system will utilize a vast database of movie information, including details such as movie titles, cast and crew information, ratings, reviews, and trailers. Users will be able to search for movies based on various criteria such as popularity, top rated, upcoming and genre. Also they will have the ability to create personalized movie watchlists and get personalized movie recommendations based on their viewing history. The project will utilize modern web development technologies and will prioritize user experience, security, and scalability. Ultimately, this system aims to be the go-to destination for movie lovers around the world.

## LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
|  | **Name** | **Page No.** |
| Fig 2.2.1 | Agile Model | 13 |
| Fig 5.1 | Use Case Diagram | 22 |
| Fig 5.2 | Class Diagram | 23 |
| Fig 5.3(a) | Seq. Diagram User (Login Register) | 24 |
| Fig 5.3(b) | Seq. Diagram Add Vehicle | 25 |
| Fig 5.3(c) | Seq. Diagram Book Vehicle | 25 |
| Fig 5.4 | Activity Diagram | 26 |

## LIST OF TABLES

|  |  |  |
| --- | --- | --- |
|  | **Name** | **Page No.** |
| Table 2.2.2 | Project Plan | 14 |
| Table 2.2.3 | Roles and Responsibilities | 14 |
| Table 7.4.1 | User Test Cases | 29 |
| Table 7.4.2 | Admin Test Cases | 30 |

1. **INTRODUCTION**
   1. **PROJECT DETAILS**

This Web Application will provide information of Movies, Web Show which are released or going to release. Users have to Login/Signup to make watchlist but they can find all information regarding movie without login.

Web application provides details about movie release date/year, genre, length, rating, review, pictures, production companies, cast and crew, recommended movies, etc.

* 1. **PURPOSE**

To make platform where people can find any information about any Movies and TV show so that people don’t have to go to many websites for finding different information about movie or show.

* 1. **SCOPE**
* People can find the best movie or show of their taste,
* People can suggest other Movie/TV Show related to specific one
  1. **OBJECTIVE**

The main objective of this project is to provide all information of the movie including synopsis, actors, production companies and recommended movies.

### Technology and Literature Review

Following technologies will be used for development/management/tracking activities….

1. HTML, CSS, React for Frontend
2. TMDB API used to fetch all movie related details

## PROJECT MANAGEMENT

### Feasibility Study

#### Technical feasibility

Viewing our project from technical point of view (thinking about various tools and technologies being used in developing the system). I have decided that following technologies will be more than enough to develop complete working system (including tech. & tools used for project tracking, monitoring etc. along with development).

For web app development: VS Code

Database: Use TMDB API

I am equipped with basic workflows of each tool and tech. and capable to explore further if required. Each of the above technologies is freely available and some of the skills yet to be learnt but it is manageable. From this, it is clear that my project is technically feasible.

#### Time schedule feasibility

I have planned the steps for completion of project in given duration. Firstly, I will perform requirement gathering & analysis in the month of December 2022. I will prepare SRS document and the GUI design tentatively by February 2023 ending. The diagrams required for the design will be tentatively completed by February 2023. For coding and unit testing 4 weeks and for system and integration testing another 2 weeks will be required. Hence tentatively by the 25th March 2023, I will be able to complete the project and ready for its demonstration at starting of April 2021. Being a 1-member team, I have to be very precise to complete my project in the estimated time.

#### Operational feasibility

I have kept the web application pretty simple and user-friendly so everyone can use it without any difficulty.

2.1.4 Implementation feasibility

The Functional requirements of project will be easily fulfilled by the technologies I have mentioned. All the technologies I am using are easily supported by all the Web Browsers. Hence, the implementation is feasible.

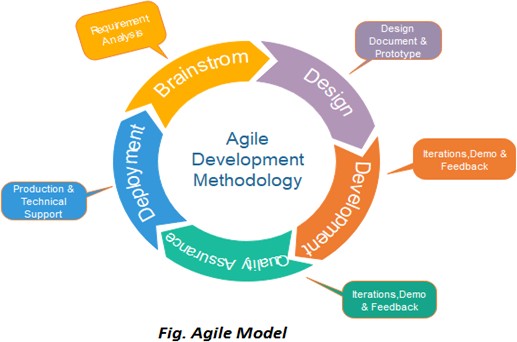
### Project Planning

#### Project Development Approach and Justification

I would be using the Agile model for project development. Agile methods break tasks into smaller iterations, or parts do not directly involve long term planning. The project scope and requirements are laid down at the beginning of the development process. Plans regarding the number of iterations, the duration and the scope of each iteration are clearly defined in advance.

Following are the phases in the Agile model are as follows:

1. Requirements gathering
2. Design the requirements
3. Construction/ iteration
4. Testing/ Quality assurance
5. Deployment
6. Feedback



*Fig 2.2.1 Agile Model*

Advantages of Agile model:

* Customer satisfaction by rapid, continuous delivery of useful software.
* People and interactions are emphasized rather than process and tools. Customers, developers and testers constantly interact with each other.
* Working software is delivered frequently (weeks rather than months).
* Face-to-face conversation is the best form of communication.
* Close, daily cooperation between business people and developers.
* Continuous attention to technical excellence and good design.
* Regular adaptation to changing circumstances.
* Even late changes in requirements are welcomed.

Disadvantages of Agile model:

* In case of some software deliverables, especially the large ones, it is difficult to assess the effort required at the beginning of the software development life cycle.
* There is a lack of emphasis on necessary designing and documentation.
* The project can easily get taken off track if the customer representative is not clear what final outcome that they want.
* Only senior programmers are capable of taking the kind of decisions required during the development process. Hence it has no place for newbie programmers, unless combined with experienced resources

#### Project Plan

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Task Name** | **Start** | **Finish** | **December** | | | | **January** | | | | **February** | | | | **March** | | | |
|  |  |  | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** |
| **Plan and** | 10/12/2022 | 09/01/2023 | | | | | | | | | | | | |  | | | |  | | | |
| **Feasibility** |  |  |  | | | |  | | | |
| **Study** |  |  |  | | | |  | | | |
| **Requirement** | 10/01/2023 | 31/01/2023 |  | | | | | | | | | | |  | | | |  | | | |
| **gathering** |  |  |  | | | |
| **Analysis** | 01/02/2023 | 07/02/2023 |  | | | |  | | | | | | | | |  | | | |
|  |  |  |  | | | |
| **Design** | 07/02/2023 | 28/02/2023 |  | | | |  | | | | | | | | | | |  | | | |
|  |  |  |  | | | |
| **Coding** | 01/03/2023 | 31/03/2023 |  | | | |  | | | | | | | | | | | | | | | |
|  |  |  |  | | | |
| **Testing** | 01/04/2023 | 14/04/2023 |  | | | |  | | | |  | | | |  | | | |
|  |  |  |

*Table 2.2.2*

#### Roles and Responsibilities

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Name | Analysis | Design | Coding | Testing | Documentation | Maintenance |
| Dedaniya Parth | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ |

## SYSTEM REQUIREMENTS STUDY

### Problems and Weaknesses of Current System

On the basis of current scenario people searching for good movie and tv shows and it will get time consuming as well as they have one fear that may be this will not up to user expectation because of lack of information and ratings

Some known problems and weaknesses of the current system include the following: -

* When people are searching of finding any movies, they got much time consume due to not getting desire recommendation.
* Sometime people get movie but due lack of any information about casts, it may not good experience.

### User Characteristics (Type of users who is dealing with the system)

There are 2 types of user in system require:

1. User
2. Admin
3. User
   * + User needs to register on our website for accessing watchlist functionality.
     + Users can use all other functionalities without login.
4. Admin
   * Admin can see all users available in system and can remove too.

### Hardware and Software Requirements (minimum requirements to run your system)

There are no such specific hardware requirements other than basic requirements such as a computer with good internet connectivity and a decent browser.

Software: -

For all the actors the basic software requirements are as follows

- Good internet connection

Hardware: -

For all the actors the basic hardware requirements are as follows

- Android Device, Desktop

### Constraints

* + 1. Hardware Limitations
* There is only one limitation of this web app, that the device must have the browser.
  + 1. Reliability Requirements
* The web app does demand much reliability and it is fully assured that the particular information about the users should be secured and flow is maintained and accessed according to the rights.

### Assumptions and Dependencies

1. Users have sufficient privileges to access the internet.
2. Browser on Device is running smoothly.
3. Database updates are giving expected and accurate results.

## SYSTEM ANALYSIS

### Requirements of New System (SRS)

#### User Requirement

Since, this web application for user interaction through movies reviews and ratings so no their requirements for user.

* + 1. **System Requirements**

### 1 Functional requirements

***R1: Register and Log in***

*R1.1 Description and Priority*

This features creates an account for a new user in the system. For a user to be able to make a watchlist he/she must have an account and must be logged into it. This feature enlists the user details in the database. This is important feature so it has high priority.

*R1.2 Stimulus/Response Sequences*

User first clicks on the button to initiate registration process. The system then prompts the user to fill out his/her first name, last name, age, email address, and their passwords. User enters the fields. System validates the user’s information and creates a new account for the user.

Once a user register, he/she clicks on the log in button to initiate the log in process. The system prompts the user for his/her email and password and verifies the information. After verification the system displays account home page to the user.

*R1.3 Functional Requirements*

REQ-1: A button for register and log in must be available.

REQ-2: A table in database must be created for all users.

***R2: View Movies***

*R2.1 Description and Priority*

With the help of this feature user can see all available movies details.

*R2.2 Stimulus/Response Sequences*

If no movie available at that time then message informing same must be displayed else user can move forward to another movie.

*R2.3 Functional Requirements*

REQ-1: A button for view movie by categories.

REQ-2: A card for each movie, by clicking this card user can see all details about movie.

***R3: Add to Watchlist***

*R3.1 Description and Priority*

This feature will add the movie to the watchlist of the user.

*R3.2 Stimulus/Response Sequences*

User clicks the button to add the movie to the waitlist, if user is logged in then movie will be added to watchlist else he/she prompts to login.

*R3.3 Functional Requirements*

REQ-1: A page dealing with the movie details must be created.

REQ-2: The table dealing with watchlist movie must be updated.

REQ-3: A table dealing with all watchlists by different users must be maintained.

***R4: Search movie by name***

*R4.1 Description and Priority*

With the help of this feature users can find movie by name.

*R4.2 Stimulus/Response Sequences*

User clicks the search-bar button to initiate find movie process and user can see movie name by writing name..

*R4.3 Functional Requirements*

REQ-1: A search bar need to be created in navbar.

REQ-2: When user enters, then movie names should be appear.

***R5: View movies by popularity, top rated and upcoming***

*R5.1 Description and Priority*

By using this feature user can see movies which are currently more popular, which are top rated and upcoming movies.

*R5.2 Stimulus/Response Sequences*

When user clicks on the popular button, then an API call to be made to TMDB and popular movie will be shown. Same for top rated button and upcoming button will be done.

*R5.3 Functional Requirements*

REQ-1: Three tab for the same to be created.

REQ-2: When user clicks on the tab, same category movie need to be shown.

***R6: Find movies by genre***

*R6.1 Description and Priority*

By using this feature user can find movies by specific genre which he/she wants to find.

*R6.2 Stimulus/Response Sequences*

When user clicks on the any genre, then that genre movies should be shown.

*R6.3 Functional Requirements*

REQ-1: Movie genre tabs to be created under the movie name in movie details page.

REQ-2: When user clicks on the genre button, same genre movies should be shown.

***R7: View recommendations***

*R7.1 Description and Priority*

By using this feature user can see recommended movies based on the current movie.

*R7.2 Stimulus/Response Sequences*

When user clicks on the movie details, there should be section of recommended movies should be shown.

*R7.3 Functional Requirements*

REQ-1: Recommended movie section to be created.

1. **Non-Functional Requirements**

***N1: Performance Requirements***

Maximum possible quick response to user's action. The changes if any made should be reflected automatically in the next screens. Also, the connections to the servers will be based on the attributes of the user like his location and server will be working 24X7 times.

***N2: Safety Requirements***

The application is password protected and also Login is done by only authorized Email.

***N3: Software Quality Attributes***

The necessary qualities of software products are

*N3.1 Security*

The application is password protected and also Login is done by only authorized Email.

*N3.2 Maintainability*

The system is to be designed so that it is easily maintained. Also it should allow incorporating new requirements in any module of system.

*N3.3 Reliability*

The web application will be able to handle multiple Signup.

*N3.4 Portability*

The application will be easily portable on any system like Windows, macOS and Linux.

***N4: Business Rules***

Only the admin can do changes in the database. The user can only search, Login and find recommendations on website.

# System Design

### Use Case Diagram

### 

Fig 5.1

### Class Diagram

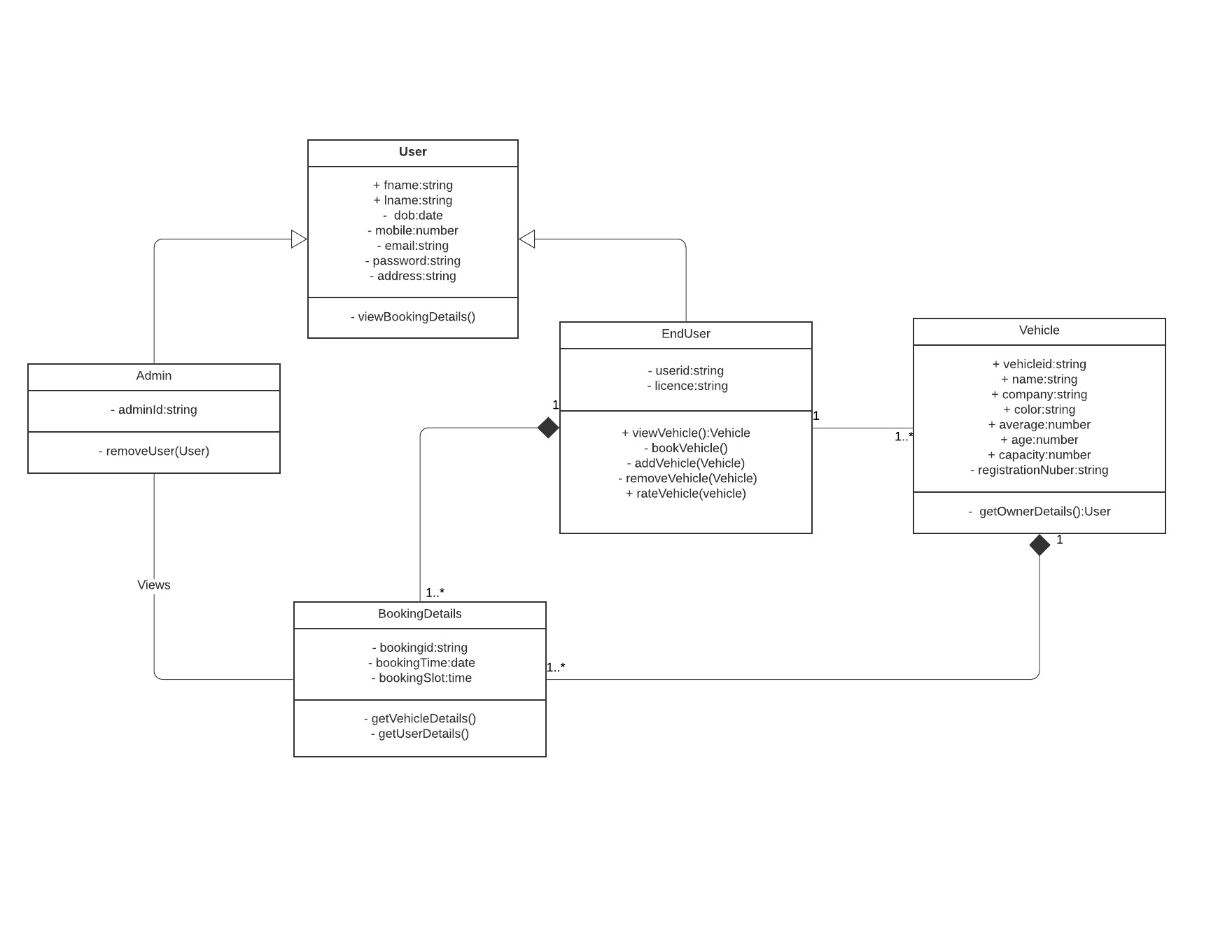
****

Fig 5.2

### Sequence Diagram

#### User (Login Register)

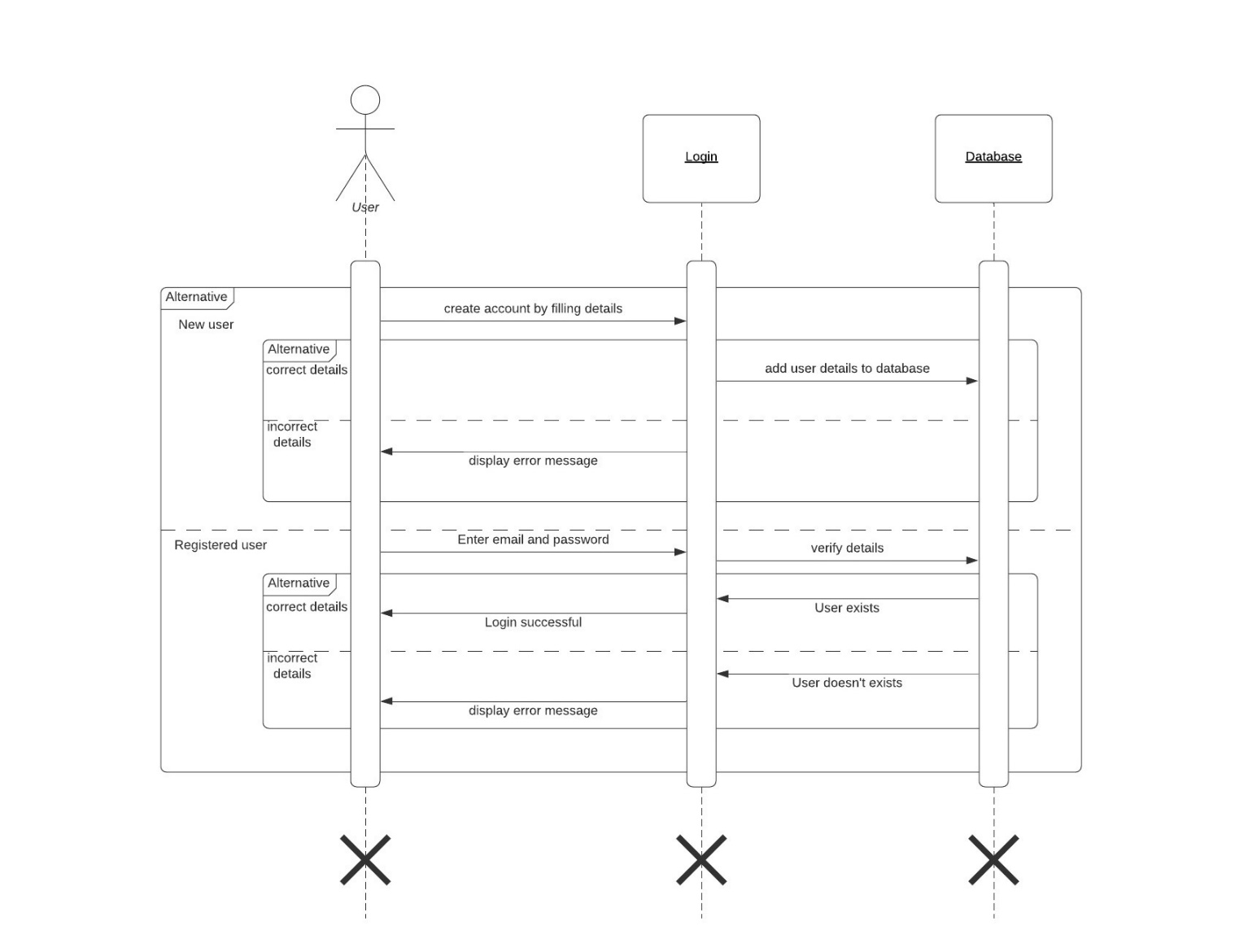
****

Fig 5.3(a)

#### Now Below are sequence diagrams for :

* + 1. **Add Vehicle** Fig 5.3(b)
    2. **Book Vehicle** Fig 5.3(c)

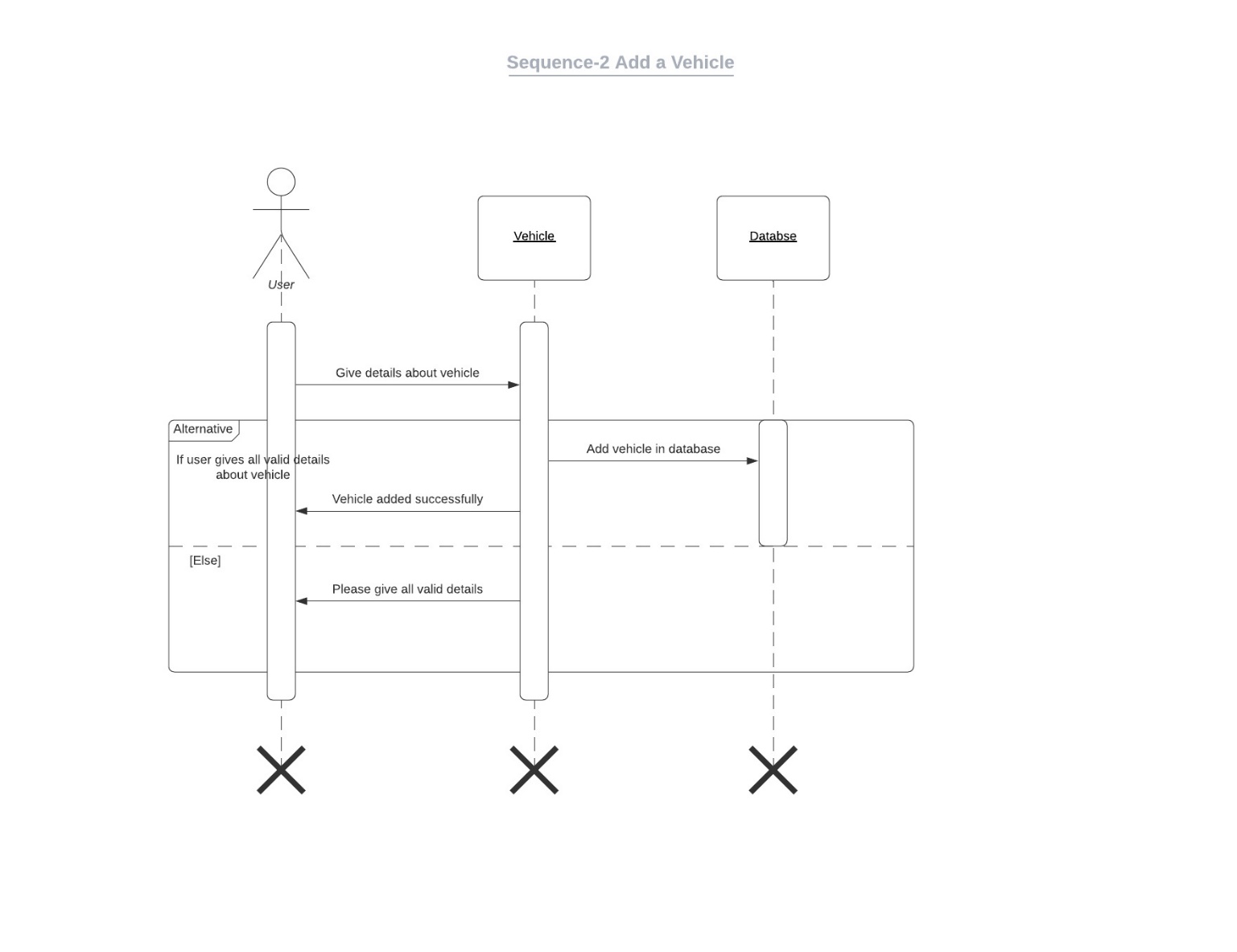


Fig 5.3(b)

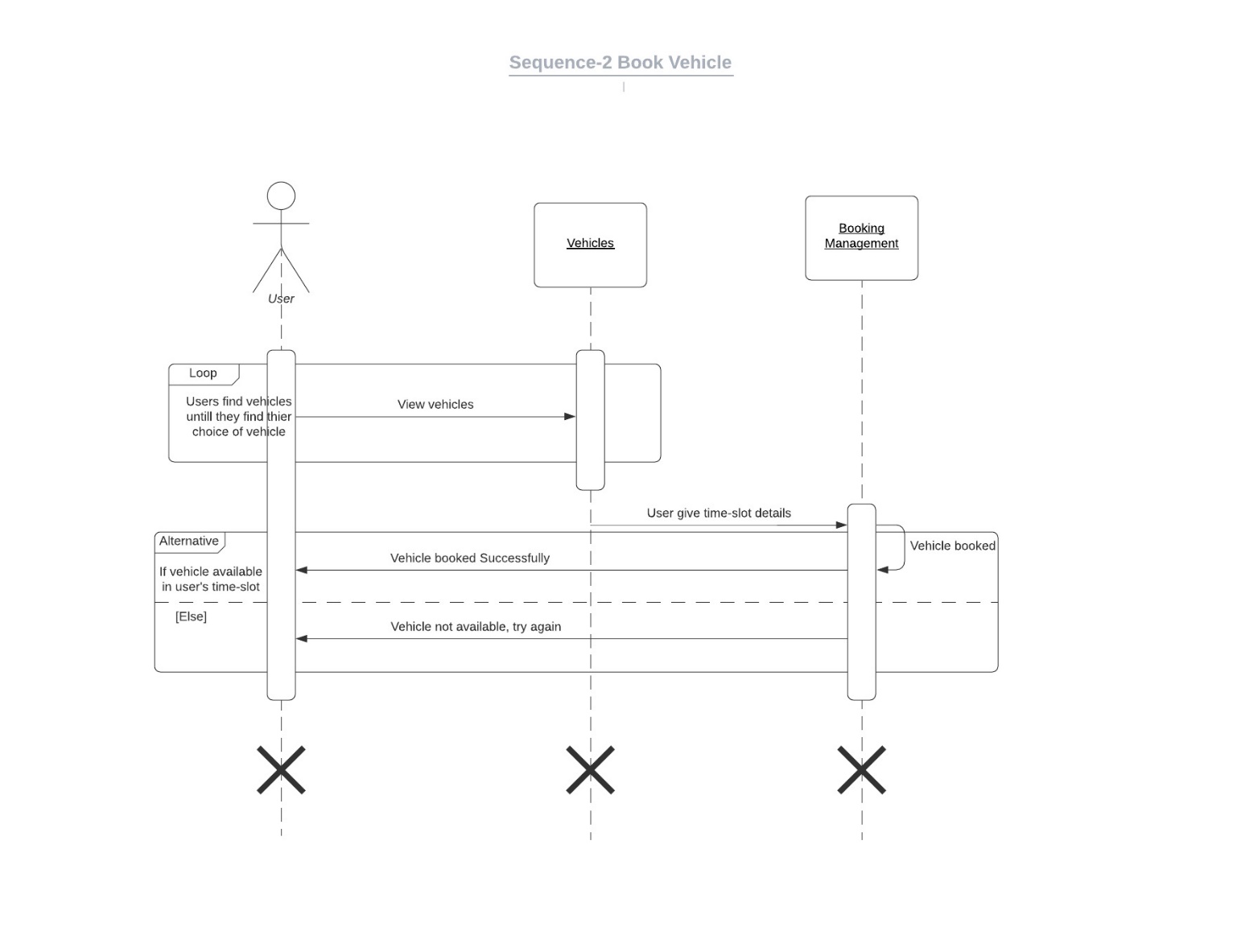


Fig 5.3(c)

### Activity Diagram

### 

Fig 5.4

## IMPLEMENTATION PLANNING

### Implementation Environment (Single vs Multi User, GUI vs Non GUI)

For implementation we have used:

Visual Studio Code

Our project is built using Visual Studio Code seeing that it is an web

application we saw fit that Visual studio code provided us with all the required basis for successful implementation of our web app. Also to for storing our data we have used Mongo database which enables our web app to run at all times.

### Program/Modules Specification

The following Modules are implemented:

User Module

There are two types of Users: Admin, User.

This module is the base for authentication and authorization to ensure the security aspect of the user

Search Module

This module is the very used module as user can find a movie by it’s name and also by categories like popular, top rated, upcoming movies and also find by genres.

Recommendation Module

This module is the base for general field where user can see details about movies and tv shows. They will get recommendation on basis of what kind of movie or tv shows they are searching for.

### Coding Standards

To make the system coding easy, easy to remember and reduce the chances of errors, some techniques are used at the time of coding of the application which is called coding standard. The coding standard which we adopted during the coding is explained as follows:

* + - Each nested block should be properly indented and spaced.
    - The code should be properly commented for understanding easily.

Comments regarding the statements increase the understandability of the code.

* + - Better to avoid use of digits in variable names.
    - The names of the function should be written in camel case starting with small letters.
    - The name of the function must describe the reason for using the function clearly and briefly.

## TESTING

### Testing Plan

The testing technique that is going to be used in the project is White box testing. In White box testing the Tester has knowledge

about the internal structure of the code or the program of the software.

### Testing Strategy

The development process repeats this testing subprocess a number of times for the following phases.

* + 1. Unit Testing.
    2. Integration Testing

Unit Testing tests a unit of code (module or program) after coding of that unit is completed.

Integration Testing tests whether the various programs that make up a system, interface with each other as desired, fit together and whether the interfaces between the programs are correct.

Testing is carried out in such a hierarchical manner to ensure that each component is correct and the assembly/combination of components is correct. Merely testing a whole system at the end would most likely throw up errors in components that would be very costly to trace and fix.

### Testing Methods

Black Box and White Box Testing*:*

In black-box testing a software item is viewed as a black box, without knowledge of its internal structure or behavior. Possible input conditions, based on the specifications (and possible sequences of input conditions), are presented as test cases.

In white-box testing knowledge of internal structure and logic is exploited. Test cases are presented such that possible paths of control flow through the software item are traced. Hence more defects than black-box testing are likely to be found.

Out of the 2 methods for testing, black box testing and white box testing, we would be using the white box testing as we are well aware of the internal functionalities of our application unlike in the black box testing, where we require a 3rd party to test our cases and the internal details are hidden from him.

### Test Cases

### 1 Users

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity** | **Test Case** | **Expected Output** | **Actual Output** | **Result** |
| Registration of Users | Validation | Successfully Registered | Successfully Registered | Pass |
| Login of Users | Validation | Login Successful | Login Successful | Pass |
| Profile | View Profile | View Profile Details Successful | View Profile Details Successful | Pass |
| View movies | View movies | Available movies shown | Available movies shown successfully | Pass |
| Add to watchlist | Add movie to watchlist | Add movie to the watchlist | Add movie to the watchlist successfully | Pass |
| Category Filter | See movie by category | See movie by category like popular, top rated, upcoming | See movie by category like popular, top rated, upcoming  successfully | Pass |
| Search movie | Search movie by name | Write movie name and find similar movies | Movie shown based on search successfully | Pass |
| Recommendation | Find recommend movies | See section of recommended movie section | See section of recommended movie section successfully | Pass |
| Movie by genre | View movie by genre | Movies of specific genre shown | Movies of specific genre shown successfully | Pass |

*Table 7.4.1 User Test Cases*

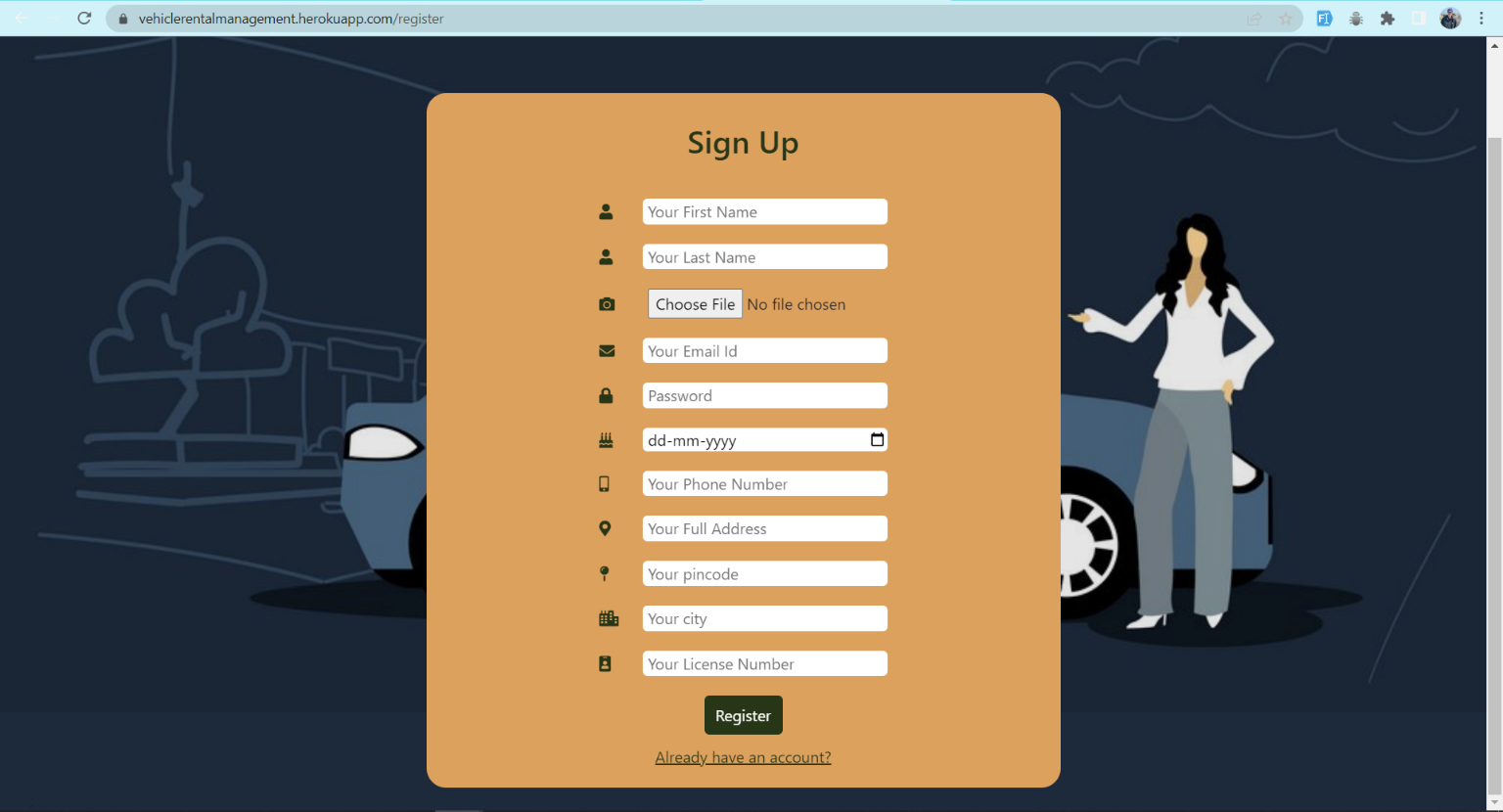
## USER MANUAL

User Manuals are manuals that enable the user of a system or application to understand the working of the system and help them to use them efficiently. It is usually written by a technical writer, although user guides are written by programmers, productor project managers, or other technical staff, particularly in smaller companies.

Follow below mentioned steps in order to work with the app:

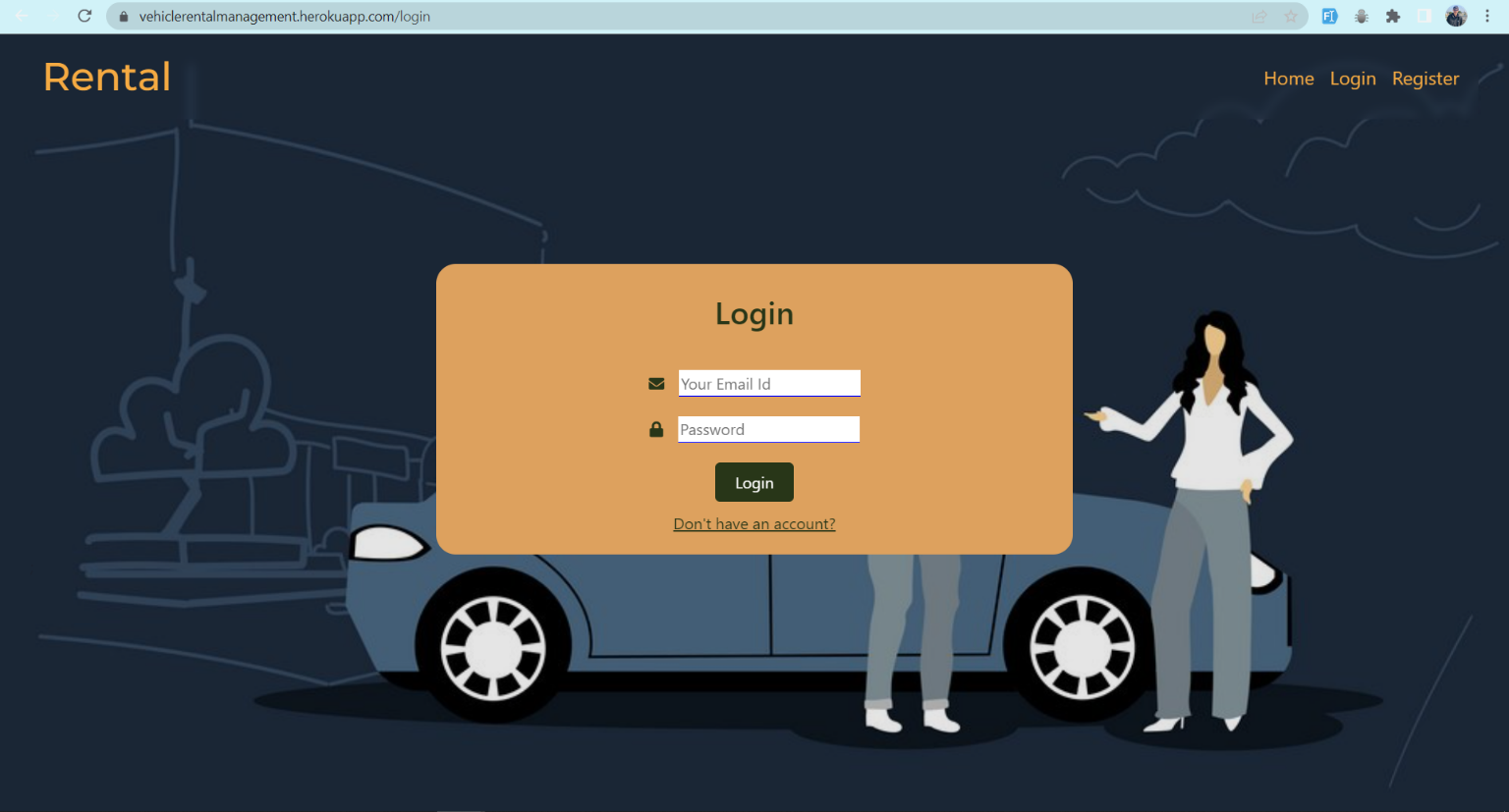
### Registration:

* Users can register themselves by clicking on register button.
* User has to fill up the form and click the register button.
* After that user is registered to the system and profile is created.



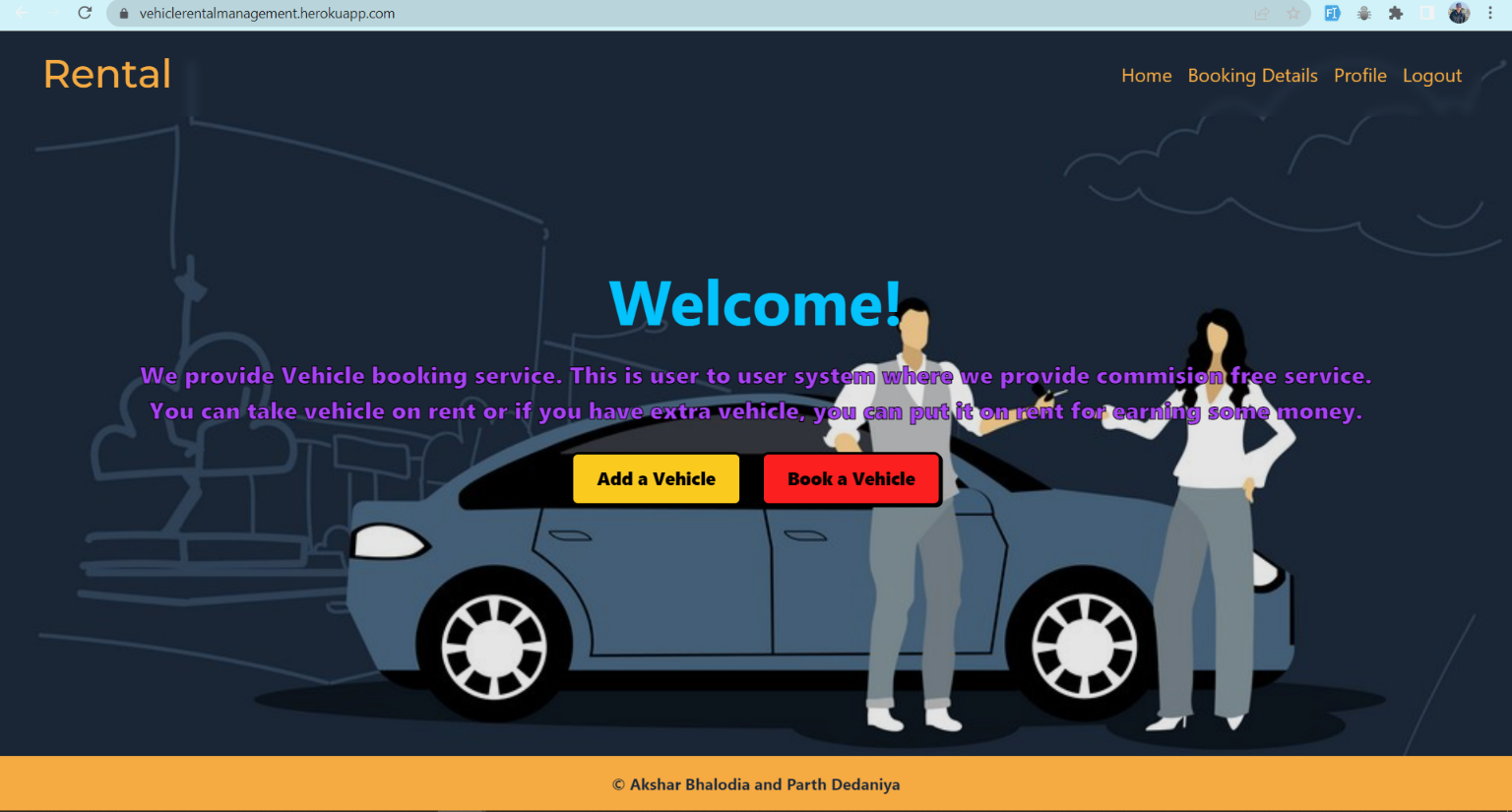
### Login:

* you would just have to be registered once else you will be simply logged into the app
* Once you fill the required credentials you will be redirected to your Home page.



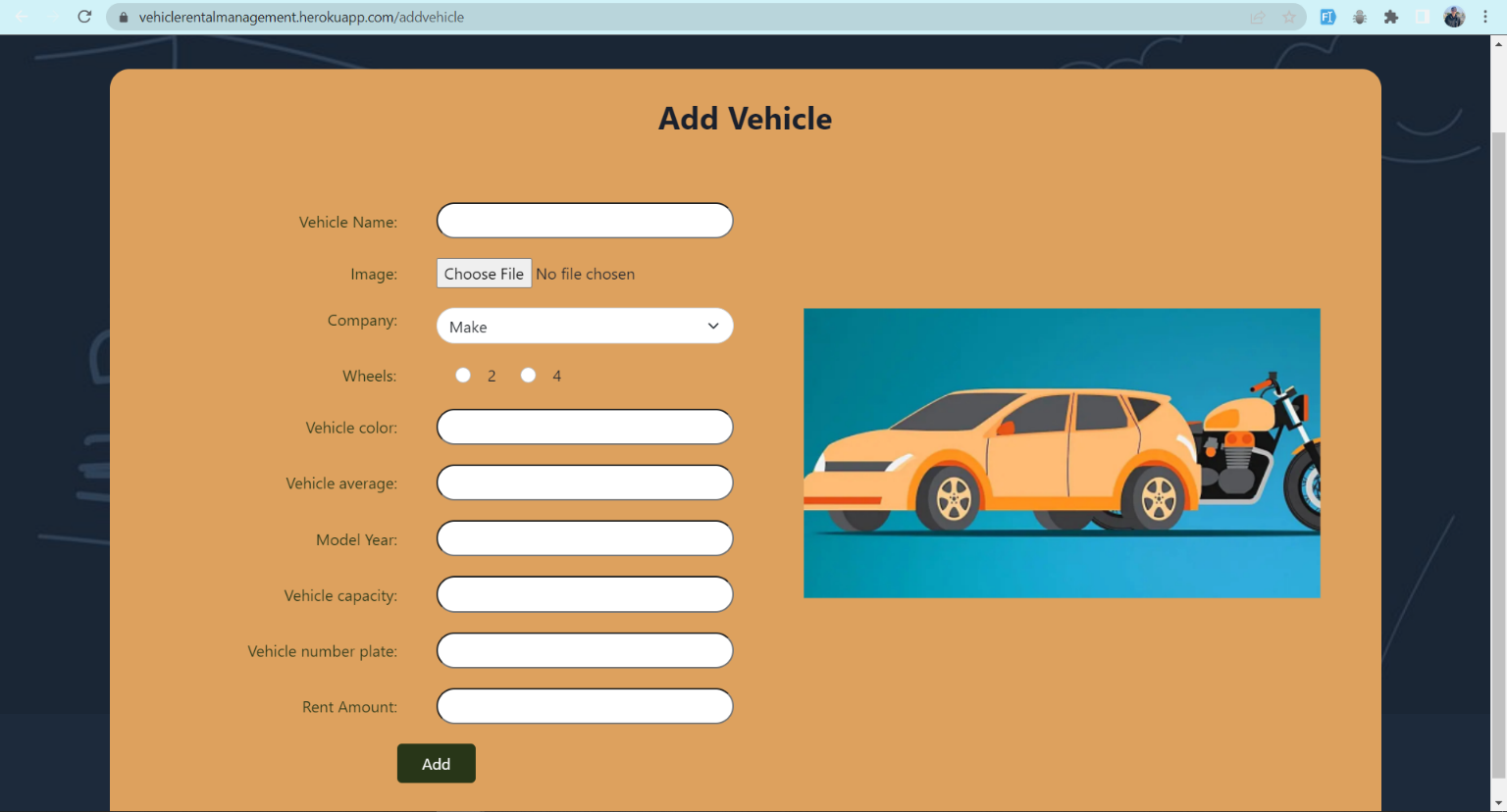
### Home Page:

* From here user can see movies on the platform.



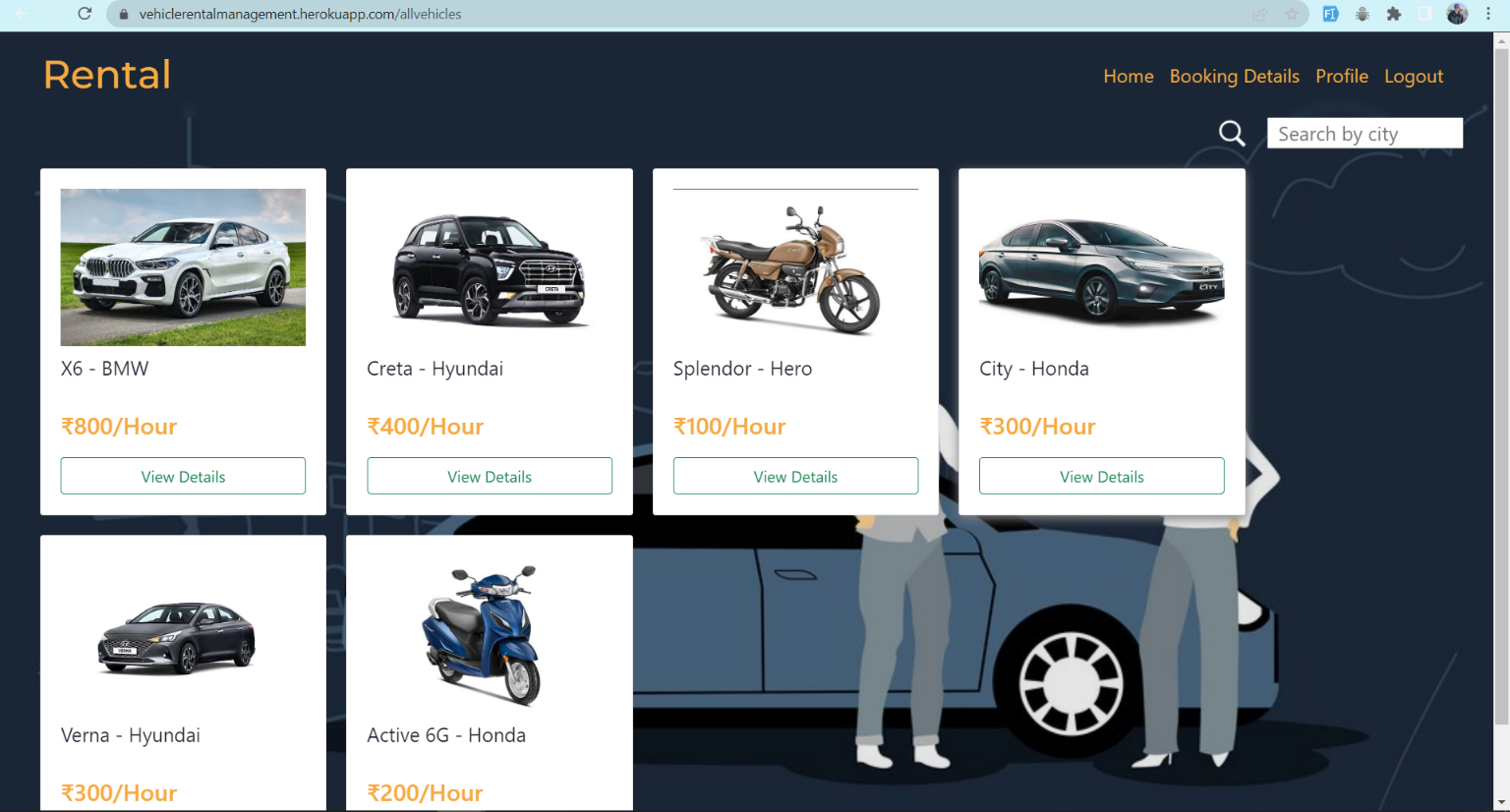
### Search movie:

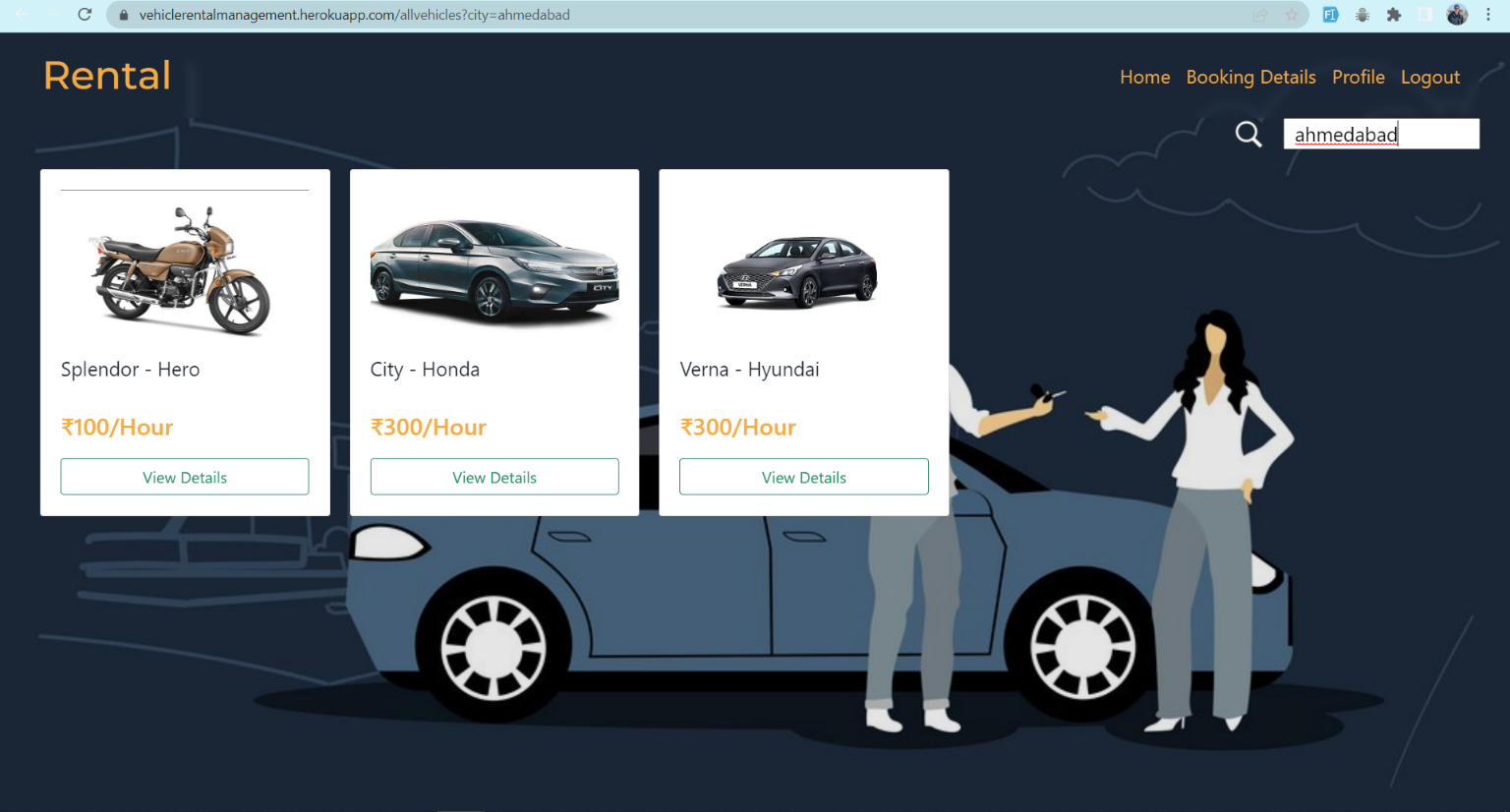
* On clicking search bar use can search movie by name
* There will be list of movie will be shown base on searched movie.



### View movies by categories:

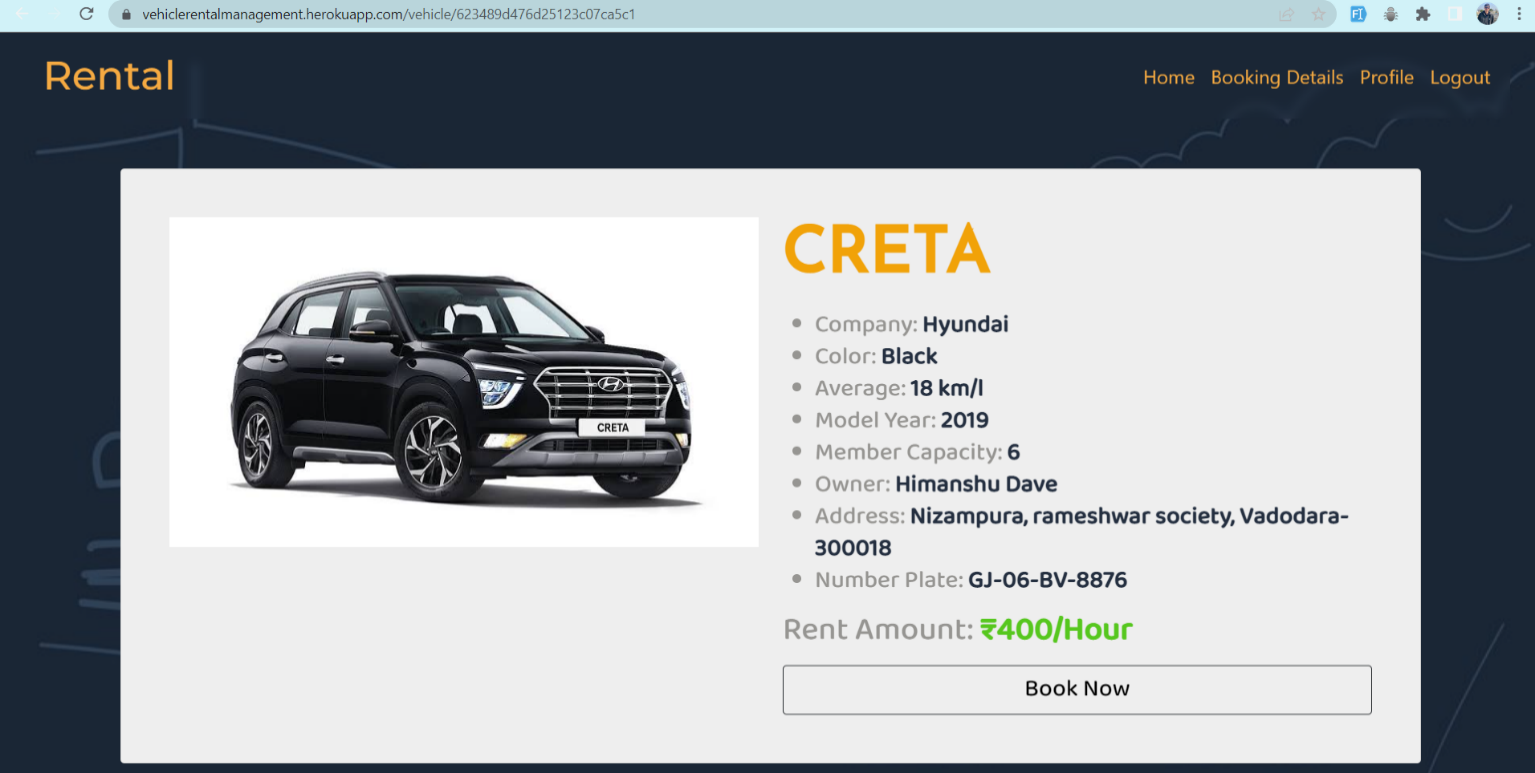
* on clicking popular users can see popular movies
* on clicking top rated users can see top rated movies
* on clicking upcoming users can see upcoming movies





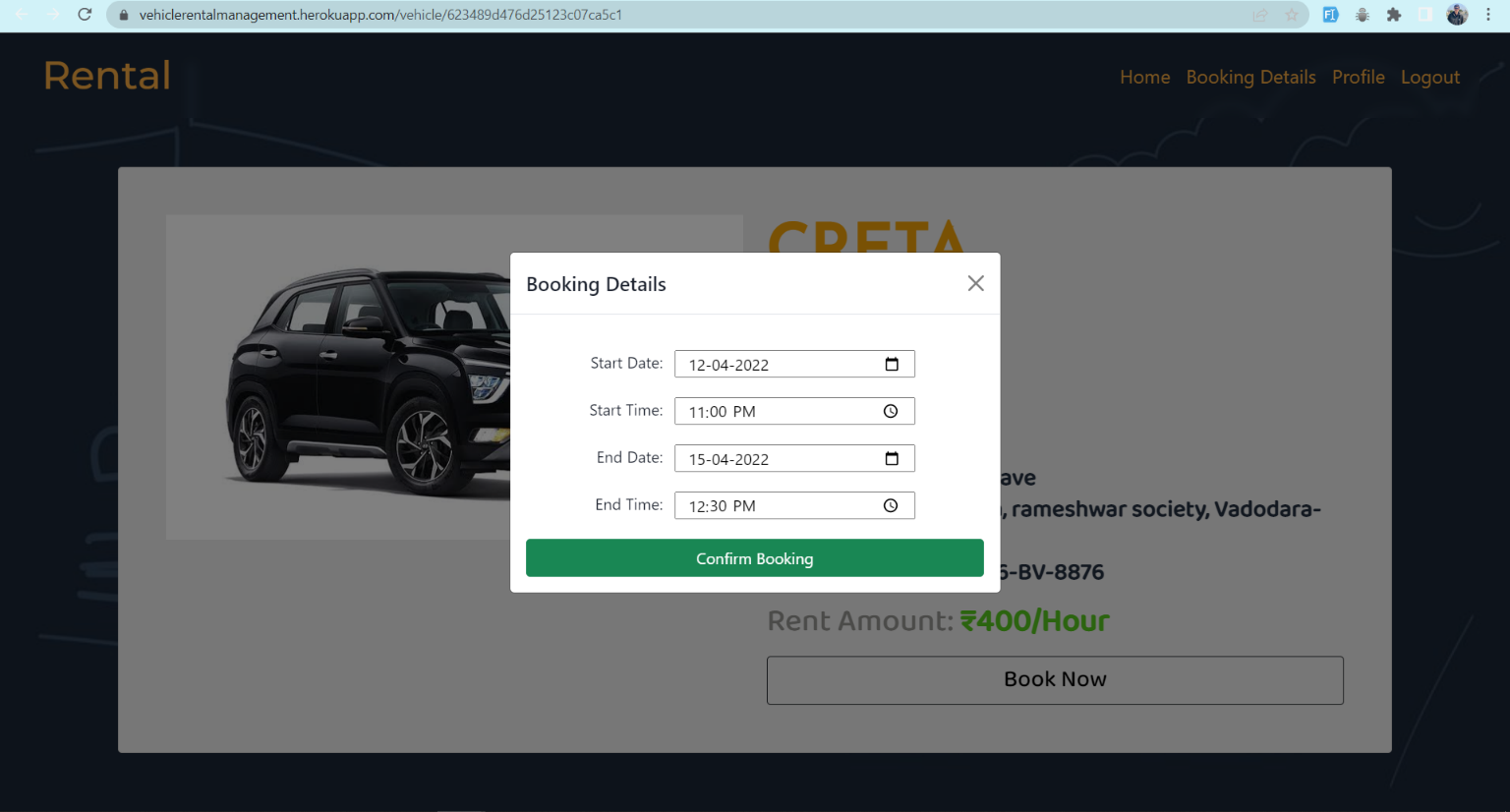
### View Details Page:

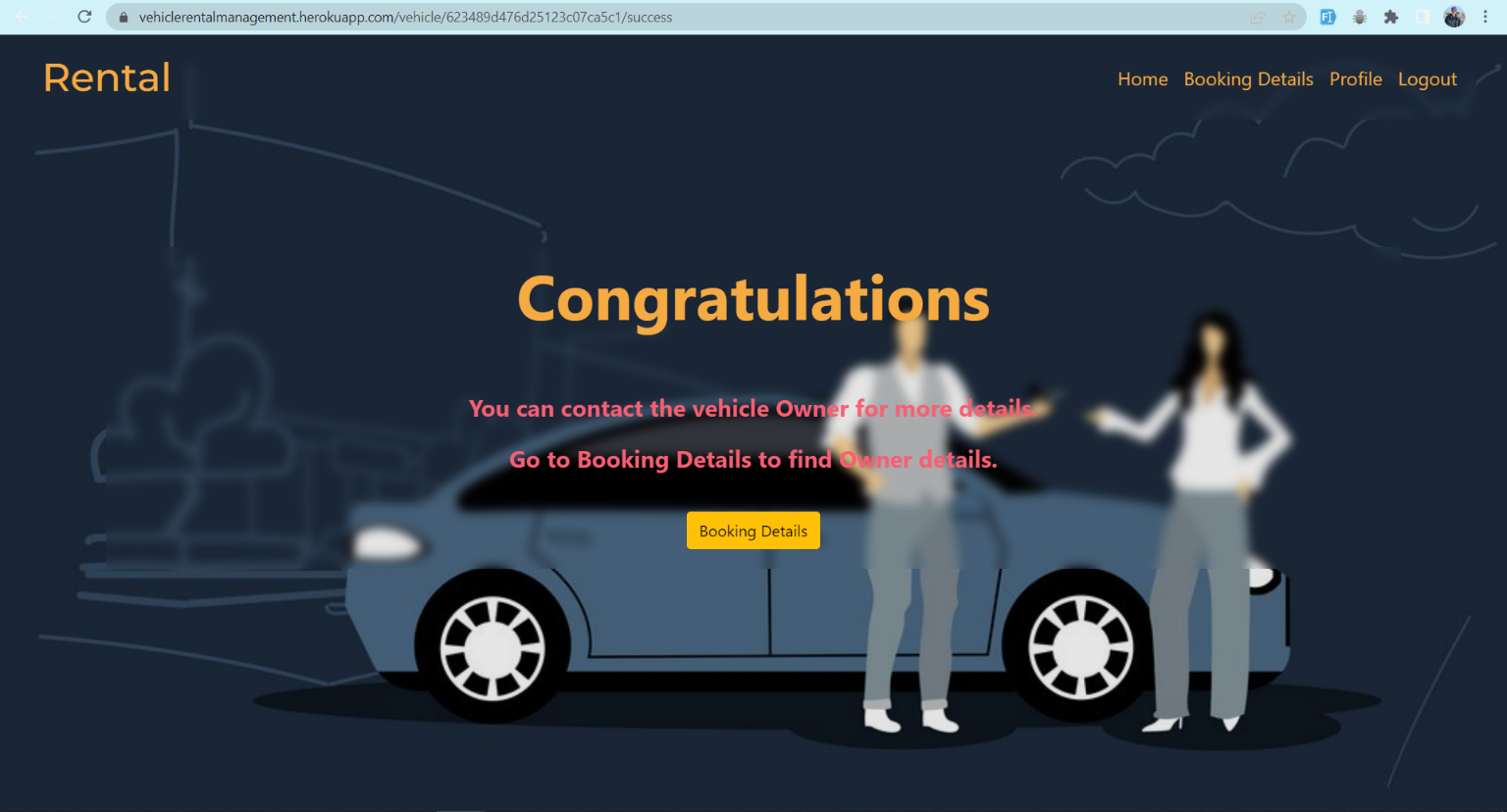
* on clicking movie card of particular movie, user is redirected to all details page of that particular movie.



### Add to watchlist:

* on clicking add to watchlist button, logged in user can add movie to their watchlist





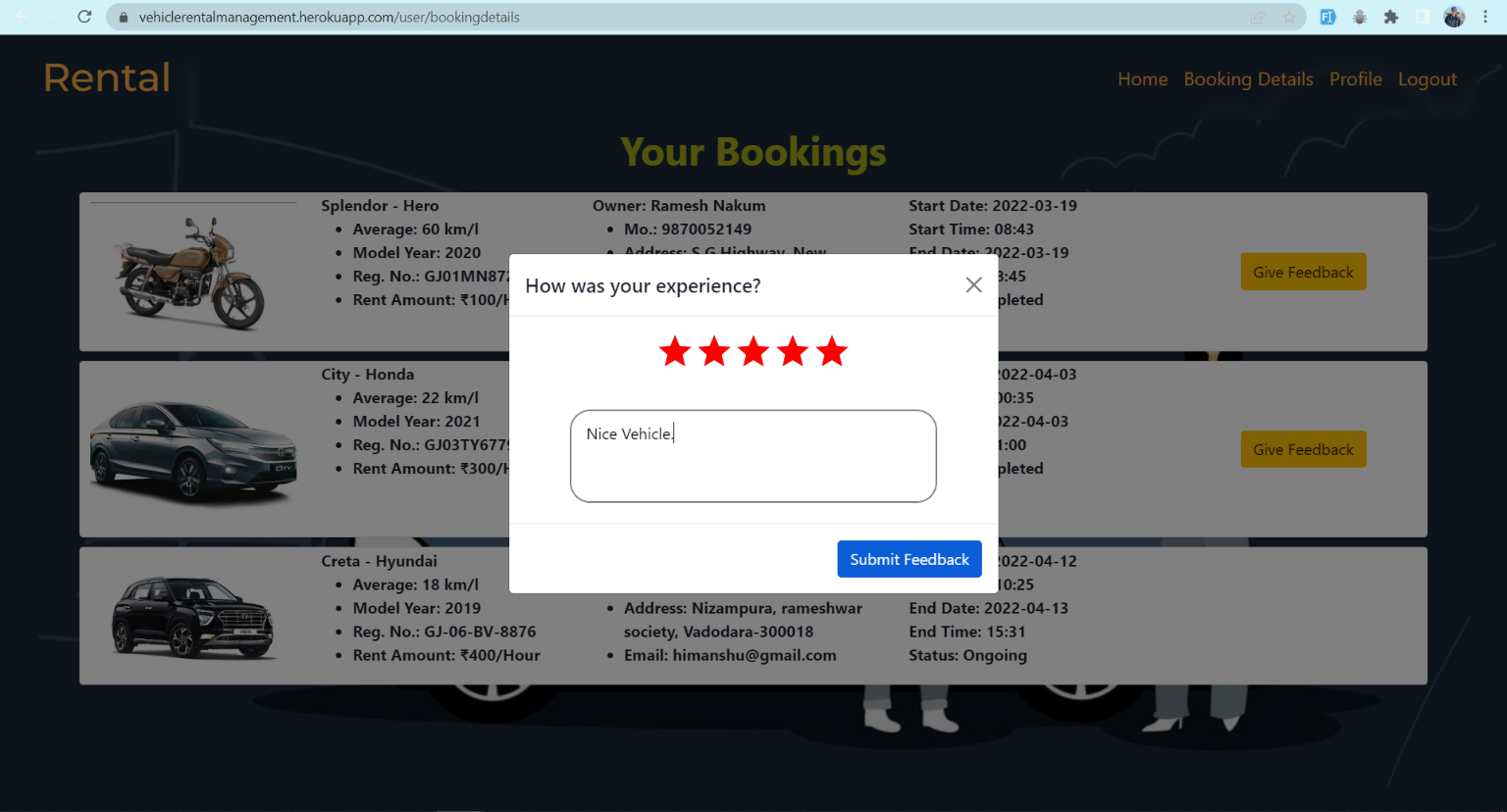
### Show Recommendations:

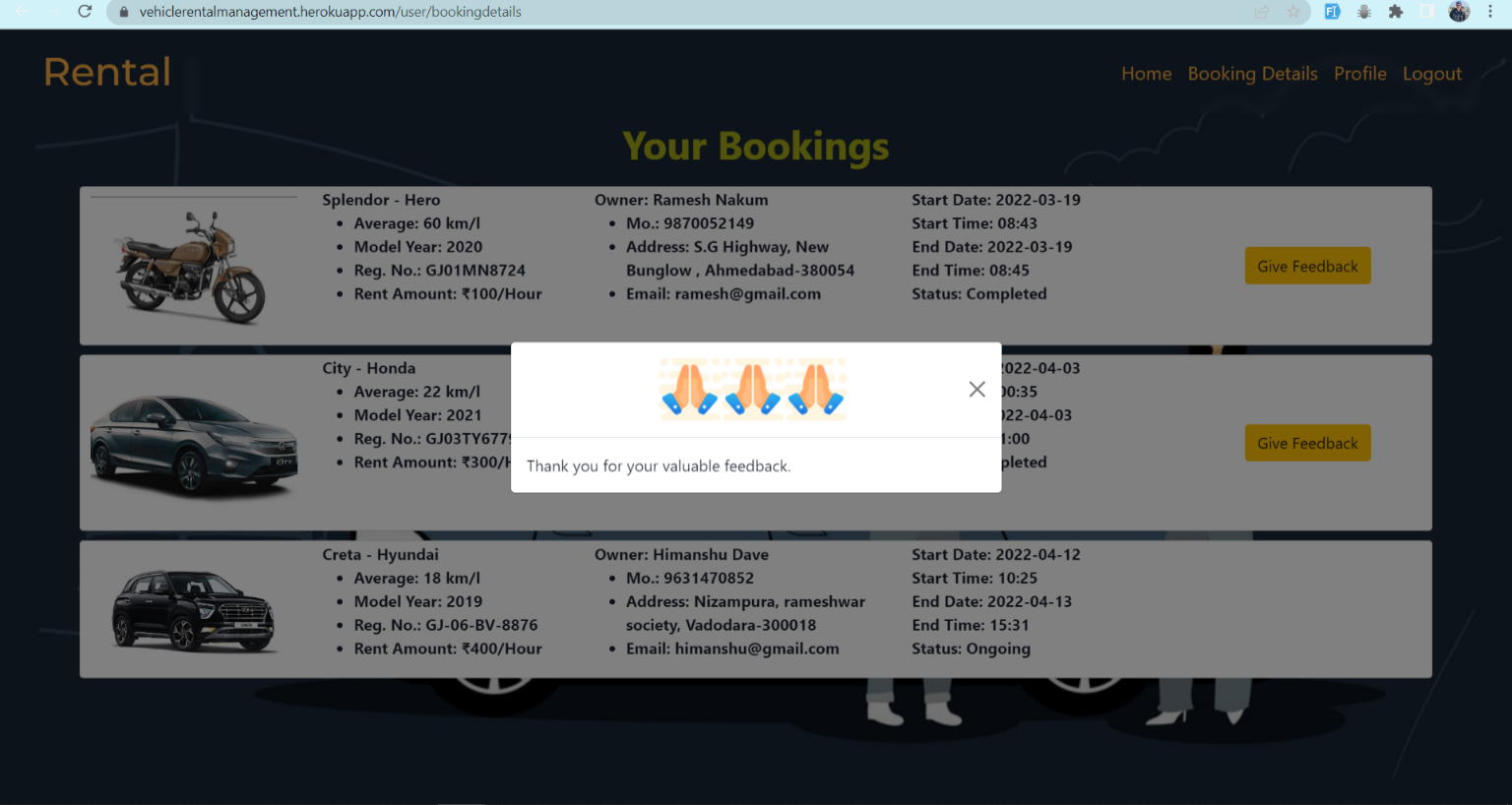
* On movie details page, there will be a section of recommendation movies based on that movie.
* User can see that recommended movie details by clicking on that.



### Movie by genre:

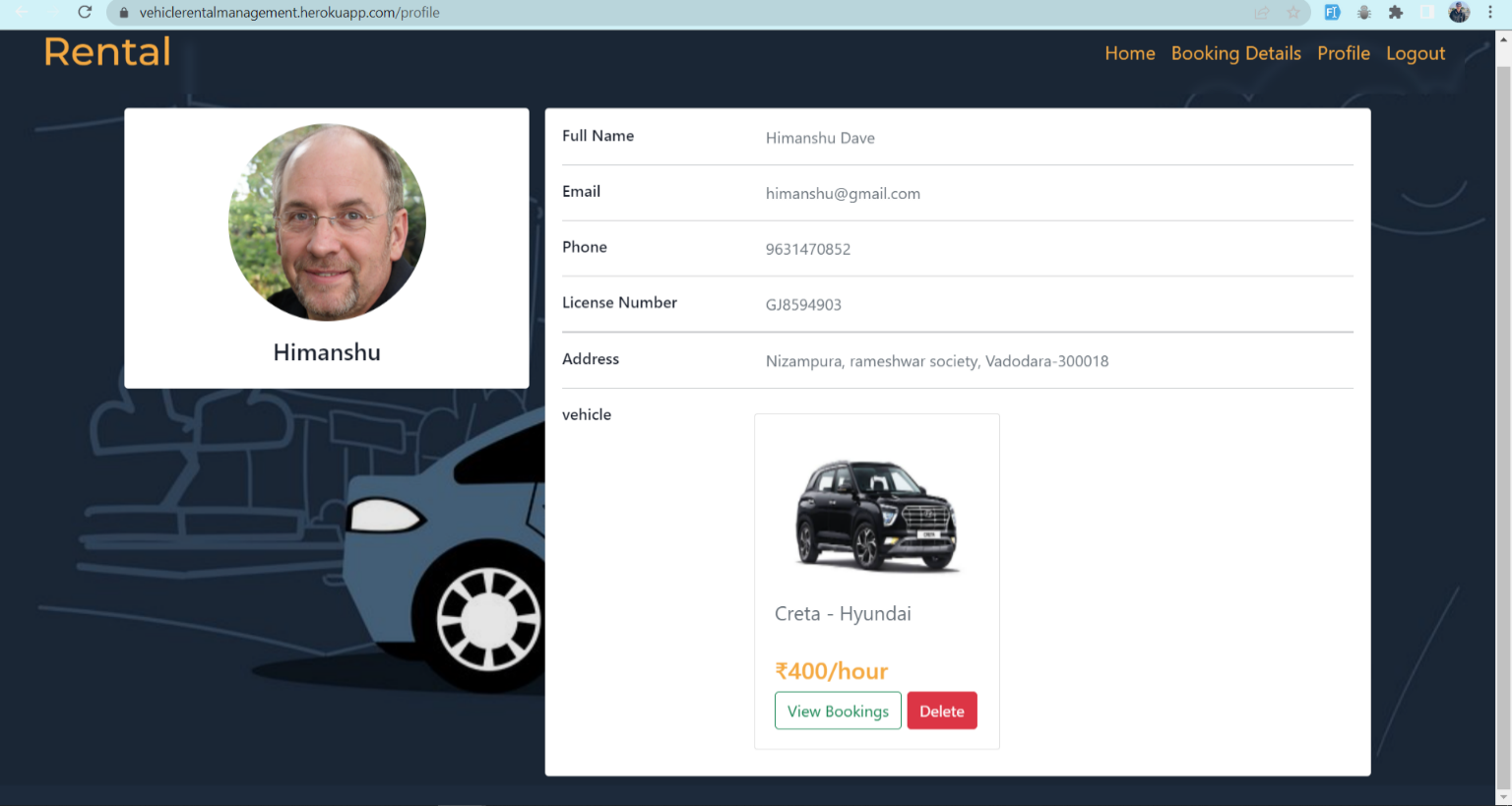
* On movie details page, there are movie genres buttons available showing genres of the movie.
* On clicking specific genre, user can find movies of that particular genre.





### Profile:

* On clicking profile button in Navbar, user is redirected to profile page.
* Here users can see all their details which they have provided at the time of registration.
* Here user can see their added movies to watchlist.
* By clicking on delete of particular movie they can remove their movie from watchlist.



## LIMITATION AND FUTURE ENHANCEMENT

### Limitation

* We have used already available API for data. If any error or problem occurrence at end of API server that affects our website too.
* I have implemented only the UI part of it. Backend is not done by me.

### Future Enhancement

* Better & easier to use UI will be prepared.
* There is a scope of add more information regarding movie like trailer of movie, find movie by actors/actress
* Add more images regarding movies.
* Functionality of see and add reviews of movie.

## CONCLUSION AND DISCUSSION

### Conclusion

According to us, this project gave all of us the confidence to believe in ourselves and a great experience of how to work as a team. It also boosted my requirement gathering, system analysis, designing aspects, technical coding as well as time management skills.

Also, I learned how to work together as a team & collaborate for making ends meet for our web app. I also got an insight into how we would have to work in future at job or startup & how we have to contribute for the good of the entity we are working.

### Discussion

#### Self-Analysis of Project Viabilities

According to me, this project is absolutely a good start for gaining hands-on experience on projects. It is useful if it is managed according to the goal for which it is made.

#### Problems Encountered and Possible Solutions

There are so many problems encountered during this project.

* + - * technical problems like maintenance of website, data redundancy
      * untechnical aspects like effective teamwork, better

communication of our team members’ individual ideas &

combine them for betterment of the website.

#### Summary of Project Work

It is a great achievement to successfully complete the project. The prior knowledge of software engineering has helped immensely in overcoming the various roadblocks. I have done work with pre-planned scheduling related with time constraints and weekly progress in project development. Also, I received guidance from Prof. Harshad B. Prajapati at all stages of my project which helped in overall betterment of the project.

All in all, it was a beautiful experience which taught me many things needed to succeed further in life during our jobs.

## REFERENCES

* Stackoverflow: [https://stackoverflow.com](https://stackoverflow.com/)
* Geeksforgeeks: <https://www.geeksforgeeks.org/>
* Youtube: <https://www.youtube.com/>
* W3 Schools: <https://www.w3schools.com/>
* React JS: <https://reactjs.org/docs/getting-started.html>
* Npm: <https://www.npmjs.com/>
* Bootstrap: <https://getbootstrap.com/docs/5.1/getting-started/introduction/>
* Font-awesome: <https://fontawesome.com/icons>
* TMDB API: https://developers.themoviedb.org/3/getting-started/introduction