

**VISVESVARAYA TECHNOLOGICAL UNIVERSITY
BELAGAVI, KARNATAKA, INDIA**



**A MINI-PROJECT REPORT
ON
“BIKE SHOWROOM MANAGEMENT SYSTEM”**

*Submitted in partial fulfillment of the requirement for the VI semester BE in
Information Science and Engineering
FS mini-project-18ISL67*

Submitted by

**ROHAN R VIBHUTI [1SG19IS083]
ROHIT B N [1SG19IS084]**

Under the guidance of

**Prof. Ambika S
Assistant Professor**



**DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING
SAPTHAGIRI COLLEGE OF ENGINEERING**

(Affiliated to Visvesvaraya Technological University, Belagavi & Approved by AICTE, New Delhi)
#14/5, Chikkasandra, Hesaraghatta Main Road, Bengaluru – 560057
(Accredited by NBA and NAAC with 'A' GRADE)
(An ISO 9001:2015 & ISO 14001:2015 Certified Institution)

2021-22

SAPTHAGIRI COLLEGE OF ENGINEERING

(Affiliated to Visvesvaraya Technological University, Belagavi & Approved by AICTE, New Delhi)

#14/5, Chikkasandra, Hesaraghatta Main Road, Bengaluru – 560057

(Accredited by NBA and NAAC with ‘A’ GRADE)

(An ISO 9001:2015 & ISO 14001:2015 Certified Institution)

Department of Information Science & Engineering



CERTIFICATE

Certified that the project work entitled "**BIKE SHOWROOM MANAGEMENT SYSTEM**" carried out by **ROHAN R VIBHUTI [1SG19IS083], ROHIT B N [1SG19IS084]** bonafide students of 6th semester, department of **Information Science & Engineering** carried out at our college **Sapthagiri College of Engineering**, Bengaluru in partial fulfillment for the 6th Semester BE, FILE STRUCTURES Mini-Project-18ISL67 by **Visvesvaraya Technological University**, Belagavi during the year 2021-22. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library. The project report has been approved as it satisfies the academic requirements in respect of Mini-Project work prescribed for the said Degree.

Signature of the Guide
Prof. Ambika S
Assistant Professor

Signature of the HOD
Dr H R Ranganatha
HOD, I.S.E., S.C.E.

External Viva

Name of the Examiners

1.
2.

Signature of the Examiners with date

- 1.....
- 2.....

ABSTRACT

The purpose of Bike Showroom Management System is to automate the existing manual system by the help of computerized equipments and full-fledged computer software fulfilling their requirements so that their valuable data/information can be stored for a longer period with an easy accessing and manipulation of the same. Bike Showroom Management System, as described above, can lead to error free, secure , reliable and fast management system. The Aim of this project is to automate its existing manual system by the help of computerized equipments and full fledged computer software, fulfilling their requirements, so that their valuable data/information can be stored for a longer period with easy accessing and manipulation of the same. Basically the project describes how to manage for good performance and better services for the clients. Our “**BIKE SHOWROOM MANAGEMENT**” is similar to this system but with greater power of file structures. The system provides a web interface for users to register, select, book and also choose the bikes of their choice. It depicts the power of files even after such well-established database systems have come into existence.

ACKNOWLEDGEMENT

Any achievement doesn't depend solely on the individual efforts but on the guidance, encouragement and co-operation of intellectuals, elders and friends. A number of personalities have helped us. We would like to take this opportunity to thank them all.

We would like to express our heart-felt gratitude to **Dr. H Ramkrishna**, Principal, Sapthagiri College of Engineering, Bengaluru, for his help and inspiration during the tenure of the course.

It is great privilege to extend our deep sense of gratitude to **Dr. H R Ranganatha**, Head of the Department, Information Science and Engineering, Sapthagiri College of Engineering, Bengaluru, who patronized throughout our career, for his constant support and encouragement and for the facilities provided to carry out this work successfully.

We wish to express our sincere thanks to our guide **Prof. Ambika S**, Assistant Professor, Information Science and Engineering, Sapthagiri College of Engineering, Bengaluru for helping us throughout and guiding us from time to time.

We also extend our sense of gratitude and sincere thanks to all faculty members and non-teaching staff members of Information Science and Engineering, Sapthagiri College of Engineering, Bengaluru for their views and encouraging ideas.

Finally, we also thank our family and friends for their co-operation and motivation.

**ROHAN R VIBHUTI [1SG19IS083]
ROHIT B N [1SG19IS084]**

TABLE OF CONTENTS

Chapter No.	Chapter Name	Page No.
1.	INTRODUCTION	1
1.1.	Introduction to File Structures	1
1.1.1.	History	2
1.1.2.	About the File	2
1.1.3.	Various Storage Kind of Fields and Records	3
1.1.4.	Applications of File Structures	4
2.	SYSTEM ANALYSIS	6
2.1	Analysis of System Application	6
2.1.1.	User-Side Analysis	6
2.1.2.	Admin-Side Analysis	6
2.2.	Structure used to store fields and records	7
2.3.	Operations performed on files	7
2.4.	Indexing	9
3.	SYSTEM DESIGN	10
3.1.	Design of the Fields and Records	10
3.2.	User Interface	11
3.2.1.	Insertion of a Record	15
3.2.2.	Display of Record	16
3.2.3.	Deletion of a Record (Admin Side Only)	17
3.2.4.	Add a Record (Admin Side Only)	17
4.	IMPLEMENTATION	16
4.1.	About Python Flask	16
4.1.2.	Features	16
4.2.	Amazon Web Services	17
4.3.	Amazon Elastic Beanstalk	17
4.4.	Pseudo Code	17
4.4.1.	User Side	18
4.4.2.	Admin Side	19

4.5.	Testing	20
4.5.1.	Unit testing	20
4.5.2.	Integration Testing	21
4.5.3.	System Testing	22
4.6.	Discussion of Results	23
4.6.1	Admin Dashboard	23
4.6.2.	Search Bikes	24
4.6.3.	Delete Bikes	25
4.6.4.	Modify/Add Bikes	26
4.6.5.	Add New Bikes	27
4.6.6.	New Added Bikes are viewed by Admin	27
4.6.7.	User Dashboard	28
4.6.8.	Choose the bike model	29
4.6.9.	Bike Booked details	29
4.6.10	Snapshots	30
4.6.11	Bike Gallery	32
4.6.12	File Contents	35
5.	CONCLUSION AND FUTURE ENHANCEMENTS	
	REFERENCES	38
		39

LIST OF FIGURES

Figure no	Name of the figure	Page no
3.2.1	Insertion of a record	11
3.2.1	Insertion of a record in website	12
3.2.2	Display a record	12
3.2.2	Display a record in website	13
3.2.3	Deletion of a record	13
3.2.3	Confirmation of a deleted record	14
3.2.4	Adding of a record	14
3.2.4	Addition of a record in text file	15
3.2.4	Modified record after addition	15
4.5.1	User login	20
4.5.1	users.txt file	21
4.6.1	Admin login page	23
4.6.1	Add new Bike details	23
4.6.2	Searching	24
4.6.2	Admin can search bikes from bikes.txt	24
4.6.3	Before deleting	25
4.6.3	After deleting	25
4.6.4	Before modification	26
4.6.4	After modification viewed in bikes.txt file	26
4.6.5	Add new Bikes	27
4.6.6	New added Bikes	27
4.6.6	Registered users for booking a bike	28
4.6.7	User login page	28
4.6.8	Booking a Bike	29

4.6.9	Bike Booking details	29
4.6.10	Snapshots	30
4.6.10	Home page	30
4.6.10	Bike Agencies	32
4.6.11	Bike Gallery	32
4.6.12	users.txt	35
4.6.12	admin.txt	35
4.6.12	bikes.txt	36
4.6.12	bookings.txt	36
4.6.12	file to be compressed	37
4.6.12	compress.txt	37

CHAPTER 1

INTRODUCTION

1.1. Introduction to File Structures

File Structures is the organization of Data in Secondary storage Device in such a way that minimizes the access time and the storage space. A File Structure is a combination of representations for data in files and of operations for accessing the data. File Structure allows applications to read, write and modify data. It also supports finding the data that matches some search criteria or reading through the data in some particular order. An improvement in file structure design may make an application hundreds of times faster. The details of the representation of the data and the implementation of the operations determine the efficiency of the file structure for particular applications.

The goal of File Structure is to get the information we need with one access to the disk. If it is not possible, then get the information with as few accesses as possible. Group information so that we are likely to get everything we need with only one trip of the disk. It is relatively easy to come up with File Structure designs that meet the general goals when the files never change. When files grow or shrink when information is added and deleted, it is much more difficult.

Goal of this course is with reference to time and space is to first minimize number of trips to the disk in order to get desired information. Ideally get what we need in one disk access or get it with a few disk accesses as possible. Secondly grouping related information so that we are likely to get everything we need with only one trip to the disk for example name, address, phone number, account balance.[1][5]

Good File Structure design must have:

- Fast access to great capacity.
- Reduce the number of disk accesses.
- By collecting data into buffers blocks or buckets.
- Manage growth by splitting these collections.
- Waiting time for the applications becomes less for the disk.

1.1.1 History

- History of File Structure design, In the beginning the access was sequential, and the cost of access grew in direct proportional to the size of the file, so Indexes were added to files.
- Indexes made it possible to keep a list of keys and pointers in a smaller file that could be searched more quickly.
- Simple indexes become difficult to manage for dynamic files in which the set of keys changes. Hence Tree Structures were introduced.
- Trees grew unevenly as records were added and deleted, resulting in long search requiring multiple disk accesses to find a record. Hence an elegant, self-adjusting binary tree structure called an AVL Tree was developed for data in memory.
- Even with a balanced binary tree, dozens of accesses were required to find a record in moderate sized files.
- A method was needed to keep a tree balanced when each node of the tree was not a single record as in a binary tree, but a file block containing hundreds of records. Hence B-trees were introduced.
- AVL trees grow from top down as records are added, B-trees grow from the bottom up.
- B-trees provided excellent access performance but, a file could not be accessed sequentially with efficiency.
- The above problem was solved using B+ tree which is a combination of a B-tree and a Sequential linked list added at the bottom level of the B-tree.
- To further reduce the number of disk accesses, Hashing was introduced for files that do not change size greatly overtime.[1][3]

1.1.2 About the File

A File is an object on a computer that stores data, information, settings, or commands used with a computer program. In a graphical user interface such as Microsoft Windows, files display as icons that relate to the program that opens the file. For example, the picture is an icon associated with adobe acrobat PDF files, if the file was on your computer, double clicking the icon in Windows would open that file in adobe acrobat or the PDF reader installed on the computer.

A File is created using a software program on the computer. For example, to create a text file we use text editor, to create an image file we use an image editor, and to create a document we use a word processor.

Files are not made for just reading the contents, we can also perform some operations on the Files.

- **Read operation:** Meant to read the information which is stored into the files.
- **Write operation:** For inserting some new contents into a file.
- Rename or Change the Name of the file.
- Copy the file from one location to the other.
- Sorting or arrange the contents of the file.
- Move or cut the file from one place to another.
- Delete a file.
- Execute Means to Run Means File Display Output We can also link a file with any other File.

These are also called Symbolic Links in the symbolic links all the files are linked by using some text Alias.

We can also link a file with any other File. These are also called Symbolic Links in the symbolic links all the files are linked by using some text Alias.

1.1.3 Various storage kinds of fields and records

Field structures:

There are many ways of adding structure to files to maintain the identity of fields. Four most common methods follow:

- Method 1: Force the fields into a predictable length. Fix the length of fields. The fields of the file vary in length to make the fields fixed length we have to predict lengths.
- Method 2: Begin each field with a length indicator, store the field length just ahead of the field.
- Method 3: Place a delimiter at the end of each field to separate it from the next field.
- Method 4: Use a “keyword = Value” expression to identify each field and its contents.

Record structures:

A record can be defined as a set of fields that belong together when the file is viewed in terms of a higher-level organization. Five most common methods follow:

- Method 1: Make the records be a predictable number of bytes in length.
- Method 2: Make the record be a predictable number of fields in length.
- Method 3: Begin each record with a length indicator consisting of a count of the number of bytes that the record contains.
- Method 4: Use a second file to keep track of the beginning byte address for each record.
- Method 5: Place a delimiter at the end of each record to separate it from the next record.

1.1.4 Application of file structure

Google Files:

Files is Google's latest app. It's free to use, lightweight (only takes up 10MB of space), and offers three core features: clean out junk files and cache, to boost phone memory and performance; browse your storage and find everything with a few taps; send pictures, share videos, and transfer large files at a fast speed (rate up to 480Mbps over encrypted direct Wi-Fi network.[1][2][5]

Amaze File Manager:

Amaze File Manager is an open-source file management tool that lets you explore all the directories on your Android device, move files and folders, rename documents, copy and paste files, and so on.

Asus file manager:

It's not every day we see an OEM app make an app list, but File Manager by ASUS is actually really good. It's compatible with most devices, even non-ASUS ones. You'll also get clean, simple interface with LAN and SMB support, cloud storage support, support for various types of files, archiving support, and more. It's entirely free with no in-app purchases and provides a great experience for a simple file browser. About the only negative part is the lack of root access.

File manager:

File Manager is blandly named, but it's actually quite good. It's a newer file manager app that gives you one of the best sets of features without adding too much bloat. You'll get basic file management features along with cloud storage features, NAS support, and more.

You can even browse your installed apps, music, and video with this and the player isn't half bad. Perhaps the best part is that the app is free with no in-app purchases and advertising

ES File Explorer:

It was one of the powerful apps that worked with files. It was banned in 2020 due to its privacy issues. It is almost known to all android users. This was a major application of files. It used most advanced functions very prior to all the other file managers even existed.

MK explorer:

MK Explorer is another newer file manager option. It's a simple option that doesn't have a whole lot of flair. That is extremely preferable if you really just want something simple. It features a Material Design interface, the basic file management features (copy, paste, delete, SD Card support for Lollipop), and root access. There are also supports for 20 languages and it has a built-in text editor, gallery, and music player. It doesn't have anything like cloud storage or network storage support, but that's not really what it's for. It's a good, cheap option.

Xplore file manager:

Xplore manager is one the more unique options on the list. It's a forced dual pane app which means you'll be managing two windows at once pretty much all the time. This is kind of cool if you're copy/pasting between folders or need to move files quickly. It also comes with support for various types of files, cloud storage, network storage, a built-in hex editor, root support, and plenty of other features.

CHAPTER 2

SYSTEM ANALYSIS

2.1 Analysis of application

‘BIKE SHOWROOM MANAGEMENT SYSTEM’ has a user-friendly web interface which can be accessed from anywhere in the World.

The front-end is written using HTML, CSS, JavaScript and Bootstrap.

The back-end managed by a Python Flask Server.

2.1.1 User-Side Analysis:

The landing page provides users with a Signup System and Login System with simple Name Details field authorization. All user details are stored in a user file.

If the user doesn’t have an account in the website ,he need to create one using signup process where the user needs to enter his name ,mail and password which is used for authorization of users login. Upon successful login, a ‘user’ cookie is set from the server side and the user is displayed with a bike marketing page where he gets to select his desired bike ,upon clicking checkout, the user is directed to a bike booking page where he needs to enter his personnel details and select the bike which he wants to buy, after all the process booking is confirmed.

Whenever a bike is booked by the customer a unique Booking ID is generated to different customers and the users entered personal data are stored in the admin dashboard and in the users.txt file where the data can be modified .

If the user wishes to cancel the booking of a bike ,he needs to contact the admin for further process

2.1.2 Admin-Side Analysis:

The landing page provides admin with a Login system with password field authorization.

After the authorization, admin logs successfully. As soon as the admin logs with his password a ‘admin’ cookie is set from the server side and the admin is directed to a admin dashboard where he is provided with option such as viewing users data ,bike bookings ,and functions such as adding data where the admins add new data to the bikes market as of new arrivals.

The admin is provided with features such as deleting and manipulating data inside a bike data, The admin can delete any data based on the unique id generated while adding a data .By taking booking ID as a key the admin can delete any specific data

2.2 Structure used to store fields and records

- A field is an item of stored data. A field could be a name, an address, a description etc.
- A record is the collection of fields that related to a single entity.

All details accepted during the login of a user is written onto a users.txt.

The bike details are added by the admin to the bikes.txt file.

Each record is separated by a ‘\n’ (next line) character and each field in the record is separated by a ‘|’(delimiter).

User Details are saved as follows:

ID| Name | Hashed password | Mobile no | Gender | Gmail\n

Admin Details are saved as follows:

ID| Name | Hashed password | Mobile no | Gender | Gmail\n

Bikes details are saved as follows:

Unique ID| Company Name| Ex Price| On road Price| Color| IN/OUT Stock |ABS/
NON-ABS|CC| Mileage| Tank Capacity|\$

2.3 Operations performed on files

A file is a collection of logically related data that is recorded on the secondary storage in the form of sequence of operations. The content of the files are defined by its creator who is creating the file. The files are used to store the required information for its later uses. There are many file operations that can be perform by the computer system. These operations are performed by the user by using the commands provided by the operating system.[1]

Some common file operations:

- File Create operation
- File Delete operation
- File Open operation
- File Close operation

- File Read operation
- File Write operation
- Binary Search operation
- File Append operation

File Create operation

- The file is created with no data.
- The file create operation is the first step of the file.
- Without creating any file, there is no any operation can be performed.

File Delete operation

- File has to be deleted when it is no longer needed just to free up the disk space.
- The file delete operation is the last step of the file.
- After deleting the file, if doesn't exist.

File Open operation

- The process must open the file before using it.

File Close operation

- The file must be closed to free up the internal table space, when all the accesses are finished and the attributes and the disk addresses are no longer needed.

File Read operation

- The file read operation is performed just to read the data that are stored in the required file.

File Write operation

- The file write operation is used to write the data to the file again, generally at the current position.

Binary Search operation

- The Binary search operation is used to search the data from the file based on key.

File Append operation

- The file append operation is used to add records on to the end of the record.

2.4 Indexing:

An Index is a tool for finding records in a file.

It consists of:

- Key field on which the index is searched.
- Reference field that tells where to find the data file record associated with particular key.

We have used a simple indexing as well as secondary indexing:

- We choose to organize the file as variable length record with a primary key field at the beginning of each record. The fields within each record are also of variable-length but are separated by delimiters.

Bookings details are saved as follows:

Unique ID| Customer Name| Bike Model| Mobile No| Gender| Gmail |Address|\$

Compressed File details are saved as follows:

ENq9mUlV2zgUx+zKQjMNTG4iBR5dJwm6Zl4sD3dbqzDjkIVKZDldALLow9FeREIloKzKnowf+f3nt8ayBExSJ9kzF4l/yMVpxQ3ETPMpZvMinhwMNP0483S/Dua0EmcPef+d1iPrswHUmX1Ws/lFvxflkk6frX8X0YlkECI HZrCAQlly9/Qg+qp4+8EaX9SYJMWWhksEM3BUPEn6J/Uqd8krhUu13phvczD/CZaGdTcO/fyCEYD/Ghikn6G3x5ivKKI+55XIZAYltlq3KVZoBQqGFHFp4EtluXuaz+ydjeef10r80JfonURL2JzE7VstcxhHMtCOIRn SiWEBPovqAtg4POmEh7Q1r+G+5fVFZkr5KcJglvOePm7ar3EWy80mWncBr7cJ/j5yC4/iUwMS4ThWNYy9zLnLhpQGVjK5KEPDUE+2rTyCvr+u0t81MsJBypmpUkeJsoxLioxTTxDWQD0Ngs2uwCzW0mUrNvGKh6GD R5BcDivXbG0OSvZiMGna4lkokQFvliJo81Q+GJXYJcyND2abqKFpkklur1wnRgJmfu054UstX3370umNpsOeymsY6wPKsZeeWczl5VGCBwTdpvZahs32Qu15Kfmlmk7c0Aa3Xe4qqXv/ZhHE7d7Y1jZvURBP Ba9lJ7Ear6ijmuflUoxBgRMbDq1+1C3hHQcAA4KAvCN/ksn2sXOPs2iQm4+ZHRKHmsOHWbEXWlfB3iDLqjMCn1Rm0Skugh5lyAR50/qVJD0AQveGbxpBrAtrlVNb78BN0XpeGEJZPW2TsgWYhiOTwbOpB8btzhRz1hj 9gXfY1BdGd+vfbkKyn4rlLnSjVledk+5iCAQg07Qrk17zNKKhjme8oBpn36P2RBByWBDZ4r2jhhBbo5pwdOo7+ew0Ab9lOJow6vZVa9pAgSgB9EbjDADVrtgAcB00NgGaoAHQquaG4LoVC/kGvi+KMDwtLECwRg p1xRo4ZHsdvVrv9p97Ftk4bGFl5cVp2eWLzH7FqtZTD9/tbey+r7OazlflOab3jD+WmeXr846wetuAhvWBETffGm4X5KwcebgYzVqw+14vFQs9D+IWG9MSCVhXY1nKBsFW/TdEglcEnp c3J6RXIMon2z6FwinPMR/3 5e4WSGGQJLHaM6YePymmwWC1007UTGU1C2Bp8rehCaNVUbLY8iPEYBu-LczMQjimY90pmsWl3TvLdCeOkDUoshf6sauL8Wab1lO/cxAlwSM113G62bf6f5MRJlJdc5pp04GjeElxomAuDGm7DPkw/ZF/ZYQn4GdQE o6x8EWG3B7Qvecg2x6yqySoapGEFodpg+qY9epxyHr3atM1/3KRCyHLBTluntr8Hf/Bd87Y7nLb3B1b+A+3m4KSiexQCSNNzgl_69JgE8WvGm-91Plu86ukUcjhl3londlo3KPF/WY1hQSPAu5sn92V9woNmNuUp1r700ZV JCYipquibLwV09yu.lurWRQkdGWK1CVKsSp2wSIXSA2TDp2TiePmlam9DtKultr26+wAMhcmAhzFvZOXiuUhbXfgve+x-HW8Q0dGce9FNageBwOoTjmmlPXT9hDEdy9CL6m6pFnVevulb+Ldx9lhEg1NhcYU9r oHoD+q9ghsADBCwwC0j/dpE2jbzl0+4IMvqlW0K393k0tXErCTFmyYdC3HbL94AEov/1mfns5LWYXX0cePqdhd4ZNceoAbKO7uZXZ+YWXKnfpfmcqC9UhLqY0Rlx0+QmjAGoR7AyjZ4Q9AcU4Ook8/YvmgCoH09 G5YEVirRQQPKEOcVGzolu8INze5hzqocwPAZus6fVvnYYxHsMy7CCgCx8zLuonWkhkifgyT9s1ctynA7+4vl2qP6G03LKK3p6pmLUzZKJLlu0vdBdErWrXui+mh6Yfim2OkjQDgcgIRUT/UNA7h7anPabsV0agzFI xQby8qvUQy92ZzN1Y7jsMON8NI63RSBWluCv5kRP4bYz9D7tp85Zs/kAHHV2067d4DVZH15q/RQ9W/qThOf/cQxzAYid+YYE7h9QOiyWo+hQj9WPluDtJa+DPKz+OpMvZsd0OeoghADDWjWYL+2HielfabsaiO9 2UiFYDXG3R1gtVsDOgxtrjkfKB81MIHB5Ulw0IF3AYxJUATmXB0TDL1H8AHqd4QQMjy9/+qr0KtC5tDktWknjlpsmwDG+YkdwbRqfQk5wx+XHjoAThdn8Q1CrWjg5fxA/sBn7sxwsqrBwGld+ON3Nc+a4oWt

CHAPTER 3

SYSTEM DESIGN

3.1 Design of the fields and records

“Bike Showroom Management System” is to manage the details of Bike, Company, Type , Customer , Booking . It manages all the information about Bike , Payment ,Booking, Bike .The Project is totally built at administrative end and thus only the administrator is guaranteed the access to Add and Update the data of Bikes. The purpose of the project is to build an application program to reduce the manual work for managing the Bike, Company, Type .It Tracks all the details about the Type ,Customers and Booking and also manages the information of the companies.

BIKE SHOWROOM

1. Home
2. About
3. Companies
4. Gallery
5. Admin
6. Bikes

1. **Login page:** Login consist of two parts:

1. Admin login page
2. User login page

2. **Admin Dashboard:** Admin part has the following functionalities:

1. Can view the number of bikes available and as well as the customers details in the record.
2. Add new bikes to the record.
3. Delete the bikes from the record.
4. Can view the registered users.
5. Can also view the details of customers.
6. Can view the bookings log.

3. **Bikes Dashboard for customers:** User part has the following functionalities:

1. Can view different Bikes Before he/she buys.
2. Can Choose Bikes from the record.

3.2 User Interface

- The user interface (UI), in the industrial design field of human-computer interaction, is the space where interactions between humans and machines occur.
- The goal of this interaction is to allow effective operation and control of the machine from the human end, whilst the machine simultaneously feeds back information that aids the operators' decision-making process.
- Examples of this broad concept of user interfaces include the interactive aspects of computer operating systems, hand tools, machinery operator controls, and process controls. The design considerations applicable when creating user interfaces are related to or involve such disciplines as ergonomics and psychology.
- The junction between a user and a computer program. An interface is a set of commands or menus through which a user communicates with a program.
- A menu driven interface is one in which you select command choices from various menus displayed on the screen.

3.2.1 Insertion of a record

- A new record is being inserted into the users.txt file when user logs in with his credentials.

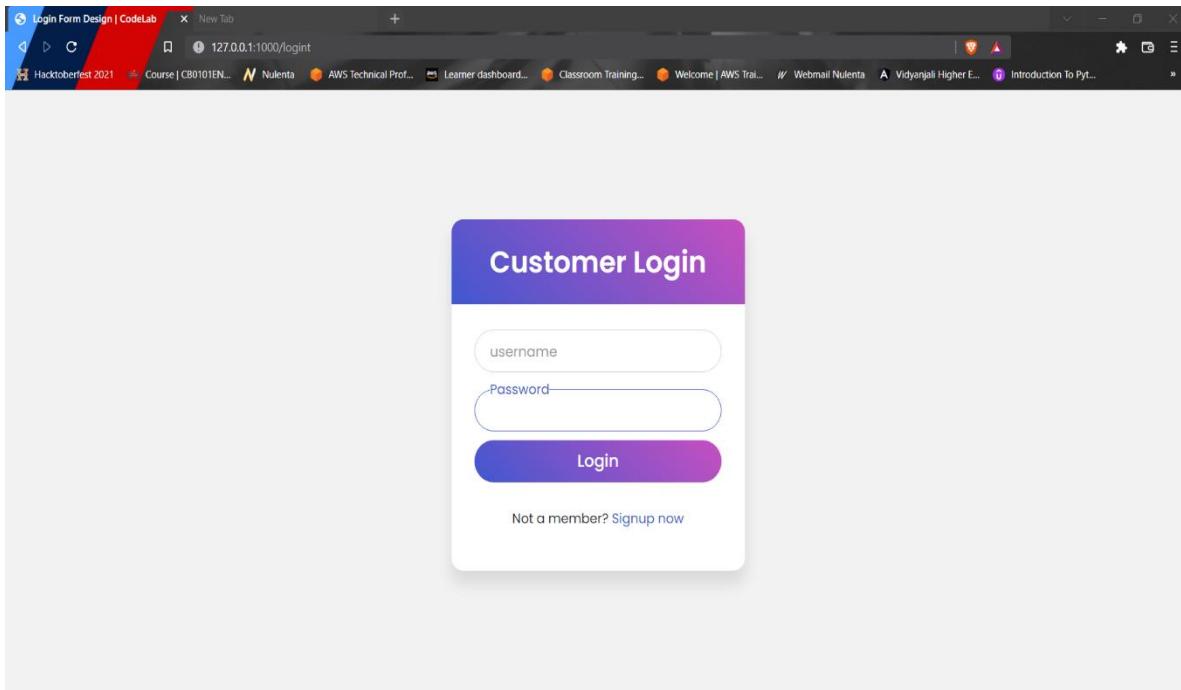
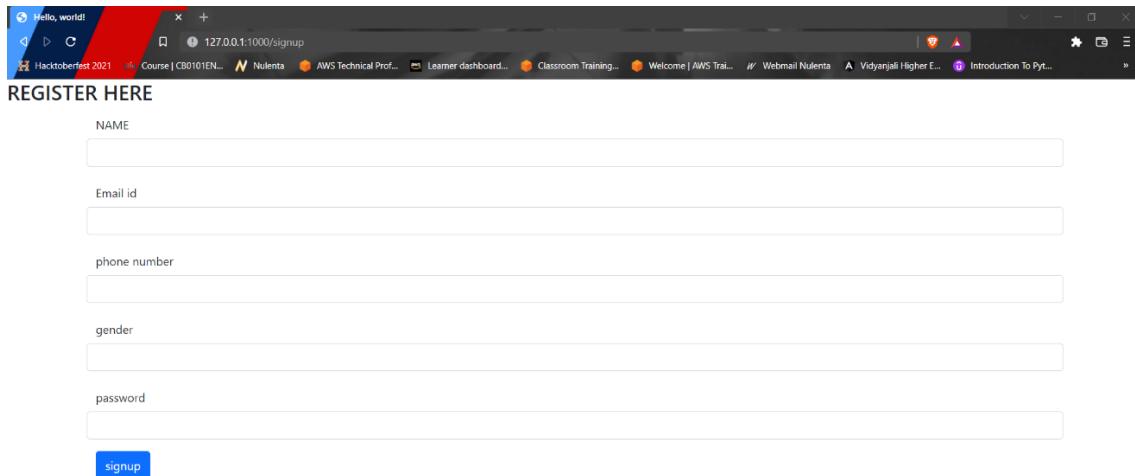


Fig 3.2.1 Insertion of a record

- If Not a User, he/she should signup for accessing the bikes data and this data is stored in users.txt file.



REGISTER HERE

NAME

Email id

phone number

gender

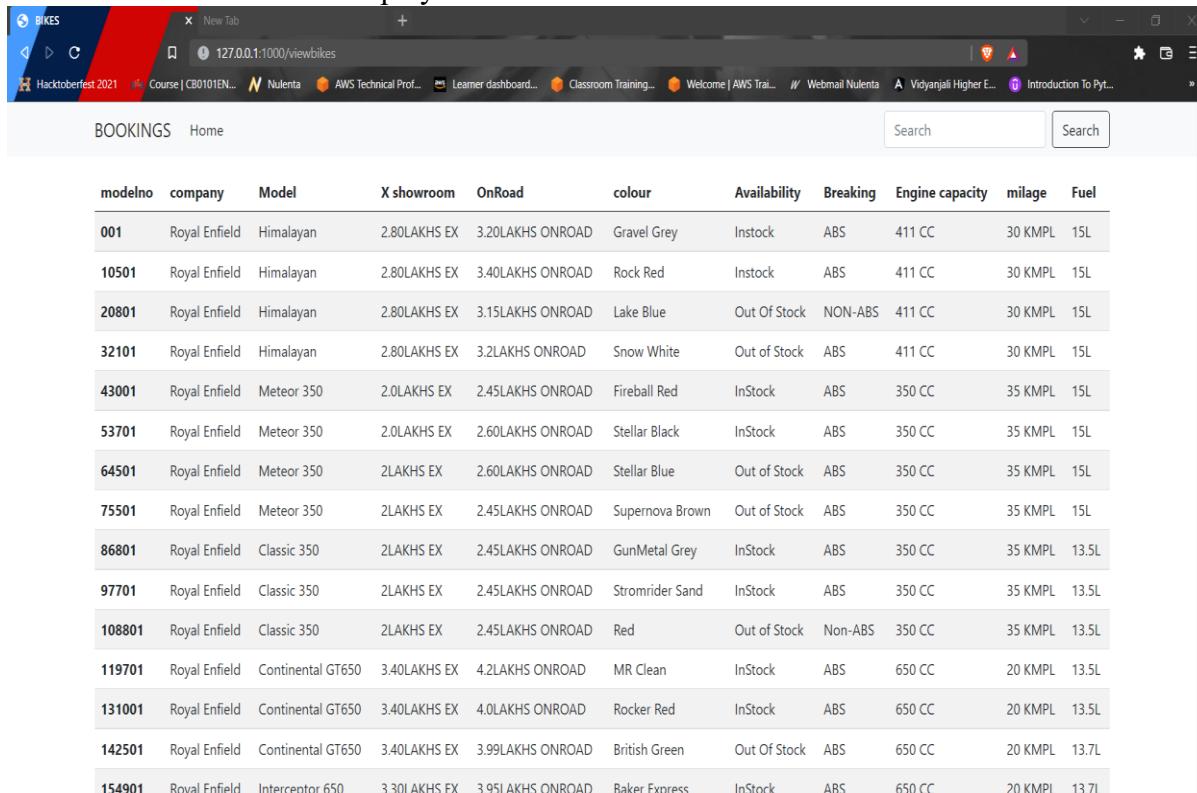
password

signup

Fig 3.2.1 Insertion of a record in website

3.2.2 Display a record

- Bikes Details are displayed for the users from the bikes.txt.



modelno	company	Model	X showroom	OnRoad	colour	Availability	Breaking	Engine capacity	mileage	Fuel
001	Royal Enfield	Himalayan	2.80LAKHS EX	3.20LAKHS ONROAD	Gravel Grey	Instock	ABS	411 CC	30 KMPL	15L
10501	Royal Enfield	Himalayan	2.80LAKHS EX	3.40LAKHS ONROAD	Rock Red	Instock	ABS	411 CC	30 KMPL	15L
20801	Royal Enfield	Himalayan	2.80LAKHS EX	3.15LAKHS ONROAD	Lake Blue	Out Of Stock	NON-ABS	411 CC	30 KMPL	15L
32101	Royal Enfield	Himalayan	2.80LAKHS EX	3.2LAKHS ONROAD	Snow White	Out of Stock	ABS	411 CC	30 KMPL	15L
43001	Royal Enfield	Meteor 350	2.0LAKHS EX	2.45LAKHS ONROAD	Fireball Red	InStock	ABS	350 CC	35 KMPL	15L
53701	Royal Enfield	Meteor 350	2.0LAKHS EX	2.60LAKHS ONROAD	Stellar Black	InStock	ABS	350 CC	35 KMPL	15L
64501	Royal Enfield	Meteor 350	2LAKHS EX	2.60LAKHS ONROAD	Stellar Blue	Out of Stock	ABS	350 CC	35 KMPL	15L
75501	Royal Enfield	Meteor 350	2LAKHS EX	2.45LAKHS ONROAD	Supernova Brown	Out of Stock	ABS	350 CC	35 KMPL	15L
86801	Royal Enfield	Classic 350	2LAKHS EX	2.45LAKHS ONROAD	GunMetal Grey	InStock	ABS	350 CC	35 KMPL	13.5L
97701	Royal Enfield	Classic 350	2LAKHS EX	2.45LAKHS ONROAD	Stromrider Sand	InStock	ABS	350 CC	35 KMPL	13.5L
108801	Royal Enfield	Classic 350	2LAKHS EX	2.45LAKHS ONROAD	Red	Out of Stock	Non-ABS	350 CC	35 KMPL	13.5L
119701	Royal Enfield	Continental GT650	3.40LAKHS EX	4.2LAKHS ONROAD	MR Clean	InStock	ABS	650 CC	20 KMPL	13.5L
131001	Royal Enfield	Continental GT650	3.40LAKHS EX	4.0LAKHS ONROAD	Rocker Red	InStock	ABS	650 CC	20 KMPL	13.5L
142501	Royal Enfield	Continental GT650	3.40LAKHS EX	3.99LAKHS ONROAD	British Green	Out Of Stock	ABS	650 CC	20 KMPL	13.7L
154901	Royal Enfield	Interceptor 650	3.30LAKHS EX	3.95LAKHS ONROAD	Baker Express	InStock	ABS	650 CC	20 KMPL	13.7L

Fig 3.2.2 Display a record

- Display of registered users are displayed from the Admin Side file.

Id	Name	phoneno	gender	gmail
1	tarun	987654321	M	tarun@gmail.com
2	gagan	654321987	M	gagan@gmail.com
3	harsha	65498712	M	harsha@gmail.com
4	pawan	545412516	m	pawan@gmail.com
5	ulla	4574231654	M	ulla@gmail.com
6	ambika	6897212	F	ambika@gmail.com
7	ambika1	548751	F	ambika@gmail.com
8	ashwin	06366617340	M	ashwin@gmail.com

Fig 3.2.2 Display a record in website

3.2.3 Deletion of a record(Admins Use Only)

- A record may be deleted according to users from bikes.txt.

modelno	company	Model	X showroom	OnRoad	colour	Availability	
001	Royal Enfield	Himalayan	2.80LAKHS EX	3.20LAKHS ONROAD	Gravel Grey	InStock	<button>Delete</button>
10501	Royal Enfield	Himalayan	2.80LAKHS EX	3.40LAKHS ONROAD	Rock Red	InStock	<button>Delete</button>
20801	Royal Enfield	Himalayan	2.80LAKHS EX	3.15LAKHS ONROAD	Lake Blue	Out Of Stock	<button>Delete</button>
32101	Royal Enfield	Himalayan	2.80LAKHS EX	3.2LAKHS ONROAD	Snow White	Out of Stock	<button>Delete</button>
43001	Royal Enfield	Meteor 350	2.0LAKHS EX	2.45LAKHS ONROAD	Fireball Red	InStock	<button>Delete</button>
53701	Royal Enfield	Meteor 350	2.0LAKHS EX	2.60LAKHS ONROAD	Stellar Black	InStock	<button>Delete</button>
64501	Royal Enfield	Meteor 350	2LAKHS EX	2.60LAKHS ONROAD	Stellar Blue	Out of Stock	<button>Delete</button>
75501	Royal Enfield	Meteor 350	2LAKHS EX	2.45LAKHS ONROAD	Supernova Brown	Out of Stock	<button>Delete</button>
86801	Royal Enfield	Classic 350	2LAKHS EX	2.45LAKHS ONROAD	GunMetal Grey	InStock	<button>Delete</button>
97701	Royal Enfield	Classic 350	2LAKHS EX	2.45LAKHS ONROAD	Stromrider Sand	InStock	<button>Delete</button>
108801	Royal Enfield	Classic 350	2LAKHS EX	2.45LAKHS ONROAD	Red	Out of Stock	<button>Delete</button>
119701	Royal Enfield	Continental GT650	3.40LAKHS EX	4.2LAKHS ONROAD	MR Clean	InStock	<button>Delete</button>
131001	Royal Enfield	Continental GT650	3.40LAKHS EX	4.0LAKHS ONROAD	Rocker Red	InStock	<button>Delete</button>
142501	Royal Enfield	Continental GT650	3.40LAKHS EX	3.99LAKHS ONROAD	British Green	Out Of Stock	<button>Delete</button>

Fig 3.2.3 Deletion of a record

- Admin can confirm that a record is deleted from the bikes.txt file.

```
bikes.txt - FS Mini Project - Visual Studio Code
File Edit Selection View Go Run Terminal Help
admin.html admin.txt bikes.txt
bikes.txt
44 453401|TVS|Apache RTR160|1.30LAKHS EX|1.55LAKHS ONROAD|white|out of Stock|Non-ABS|160 CC|40 KMPL|12L|#
45 463801|TVS|jupiter|78,04|EX|98,348 ONROAD|Royal Wine|InStock|UBS|110 CC|45 KMPL|4L|#
46 472501|TVS|jupiter|78,04|EX|98,348 ONROAD|blue|InStock|UBS|110 CC|45 KMPL|4L|#
47 489601|TVS|jupiter|78,045 EX|98,348 ONROAD|Peral White|out of Stock|UBS|110 CC|40 KMPL|4L|#
48 489901|TVS|jupiter|78,045 EX|98,348 ONROAD|Grey|out of Stock|UBS|110 CC|40 KMPL|4L|#
49 498501|TVS|Starmity Plus|55.069 EX|93,567 ONROAD|Grey silver|InStock|ABS|110 CC|40 KMPL|10L|#
50 508801|TVS|Starmity Plus|55.069 EX|93,567 ONROAD|Red|out of Stock|ABS|110 CC|40 KMPL|10L|#
51 517201|TVS|Wego|55.069 EX|80,789 ONROAD|white|InStock|UBS|110 CC|45 KMPL|4L|#
52 525101|TVS|Wego|55.069 EX|80,789 ONROAD|Blue|out of Stock|UBS|110 CC|45 KMPL|4L|#
53 533401|KTM|Duke 390|2.70LAKHS EX|3.90LAKHS ONROAD|white|InStock|ABS|390 CC|28 KMPL|13.4L|#
54 542601|KTM|Duke 390|2.70LAKHS EX|3.90LAKHS ONROAD|Silver|InStock|ABS|390 CC|28 KMPL|13.4L|#
55 551901|KTM|Duke 390|2.70LAKHS EX|3.90LAKHS ONROAD|Orange|out of Stock|ABS|390 CC|28 KMPL|13.4L|#
56 561701|KTM|Duke 390|2.70LAKHS EX|3.90LAKHS ONROAD|Metallic Silver|out of Stock|ABS|390 CC|28 KMPL|13.4L|#
57 572401|KTM|RC 390|2.80LAKHS EX|4.01LAKHS ONROAD|Metallic Grey|out of Stock|ABS|390 CC|26 KMPL|13.4L|#
58 582701|KTM|RC 390|2.80LAKHS EX|4.01LAKHS ONROAD|Black|out of Stock|Non-ABS|390 CC|26 KMPL|13.4L|#
59 592601|KTM|RC 390|2.80LAKHS EX|4.01LAKHS ONROAD|Racing Blue|InStock|ABS|390 CC|26 KMPL|13.4L|#
60 602201|KTM|RC 390|2.80LAKHS EX|4.01LAKHS ONROAD|Orange|InStock|ABS|390 CC|26 KMPL|13.4L|#
61 611301|KTM|Duke 250|2.35LAKHS EX|3.00LAKHS ONROAD|Silver|out of Stock|ABS|250 CC|30 KMPL|13.4L|#
62 621101|KTM|Duke 250|2.35LAKHS EX|3.00LAKHS ONROAD|Orange|InStock|ABS|250 CC|30 KMPL|13.4L|#
63 630401|KTM|Duke 200|2.01LAKHS EX|2.40LAKHS ONROAD|white|InStock|ABS|200 CC|35 KMPL|13.4L|#
64 639601|KTM|Duke 200|2.01LAKHS EX|2.40LAKHS ONROAD|Orange|out of Stock|ABS|200 CC|35 KMPL|13.4L|#
65 649401|HONDA|CBX200|1.45LAKHS EX|1.70LAKHS ONROAD|Red|InStock|ABS|200 CC|35 KMPL|12L|#
66 658201|HONDA|CBX200|1.45LAKHS EX|1.70LAKHS ONROAD|white|out of Stock|ABS|200 CC|35 KMPL|12L|#
67 667701|HONDA|Hornet 2.0|1.40LAKHS EX|1.75LAKHS ONROAD|Racing Blue|InStock|ABS|200 CC|40 KMPL|12L|#
68 677701|HONDA|Hornet 2.0|1.40LAKHS EX|1.75LAKHS ONROAD|Red Metallic|out of Stock|ABS|200 CC|40 KMPL|12L|#
69 688301|HONDA|Xblade|91,255 EX|1.30LAKHS ONROAD|Racing Red|InStock|ABS|162.5 CC|40 KMPL|12L|#
70 697701|HONDA|Xblade|91,255 EX|1.30LAKHS ONROAD|silver|out of Stock|ABS|162.5 CC|40 KMPL|12L|#
71 707201|HONDA|Activa og|75,897 EX|1.02LAKHS ONROAD|Red|out of Stock|UBS|110 CC|50 KMPL|4L|#
72 716401|HONDA|Activa og|75,897 EX|1.02LAKHS ONROAD|Blue|InStock|UBS|110 CC|50 KMPL|4L|#
73 725201|HONDA|Activa og|75,897 EX|1.02LAKHS ONROAD|Natte Black|InStock|UBS|110 CC|50 KMPL|4L|#
74 734701|HONDA|Dio|75,897 EX|98,856 ONROAD|Natte Black|InStock|UBS|110 CC|50 KMPL|4L|#
75 743301|HONDA|Dio|75,897 EX|98,856 ONROAD|Silver|out of Stock|UBS|110 CC|50 KMPL|4L|#

```

Fig 3.2.3 Confirmation of a deleted record

3.2.4 Add a Record (Admins Use Only)

- A record is Added to file according to new introduction of bikes into the market.

modelno	company	Model	X showroom	OnRoad	colour	Availability

Fig 3.2.4 Adding of a record

- A record is Added from the bikes.txt file and it is displayed.

```

File Edit Selection View Go Run Terminal Help
bikes.txt - FS Mini Project - Visual Studio Code
admin.html admin.txt bikes.txt
bikes.txt
1 001|Royal Enfield|Himalayan|2,80LAKHS ONROAD|Gravel Grey|InStock|ABS|411 CC|30 KMPL|15L|#
2 10501|Royal Enfield|Himalayan|2,80LAKHS ONROAD|Rock Red|InStock|ABS|411 CC|30 KMPL|15L|#
3 20801|Royal Enfield|Himalayan|2,80LAKHS EX|3,40LAKHS ONROAD|Lake Blue|out of Stock|NON-ABS|411 CC|30 KMPL|15L|#
4 32101|Royal Enfield|Himalayan|2,80LAKHS EX|3,15LAKHS ONROAD|Snow White|out of Stock|ABS|411 CC|30 KMPL|15L|#
5 43001|Royal Enfield|Meteor 350|2,0LAKHS EX|2,45LAKHS ONROAD|Fireball Red|InStock|ABS|350 CC|35 KMPL|15L|#
6 53701|Royal Enfield|Meteor 350|2,0LAKHS EX|2,60LAKHS ONROAD|Stellar Black|InStock|ABS|350 CC|35 KMPL|15L|#
7 64501|Royal Enfield|Meteor 350|2,0LAKHS ONROAD|Stellar Blue|out of Stock|ABS|350 CC|35 KMPL|15L|#
8 75501|Royal Enfield|Meteor 350|2LAKHS EX|2,45LAKHS ONROAD|Supernova Brown|out of Stock|ABS|350 CC|35 KMPL|15L|#
9 86801|Royal Enfield|Classic 350|2LAKHS EX|2,45LAKHS ONROAD|GumMetal Grey|InStock|ABS|350 CC|35 KMPL|13.5L|#
10 97701|Royal Enfield|Classic 350|2LAKHS EX|2,45LAKHS ONROAD|Strider Sand|InStock|ABS|350 CC|35 KMPL|13.5L|#
11 108801|Royal Enfield|Classic 350|2LAKHS EX|2,45LAKHS ONROAD|Red|out of Stock|Non-ABS|350 CC|35 KMPL|13.5L|#
12 119701|Royal Enfield|Continental GT650|3,40LAKHS EX|4,2LAKHS ONROAD|Clean|InStock|ABS|650 CC|20 KMPL|13.5L|#
13 131001|Royal Enfield|Continental GT650|3,40LAKHS EX|4,0LAKHS ONROAD|Rocker Red|InStock|ABS|650 CC|20 KMPL|13.5L|#
14 142501|Royal Enfield|Continental GT650|3,40LAKHS EX|3,99LAKHS ONROAD|British Green|out of Stock|ABS|650 CC|20 KMPL|13.7L|#
15 154901|Royal Enfield|Interceptor 650|3,30LAKHS EX|3,95LAKHS ONROAD|Baker Express|InStock|ABS|650 CC|20 KMPL|13.7L|#
16 166601|Royal Enfield|Interceptor 650|3,30LAKHS EX|3,97LAKHS ONROAD|Orange Crush|InStock|ABS|650 CC|20 KMPL|13.7L|#
17 178201|Royal Enfield|Interceptor 650|3,32LAKHS EX|3,96LAKHS ONROAD|Canyon Red|out of Stock|ABS|650 CC|20 KMPL|13.7L|#
18 190101|Royal Enfield|Interceptor 650|3,30LAKHS EX|3,96LAKHS ONROAD|Canyon Red|out of Stock|ABS|650 CC|20 KMPL|13.7L|#
19 202001|Royal Enfield|Bullet 500|1,98LAKHS EX|2,3LAKHS ONROAD|Forest Green|InStock|ABS|350 CC|35 KMPL|13.5L|#
20 213001|Royal Enfield|Bullet 500|1,98LAKHS EX|2,3LAKHS ONROAD|Royal Blue|out of Stock|Non-ABS|350 CC|35 KMPL|13.5L|#
21 224701|Yamaha|R15M|1,98LAKHS EX|2,48LAKHS ONROAD|Racing Blue|InStock|ABS|155 CC|40 KMPL|11L|#
22 234201|Yamaha|R15M|1,98LAKHS EX|2,60LAKHS ONROAD|Monster Edition|InStock|ABS|155 CC|40 KMPL|11L|#
23 244001|Yamaha|R15M|1,98LAKHS EX|2,48LAKHS ONROAD|Metallic Black|out of Stock|ABS|155 CC|40 KMPL|11L|#
24 254301|Yamaha|R15M|1,98LAKHS EX|2,48LAKHS ONROAD|Thunder Grey|out of Stock|ABS|155 CC|40 KMPL|11L|#
25 264401|Yamaha|MT-15V2|1,45LAKHS EX|2,10LAKHS ONROAD|Ice Fluo Vermillion|InStock|ABS|155 CC|45 KMPL|10L|#
26 275001|Yamaha|MT-15V2|1,45LAKHS EX|2,08LAKHS ONROAD|Metallic Black|InStock|ABS|155 CC|45 KMPL|10L|#
27 285101|Yamaha|MT-15V2|1,45LAKHS EX|2,08LAKHS ONROAD|Cyan Strom|out of Stock|ABS|155 CC|45 KMPL|10L|#
28 295301|Yamaha|MT-15V2|1,45LAKHS EX|2,08LAKHS ONROAD|Matte Blue|out of Stock|ABS|155 CC|45 KMPL|10L|#
29 305501|Yamaha|FZ V3|1,28LAKHS EX|1,65LAKHS ONROAD|Metallic Grey|InStock|ABS|149 CC|50 KMPL|13L|#
30 315301|Yamaha|FZ V3|1,22LAKHS EX|1,67LAKHS ONROAD|Metallic Red|out of Stock|ABS|149 CC|50 KMPL|13L|#
31 325501|Yamaha|Ray ZR|77,410 EX|96,863 ONROAD|Rally Red|InStock|UBS|125 CC|50 KMPL|4L|#
32 334301|Yamaha|Ray ZR|77,410 EX|96,863 ONROAD|Sparkle Green|out of Stock|UBS|125 CC|50 KMPL|4L|#

```

Fig 3.2.4 Addition of a record in text file.

- When the new record is added, the data which is added earlier will be displayed.

ID	Brand	Model	Price	Color	Stock Status	Action
743301	HONDA	Dio	75,897 EX	98,856 ONROAD	Silver	Out of Stock
751901	BAJAJ	Dominor 400	1.95LAKHS EX	2.95LAKHS ONROAD	Green	InStock
761401	BAJAJ	Dominor 400	1.95LAKHS EX	2.95LAKHS ONROAD	Black	InStock
770901	BAJAJ	Dominor 400	1.95LAKHS EX	2.95LAKHS ONROAD	Green	Out of Stock
781301	BAJAJ	Dominor 400	1.95LAKHS EX	2.95LAKHS ONROAD	Blue	Out of Stock
791601	BAJAJ	Dominor 250	1.71LAKHS EX	2.10LAKHS ONROAD	Red	InStock
801301	BAJAJ	Dominor 250	1.71LAKHS EX	2.10LAKHS ONROAD	Black	InStock
810801	BAJAJ	Dominor 250	1.71LAKHS EX	2.10LAKHS ONROAD	Yellow	InStock
820401	BAJAJ	Dominor 250	1.71LAKHS EX	2.10LAKHS ONROAD	Silver	InStock
830401	BAJAJ	RS 200	1.53LAKHS EX	2.00LAKHS ONROAD	Racing Blue	InStock
840001	BAJAJ	RS 200	1.53LAKHS EX	2.00LAKHS ONROAD	Graphite Black	InStock
849901	BAJAJ	RS 200	1.53LAKHS EX	2.00LAKHS ONROAD	Red	Out Of Stock
859201	BAJAJ	RS 200	1.53LAKHS EX	2.00LAKHS ONROAD	Silver	Out Of Stock
869201	BAJAJ	NS 200	1.40LAKHS EX	1.90LAKHS ONROAD	Matte Black	InStock
878801	BAJAJ	NS 200	1.40LAKHS EX	1.90LAKHS ONROAD	Mirage White	Out Of Stock

Fig 3.2.4 Modified record after Addition of record

CHAPTER 4

IMPLEMENTATION

4.1 About Python Flask

Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools.[4]

4.1.2 Features:

- **Development Server and Debugger** – Flask provides a run command to run the application with a development server. In development mode, this server provides an interactive debugger and will reload when code is changed.
- **Very Fast** – The built-in Flask web server is provided for development convenience. With it you can make your app accessible on your local machine without having to set up other services and make them play together nicely. However, it is only meant to be used by one person at a time, and is built this way.
- **Integrated support for Unit Testing** – Unit Testing is defined as a type of software testing where individual components of a software are tested. Unit Testing of software product is carried out during the development of an application. An individual component may be either an individual function or a procedure. Unit Testing is typically performed by the developer. It is a testing method using which every independent modules are tested to determine if there are any issue by the developer himself.
- **Uses Jinja templating** – Jinja is a web template engine for the Python programming language. Jinja is similar to the Django template engine but provides Python-like expressions while ensuring that the templates are evaluated in a sandbox. It is a text-based template language and thus can be used to generate any markup as well as sourcecode.[1]

4.2 Amazon Web Services:

Amazon Web Services (AWS) is a subsidiary of Amazon providing on-demand cloud computing platforms and APIs to individuals, companies, and governments, on a metered pay-as-you-go basis. These cloud computing web services provide a variety of basic abstract technical infrastructure and distributed computing building blocks and tools. One of these services is Amazon Elastic Compute Cloud (EC2), which allows users to have at their disposal a virtual cluster of computers, available all the time, through the Internet. AWS's virtual computers emulate most of the attributes of a real computer, including hardware central processing units (CPUs) and graphics processing units (GPUs) for processing; local/RAM memory; hard-disk/SSD storage; a choice of operating systems; networking; and pre-loaded application software such as web servers, databases, and customer relationship management (CRM).[1][3]

4.3 Amazon Elastic Beanstalk:

AWS Elastic Beanstalk is an easy-to-use service for deploying and scaling web applications and services developed with Java, .NET, PHP, Node.js, Python, Ruby, Go, and Docker on familiar servers such as Apache, Nginx, Passenger, and IIS.

You can simply upload your code and Elastic Beanstalk automatically handles the deployment, from capacity provisioning, load balancing, auto-scaling to application health monitoring. At the same time, you retain full control over the AWS resources powering your application and can access the underlying resources at any time.

The ‘Bike Showroom Management’ is built using **Python Flask** for backend support and **HTML, CSS** and **Bootstrap** for client’s side design and development. It is hosted on Website providing Users a hassle free experience.[2]

4.4 Pseudo Code

It’s simply an implementation of an algorithm in the form of annotations and informative text written in plain English. It has no syntax like any of the programming language and thus can’t be compiled or interpreted by the computer.

As Python Flask is a modern language developed under Python it is not required to open and close files. We can read and write directly all the other job is handled by the Python flask compiler itself.[5]

4.4.1 User Side:

Signup/Login Module Pseudo Code:

Step 1: Get input values of name, email-id ,phone number ,gender , password variables.

Step 2: Check whether the inputs are correct.

Step 3: Read users.txt and check whether registered or not If
registered

Set User Details cookie

Forward as customers module

Step 4: Read users.txt and check if user has already booked a bike If
true

Set user Details cookie

Forward to booked module

Step 3: Read users.txt and check whether user already exists using unique Id: If
exists

Set user Details cookie

Forward to user module

Step 4: Format the values by adding ‘|’ (delimiter) between fields and a ‘\n’ at end of line.

Step 5: If user not exists, add the values to a local variable that contains the read data.

Step 6: Write the data to users.txt file.

User Module:

Step 1: Display name of user from the cookie that was set.

Step 2: Allow to provide input to search their bikes.

Step 3: Get input from user i.e Bikes and Agencies.

Step 4: Fetch Bikes details from bikes.txt file.

Step 5: Split on newline character and then on delimiter and store to a variable.

Step 6: Loop though the variable array of centers to display Bikes matching the request.

Booked Module:

Step 1: Read booking.txt and fetch record matching users name, number ,email-d and password which is set via cookie.

Step 2: Display the name of user and Bike's details booked.

Users Booking Module:

Step 1: Read bookings.txt and display user's details and Bikes.

4.4.2 Admin Side:

Login Module Pseudo Code:

Step 1: Get input value password variable.

Step 2: Check whether input is correct.

Step 3: Admin cookie is set and then directed to admin module.

Customers Module:

Step 1: Allow to provide input to search different Bikes.

Step 2: Get input from admin

Step 3: Fetch Bike details and Customers details from
bikes.txt and customers.txt.

Step 4: Split on newline character and then on delimiter and store to a variable.

Step 5: Loop though the variable array of bikes to display bikes matching the request.

4.5 Testing

4.5.1 Unit Testing

Unit testing is software testing method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures, are tested to determine whether they are fit for use.[1]

- We are going to test the components of user login.
 - When the user logs in with the credentials like ,Name,Mobile number, password ,gender, email-id it should reflect in the users.txt file.

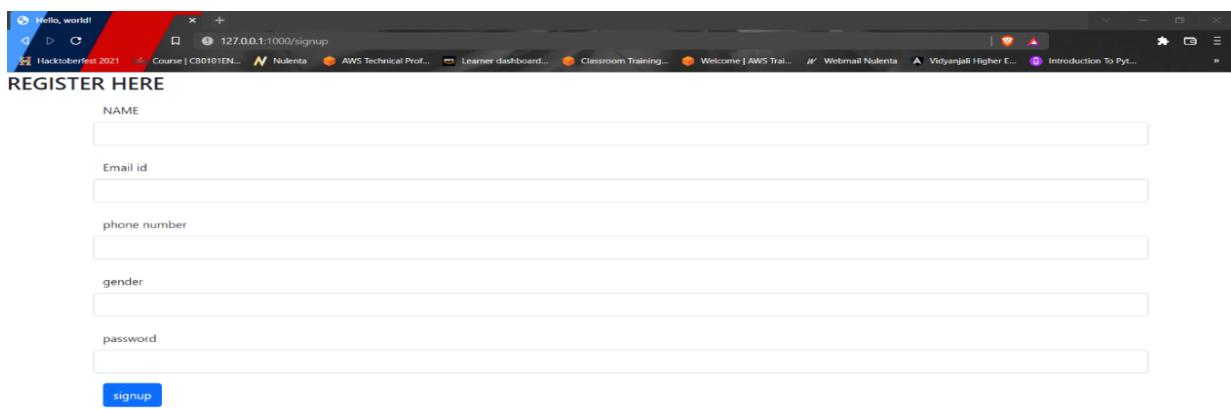


Fig 4.5.1 user login

The screenshot shows the VS Mini Project interface with several tabs open:

- File
- Edit
- Selection
- View
- Go
- Run
- Terminal
- Help

Open tabs include:

- EXPLORER
- FS MINI PROJECT
- base1.html
- base2.html
- book.html
- bookings.html
- booklist.html
- crud.html
- gallery.html
- home.html
- index.html
- index1.html
- index2.html
- index3.html
- index4.html
- index5.html
- login.html
- organ.html
- search.html
- signup.html
- userlist.html
- viewbooks.html
- admin.txt
- app.py
- bike1.txt
- bikes.txt
- bookings.txt
- comp.txt
- compress.py
- temp.txt
- users.txt

The code editor displays the content of the users.txt file:

```
1 | 1[tarun]ec8d0b1b40f4aefab2af3f43618f59cbe8a7af1fe1fba5f13ae391dbd1c3c0591//0/0/e8e1bdy762441b077051bb68d7f1b82c499454de5h
2 | 2[roshan]b900aee2c43fdbeed01fb8c8d02b1b5c5a1e4d2009009a0a595031ac1c1e9fbab1bd91cf02fa2rhe27adcf1fa80ad1df4e6a042a0bfb93d1558a0ed279a0fc76
3 | 3[harsh]b08c7d6d4aew20caabfb01175c03344e042a12090679a0129cfc5f6d4afelbf1f1d7f405f230866edc0fcfc0f082d16630f42d462b136d76a30308b0d0e10891cf656
4 | 4[pawan]3ddc6061a2bd3f4d178c9411cd04bf70805931c7f4adda198f78e5d52734bc21f4d818072499d9ebe3a315c8d4ad94c4dc31c55492fc0a1b41d2429846d5c
5 | 5[julias]fabf3bea31cfc0f51679c7dcf7c1662391d0f4eaa8641ac8f194e4a5d7ff1e85ed4f181315a4c0e08905d14c8f89a7f5e90765443c6002d6f16acff19e0048fb533
6 | 6[ambika]701f4718b14e588ba64d4097c8b904227b67647a282164d3dhc98f6443a0d06f798720b26b4d01237ada110d25c69eb859742zf719d4a45f16c039381c920e
7 | 7[ambika]64aa5f35a8971fac7a1857b7f320ad1c18347d0d2d31f598133745f07e7300a35b98100c2197632676690b1fae179bc160dce87a0e0276abbedfafb1ca
8 | 8[ashwin]2747114150ffcc31445a139c8eb70f397f8d8089c01d4adcc01a88956d1c908Fc1111e680533b17cc249d9f48568cde4c3668f8d88fe131ab0e688115d6
9 | 9[pranam]d705807f3d6062c13c49182b2607700193110a2088ed3de68169f9e867d19a3989a15223daeaeb36b05713c6f3107090c78a63ac55dd7033063b0637e
10 | 10[pananya]b7bdc98087d129ca154341d1a74379b9f757d61df56fabe62295968aaf73ce1ce0a7a786d22f51451d327320e2101986dd430b600163bd4013f84f7c8b2
11 |
```

Bottom status bar:

- PROBLEMS
- OUTPUT
- TERMINAL
- DEBUG CONSOLE

Message bar:

Install the latest PowerShell for new features and improvements! <https://aka.ms/powershell>

File path:

PS C:\Users\Lenovo\OneDrive\Documents\VS Mini Project>

System status:

24°C Partly cloudy

23:33 20.07.2023

Fig 4.5.1 Users.txt file

4.5.2 Integration Testing

Integration testing is a phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before validation testing. Integration testing follows two approaches known as ‘Top down’ approach and ‘Bottom Down’ approach. Admin and user have relation between them, here on the admin side i.e., in managing the details of bikes and customers we update, delete and add new bikes into the Bikes list. As soon we do any changes in the admin side, user side also gets updated. In this way we have developed small units and then tested by integrating them using integration testing.[1]

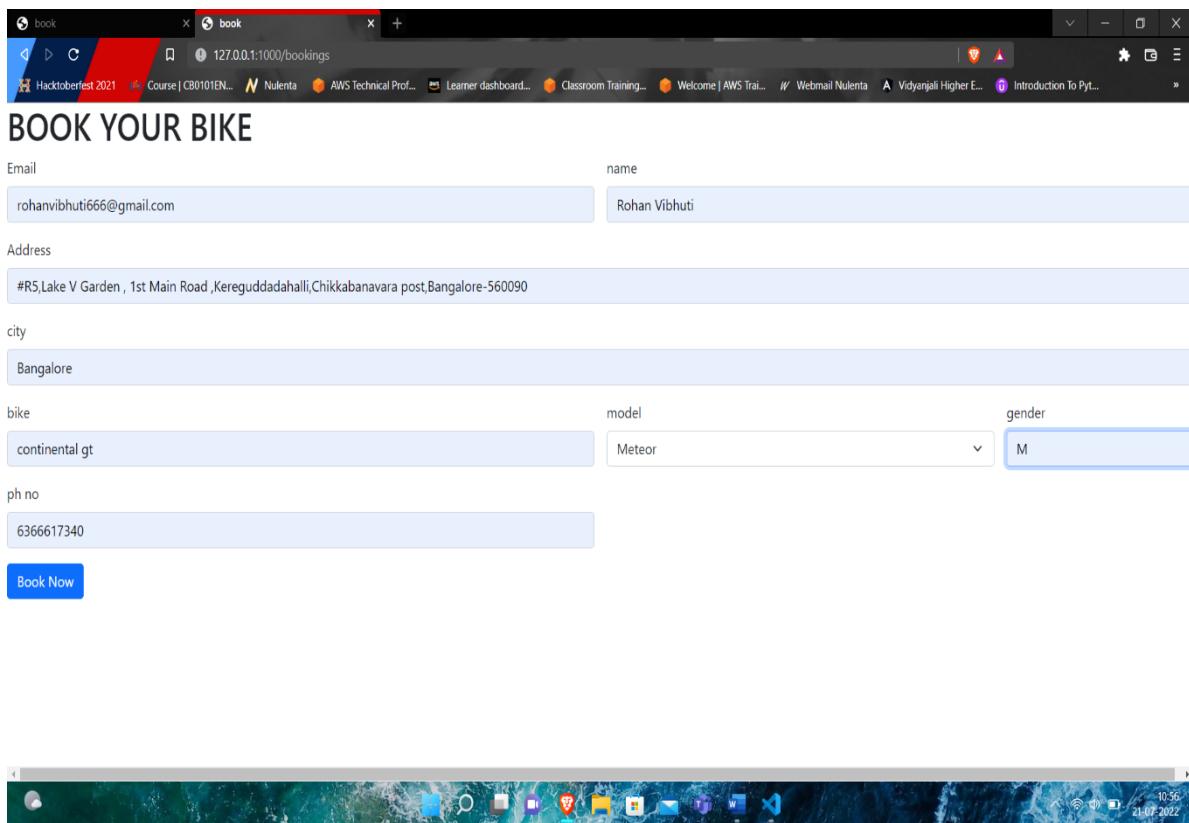
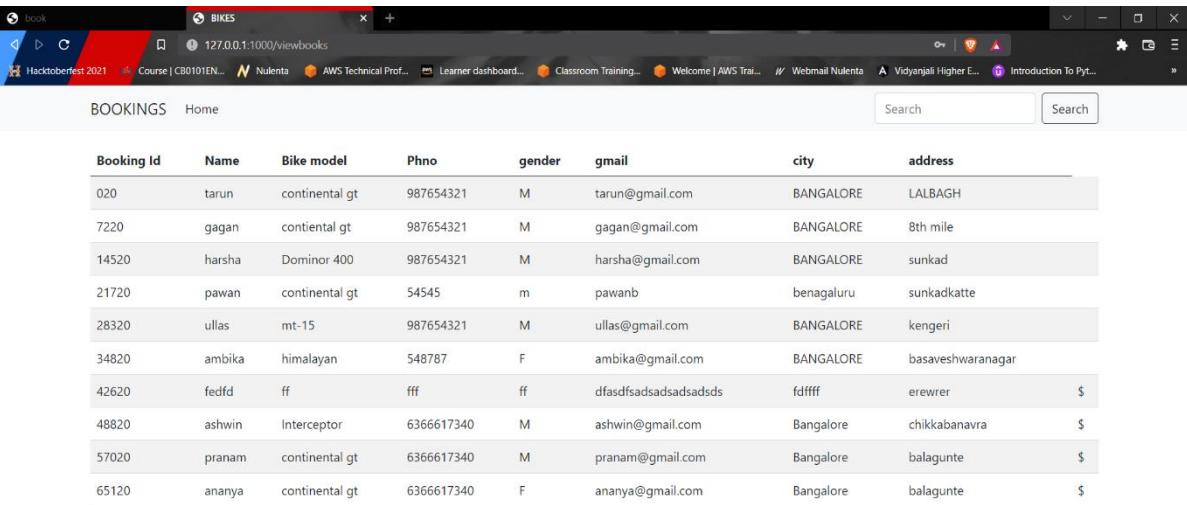


Fig 4.5.2 Booking module Page

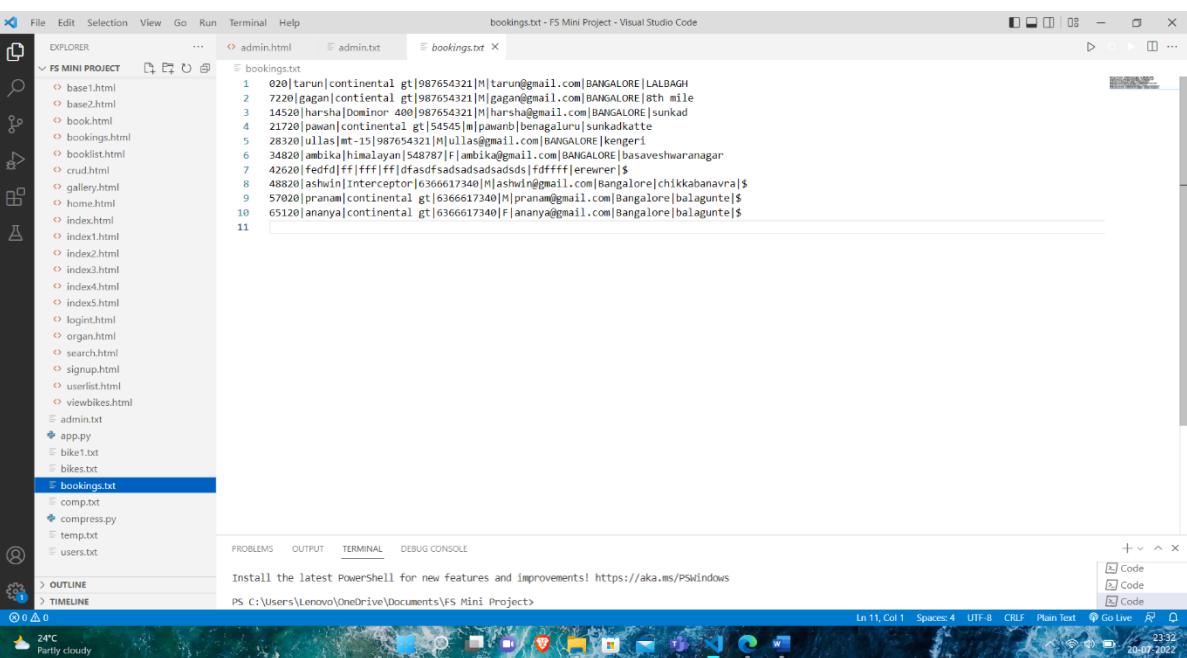


The screenshot shows a web browser window with the title 'BIKES' and the URL '127.0.0.1:1000/viewbooks'. The page displays a table of bike bookings with columns: Booking Id, Name, Bike model, Phno, gender, gmail, city, and address. The data is as follows:

Booking Id	Name	Bike model	Phno	gender	gmail	city	address
020	tarun	continental gt	987654321	M	tarun@gmail.com	BANGALORE	LALBAGH
7220	gagan	continental gt	987654321	M	gagan@gmail.com	BANGALORE	8th mile
14520	harsha	Dominor 400	987654321	M	harsha@gmail.com	BANGALORE	sunkad
21720	pawan	continental gt	54545	m	pawarb	benagaluru	sunkadkatte
28320	ullas	mt-15	987654321	M	ullas@gmail.com	BANGALORE	kengeri
34820	ambika	himalayan	548787	F	ambika@gmail.com	BANGALORE	basaveshwaranagar
42620	fedfd	ff	fff	ff	dfasdfsadsadsadsds	fdffff	erewrer
48820	ashwin	Interceptor	6366617340	M	ashwin@gmail.com	Bangalore	chikkabnavra
57020	pranam	continental gt	6366617340	M	pranam@gmail.com	Bangalore	balagunte
65120	ananya	continental gt	6366617340	F	ananya@gmail.com	Bangalore	balagunte



Fig 4.5.2 Admin Side Booking Log



```
bookings.txt - FS Mini Project - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER admin.html admin.txt bookings.txt
bookings.txt
1 020|tarun|continental gt|987654321|M|tarun@gmail.com|BANGALORE|LALBAGH
2 7220|gagan|continental gt|987654321|M|gagan@gmail.com|BANGALORE|8th mile
3 14520|harsha|Dominor 400|987654321|M|harsha@gmail.com|BANGALORE|sunkad
4 21720|pawan|continental gt|54545|m|pawarb|benagaluru|sunkadkatte
5 28320|ullas|mt-15|987654321|M|ullas@gmail.com|BANGALORE|kengeri
6 34820|ambika|himalayan|548787|F|ambika@gmail.com|BANGALORE|basaveshwaranagar
7 42620|fedfd|ff|fff|ff|dfasdfsadsadsadsds|fdffff|erewrer|
8 48820|ashwin|Interceptor|6366617340|M|ashwin@gmail.com|Bangalore|chikkabnavra|
9 57020|pranam|continental gt|6366617340|M|pranam@gmail.com|Bangalore|balagunte|
10 65120|ananya|continental gt|6366617340|F|ananya@gmail.com|Bangalore|balagunte|
```

4.5.3 System Testing

System testing is a level of software testing where complete and integrated software is tested. The process of testing an integrated system to verify that it needs specified requirements. The process of testing an integrated system to verify that it needs specified requirements. This program works well on the Windows operating system and as well as on the Linux operating system. The IDE used for coding is Visual Studio Code. It is recommended to run the project in an IDE that supports the latest version of the programming language.[1]

4.6 Discussion of Results

4.6.1 Admin Dashboard

Admin logs in with his password.

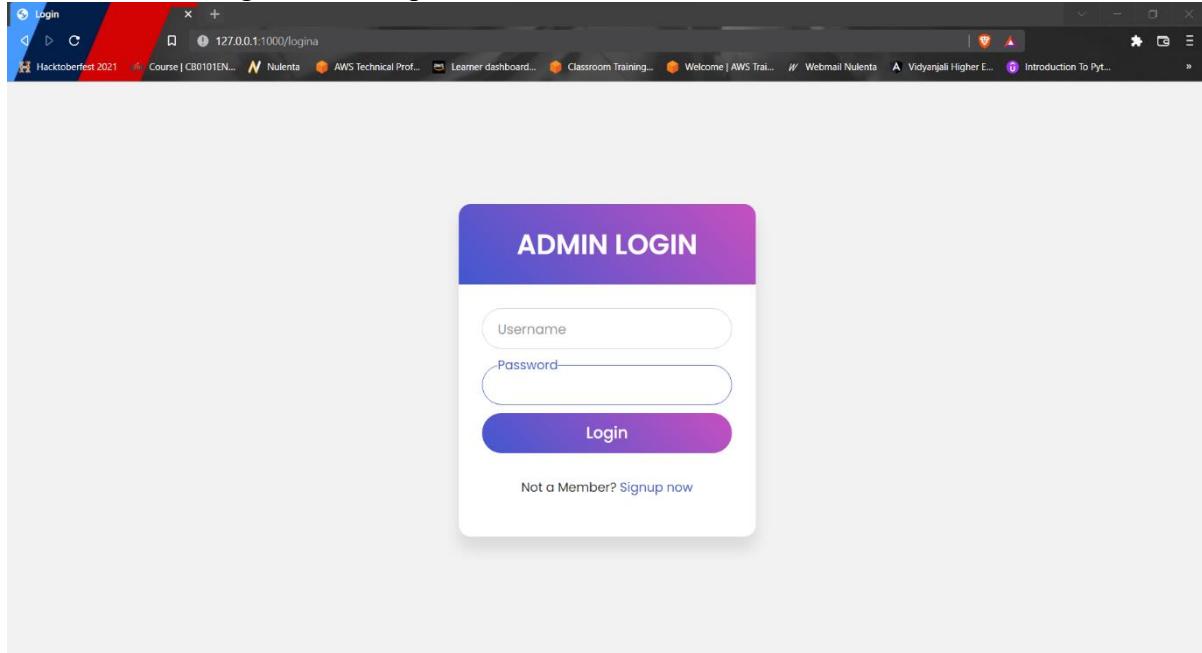


Fig 4.6.1 Admin login page

A screenshot of a web browser showing the 'Add New Bike Details' page. The page has a dark header with the text 'BIKES' and '127.0.0.1:1000/addbike'. Below the header, there is a form with various input fields: 'modelno', 'company', 'Model', 'X showroom', 'OnRoad', 'colour', and 'Availability'. Each field has a placeholder text. At the bottom of the form, there is a blue 'ADD' button and a table with columns: 'modelno', 'company', 'Model', 'X showroom', 'OnRoad', 'colour', and 'Availability'. The table currently has one row of data.

Fig 4.6.1 Add New Bike Details

4.6.2. Search Bikes

Admin can search Bikes by using this list.

ID	Brand	Model	Price	Stock Status	Color	Availability	Action
743301	HONDA	Dio	75,897 EX	98,856 ONROAD	Silver	Out of Stock	Delete
751901	BAJAJ	Dominor 400	1.95LAKHS EX	2.95LAKHS ONROAD	Green	InStock	Delete
761401	BAJAJ	Dominor 400	1.95LAKHS EX	2.95LAKHS ONROAD	Black	InStock	Delete
770901	BAJAJ	Dominor 400	1.95LAKHS EX	2.95LAKHS ONROAD	Green	Out of Stock	Delete
781301	BAJAJ	Dominor 400	1.95LAKHS EX	2.95LAKHS ONROAD	Blue	Out of Stock	Delete
791601	BAJAJ	Dominor 250	1.71LAKHS EX	2.10LAKHS ONROAD	Red	InStock	Delete
801301	BAJAJ	Dominor 250	1.71LAKHS EX	2.10LAKHS ONROAD	Black	InStock	Delete
810801	BAJAJ	Dominor 250	1.71LAKHS EX	2.10LAKHS ONROAD	Yellow	InStock	Delete
820401	BAJAJ	Dominor 250	1.71LAKHS EX	2.10LAKHS ONROAD	Silver	InStock	Delete
830401	BAJAJ	RS 200	1.53LAKHS EX	2.00LAKHS ONROAD	Racing Blue	InStock	Delete
840001	BAJAJ	RS 200	1.53LAKHS EX	2.00LAKHS ONROAD	Graphite Black	InStock	Delete
849901	BAJAJ	RS 200	1.53LAKHS EX	2.00LAKHS ONROAD	Red	Out Of Stock	Delete
859201	BAJAJ	RS 200	1.53LAKHS EX	2.00LAKHS ONROAD	Silver	Out Of Stock	Delete
869201	BAJAJ	NS 200	1.40LAKHS EX	1.90LAKHS ONROAD	Matte Black	InStock	Delete
878801	BAJAJ	NS 200	1.40LAKHS EX	1.90LAKHS ONROAD	Mirage White	Out Of Stock	Delete

Fig 4.6.2 Searching

```

File Edit Selection View Go Run Terminal Help
admin.html admin.txt bikes.txt
bikes.txt - FS Mini Project - Visual Studio Code
File Edit Selection View Go Run Terminal Help
admin.html admin.txt bikes.txt
bikes.txt
44 453401|TVS|Apache RTR160|1.30LAKHS EX|1.55LAKHS ONROAD|white|Out of Stock|Non-ABS|160 CC|40 KMPL|12L#
45 463801|TVS|Jupiter|78,045 EX|98,948 ONROAD|Royal Wine|InStock|ABS|110 CC|45 KMPL|4L#
46 472501|TVS|Jupiter|78,045 EX|98,948 ONROAD|blue|InStock|ABS|110 CC|45 KMPL|4L#
47 480601|TVS|Jupiter|78,045 EX|98,948 ONROAD|Peral White|Out of Stock|ABS|110 CC|40 KMPL|4L#
48 488901|TVS|Jupiter|78,045 EX|98,948 ONROAD|Grey|Out of Stock|ABS|110 CC|40 KMPL|4L#
49 498501|TVS|Starcity Plus|55,069 EX|93,567 ONROAD|Grey silver|InStock|ABS|110 CC|40 KMPL|10L#
50 508801|TVS|Starcity Plus|55,069 EX|93,567 ONROAD|Red|Out of Stock|ABS|110 CC|40 KMPL|4L#
51 517201|TVS|Wego|55,069 EX|80,789 ONROAD|White|InStock|ABS|110 CC|45 KMPL|4L#
52 525101|TVS|Wego|55,069 EX|80,789 ONROAD|Blue|Out of Stock|ABS|110 CC|45 KMPL|4L#
53 533401|KTM|Duke 390|2,70LAKHS EX|3,90LAKHS ONROAD|white|InStock|ABS|390 CC|28 KMPL|13.4L#
54 542601|KTM|Duke 390|2,70LAKHS EX|3,90LAKHS ONROAD|Silver|InStock|ABS|390 CC|28 KMPL|13.4L#
55 551901|KTM|Duke 390|2,70LAKHS EX|3,90LAKHS ONROAD|Orange|Out of Stock|ABS|390 CC|28 KMPL|13.4L#
56 561701|KTM|Duke 390|2,70LAKHS EX|3,90LAKHS ONROAD|Silver|Out of Stock|ABS|390 CC|28 KMPL|13.4L#
57 572401|KTM|RC 390|2,00LAKHS EX|4,01LAKHS ONROAD|Metallic Grey|Out of Stock|ABS|390 CC|26 KMPL|13.4L#
58 582701|KTM|RC 390|2,00LAKHS EX|4,01LAKHS ONROAD|Black|Out of Stock|ABS|390 CC|26 KMPL|13.4L#
59 592601|KTM|RC 390|2,00LAKHS EX|4,01LAKHS ONROAD|Racing Blue|InStock|ABS|390 CC|26 KMPL|13.4L#
60 602201|KTM|RC 390|2,00LAKHS EX|4,01LAKHS ONROAD|Orange|InStock|ABS|390 CC|26 KMPL|13.4L#
61 611301|KTM|Duke 250|2,35LAKHS EX|3,00LAKHS ONROAD|Silver|Out of Stock|ABS|250 CC|30 KMPL|13.4L#
62 621101|KTM|Duke 250|2,35LAKHS EX|3,00LAKHS ONROAD|Orange|InStock|ABS|250 CC|30 KMPL|13.4L#
63 630401|KTM|Duke 200|2,01LAKHS EX|2,40LAKHS ONROAD|White|InStock|ABS|200 CC|30 KMPL|13.4L#
64 639601|KTM|Duke 200|2,01LAKHS EX|2,40LAKHS ONROAD|Orange|Out of Stock|ABS|200 CC|35 KMPL|13.4L#
65 649401|HONDA|CBX200|1,45LAKHS EX|1,70LAKHS ONROAD|Red|InStock|ABS|200 CC|30 KMPL|12L#
66 658201|HONDA|CBX200|1,45LAKHS EX|1,70LAKHS ONROAD|White|Out of Stock|ABS|200 CC|35 KMPL|12L#
67 667701|HONDA|Hornet 2.0|1,40LAKHS EX|1,75LAKHS ONROAD|Racing Blue|InStock|ABS|200 CC|40 KMPL|12L#
68 677701|HONDA|Hornet 2.0|1,40LAKHS EX|1,75LAKHS ONROAD|Red Metallic|Out of Stock|ABS|200 CC|40 KMPL|12L#
69 688301|HONDA|Xblade|91,255 EX|1,30LAKHS ONROAD|Racing Red|InStock|ABS|162.5 CC|40 KMPL|12L#
70 697701|HONDA|Xblade|91,255 EX|1,30LAKHS ONROAD|Silver|Out of Stock|ABS|162.5 CC|40 KMPL|12L#
71 707201|HONDA|Activia eg|75,897 EX|1,02LAKHS ONROAD|Red|Out of Stock|ABS|110 CC|50 KMPL|4L#
72 716401|HONDA|Activia eg|75,897 EX|1,02LAKHS ONROAD|Blue|InStock|ABS|110 CC|50 KMPL|4L#
73 725201|HONDA|Activia eg|75,897 EX|1,02LAKHS ONROAD|Matte Black|InStock|ABS|110 CC|50 KMPL|4L#
74 734701|HONDA|Dio|75,897 EX|98,856 ONROAD|Matte Black|InStock|ABS|110 CC|50 KMPL|4L#
75 743301|HONDA|Dio|75,897 EX|98,856 ONROAD|Silver|Out of Stock|ABS|110 CC|50 KMPL|4L#

```

Fig 4.6.2 Admin can also search bikes details from bikes.txt file

4.6.13 Delete Bikes

Admin can delete the bike details.

modelno	company	Model	X showroom	OnRoad	colour	Availability	
001	Royal Enfield	Himalayan	2.80LAKHS EX	3.20LAKHS ONROAD	Gravel Grey	InStock	<button>Delete</button>
10501	Royal Enfield	Himalayan	2.80LAKHS EX	3.40LAKHS ONROAD	Rock Red	InStock	<button>Delete</button>
20801	Royal Enfield	Himalayan	2.80LAKHS EX	3.15LAKHS ONROAD	Lake Blue	Out Of Stock	<button>Delete</button>
32101	Royal Enfield	Himalayan	2.80LAKHS EX	3.2LAKHS ONROAD	Snow White	Out of Stock	<button>Delete</button>
43001	Royal Enfield	Meteor 350	2.0LAKHS EX	2.45LAKHS ONROAD	Fireball Red	InStock	<button>Delete</button>
53701	Royal Enfield	Meteor 350	2.0LAKHS EX	2.60LAKHS ONROAD	Stellar Black	InStock	<button>Delete</button>
64501	Royal Enfield	Meteor 350	2LAKHS EX	2.60LAKHS ONROAD	Stellar Blue	Out of Stock	<button>Delete</button>
75501	Royal Enfield	Meteor 350	2LAKHS EX	2.45LAKHS ONROAD	Supernova Brown	Out of Stock	<button>Delete</button>
86801	Royal Enfield	Classic 350	2LAKHS EX	2.45LAKHS ONROAD	GunMetal Grey	InStock	<button>Delete</button>
97701	Royal Enfield	Classic 350	2LAKHS EX	2.45LAKHS ONROAD	Stromrider Sand	InStock	<button>Delete</button>
108801	Royal Enfield	Classic 350	2LAKHS EX	2.45LAKHS ONROAD	Red	Out of Stock	<button>Delete</button>
119701	Royal Enfield	Continental GT650	3.40LAKHS EX	4.2LAKHS ONROAD	MR Clean	InStock	<button>Delete</button>
131001	Royal Enfield	Continental GT650	3.40LAKHS EX	4.0LAKHS ONROAD	Rocker Red	InStock	<button>Delete</button>
142501	Royal Enfield	Continental GT650	3.40LAKHS EX	3.99LAKHS ONROAD	British Green	Out Of Stock	<button>Delete</button>

Fig 4.6.3 Before deleting

533401	KTM	Duke 390	2.70LAKHS EX	3.90LAKHS ONROAD	White	InStock	<button>Delete</button>
542601	KTM	Duke 390	2.70LAKHS EX	3.90LAKHS ONROAD	Silver	InStock	<button>Delete</button>
551901	KTM	Duke 390	2.70LAKHS EX	3.90LAKHS ONROAD	Orange	Out Of Stock	<button>Delete</button>
561701	KTM	Duke 390	2.70LAKHS EX	3.90LAKHS ONROAD	Metallic Silver	Out of Stock	<button>Delete</button>
572401	KTM	RC 390	2.80LAKHS EX	4.01LAKHS ONROAD	Metallic Grey	Out of Stock	<button>Delete</button>
582701	KTM	RC 390	2.80LAKHS EX	4.01LAKHS ONROAD	Black	Out of Stock	<button>Delete</button>
592601	KTM	RC 390	2.80LAKHS EX	4.01LAKHS ONROAD	Racing Blue	InStock	<button>Delete</button>
602201	KTM	RC 390	2.80LAKHS EX	4.01LAKHS ONROAD	Orange	InStock	<button>Delete</button>
611301	KTM	Duke 250	2.35LAKHS EX	3.00LAKHS ONROAD	Silver	Out of Stock	<button>Delete</button>
621101	KTM	Duke 250	2.35LAKHS EX	3.00LAKHS ONROAD	Orange	InStock	<button>Delete</button>
630401	KTM	Duke 200	2.01LAKHS EX	2.40LAKHS ONROAD	White	InStock	<button>Delete</button>
639601	KTM	Duke 200	2.01LAKHS EX	2.40LAKHS ONROAD	Orange	Out of Stock	<button>Delete</button>
649401	HONDA	CBX200	1.45LAKHS EX	1.70LAKHS ONROAD	Red	InStock	<button>Delete</button>
658201	HONDA	CBX200	1.45LAKHS EX	1.70LAKHS ONROAD	White	Out Of Stock	<button>Delete</button>
667701	HONDA	Hornet 2.0	1.40LAKHS EX	1.75LAKHS ONROAD	Racing Blue	InStock	<button>Delete</button>

Fig 4.6.3 After deleting

4.6.4 Modify /Add Bikes

Admin can modify/add the particular new bikes.

Fig 4.6.4 Before modification

```

1 001|Royal Enfield|Himalayan|2.80LAHKS ONROAD|Gravel Grey|InStock|ABS|411 CC|30 KMPL|15L|#  

2 10501|Royal Enfield|Himalayan|2.80LAHKS EX|3.40LAHKS ONROAD|Rock Red|InStock|ABS|411 CC|30 KMPL|15L|#  

3 20801|Royal Enfield|Himalayan|2.80LAHKS EX|3.15LAHKS ONROAD|Lake Blue|Out of Stock|NON-ABS|411 CC|30 KMPL|15L|#  

4 32101|Royal Enfield|Meteor|2.80LAHKS EX|3.2LAHKS ONROAD|Snow White|Out of Stock|ABS|411 CC|30 KMPL|15L|#  

5 43001|Royal Enfield|Meteor|2.0LAHKS EX|2.45LAHKS ONROAD|Fireball Red|InStock|ABS|350 CC|35 KMPL|15L|#  

6 53701|Royal Enfield|Meteor 350|2.0LAHKS EX|2.60LAHKS ONROAD|Stellar Black|InStock|ABS|350 CC|35 KMPL|15L|#  

7 64501|Royal Enfield|Meteor 350|2LAHKS EX|2.60LAHKS ONROAD|Stellar Blue|Out of Stock|ABS|350 CC|35 KMPL|15L|#  

8 75501|Royal Enfield|Meteor 350|2LAHKS EX|2.45LAHKS ONROAD|Supernova Brown|Out of Stock|ABS|350 CC|35 KMPL|15L|#  

9 86801|Royal Enfield|Metris|2.0LAHKS EX|2.45LAHKS ONROAD|GumMetal Grey|InStock|ABS|350 CC|35 KMPL|15L|#  

10 97701|Royal Enfield|Metris|2.0LAHKS EX|2.45LAHKS ONROAD|Stromrider Sand|InStock|ABS|350 CC|35 KMPL|15L|#  

11 108801|Royal Enfield|Classic 350|3.0LAHKS EX|2.45LAHKS ONROAD|Red|Out of Stock|NON-ABS|350 CC|35 KMPL|15L|#  

12 119701|Royal Enfield|Continental GT650|3.40LAHKS EX|4.2LAHKS ONROAD|Clean|InStock|ABS|650 CC|20 KMPL|13.5L|#  

13 131001|Royal Enfield|Continental GT650|3.40LAHKS EX|4.0LAHKS ONROAD|Rocker Red|InStock|ABS|650 CC|20 KMPL|13.5L|#  

14 142501|Royal Enfield|Continental GT650|3.40LAHKS EX|3.99LAHKS ONROAD|British Green|Out of Stock|ABS|650 CC|20 KMPL|13.7L|#  

15 154901|Royal Enfield|Interceptor 650|3.30LAHKS EX|3.95LAHKS ONROAD|Baker Express|InStock|ABS|650 CC|20 KMPL|13.7L|#  

16 166601|Royal Enfield|Interceptor 650|3.33LAHKS EX|3.97LAHKS ONROAD|Orange Crush|InStock|ABS|650 CC|20 KMPL|13.7L|#  

17 178201|Royal Enfield|Interceptor 650|3.32LAHKS EX|3.96LAHKS ONROAD|Canyon Red|Out of Stock|ABS|650 CC|20 KMPL|13.7L|#  

18 190101|Royal Enfield|Interceptor 650|3.32LAHKS EX|3.96LAHKS ONROAD|Canyon Red|Out of Stock|ABS|650 CC|20 KMPL|13.7L|#  

19 202001|Royal Enfield|Bullet 350|1.98LAHKS EX|2.3LAHKS ONROAD|Forest Green|InStock|ABS|350 CC|35 KMPL|13.5L|#  

20 213001|Royal Enfield|Bullet 350|1.98LAHKS EX|2.3LAHKS ONROAD|Royal Blue|Out of Stock|NON-ABS|350 CC|35 KMPL|13.5L|#  

21 224701|Yamaha|R15M|1.98LAHKS EX|2.48LAHKS ONROAD|Racing Blue|InStock|ABS|155 CC|40 KMPL|11L|#  

22 234201|Yamaha|R15M|1.98LAHKS EX|2.60LAHKS ONROAD|Monster Edition|InStock|ABS|155 CC|40 KMPL|11L|#  

23 244001|Yamaha|R15M|1.98LAHKS EX|2.48LAHKS ONROAD|Metallic Black|Out of Stock|ABS|155 CC|40 KMPL|11L|#  

24 254301|Yamaha|R15M|1.98LAHKS EX|2.48LAHKS ONROAD|Thunder Grey|Out of Stock|ABS|155 CC|40 KMPL|11L|#  

25 264401|Yamaha|MT-15V2|1.45LAHKS EX|2.10LAHKS ONROAD|Ice Fluo Vermillion|InStock|ABS|155 CC|45 KMPL|10L|#  

26 275901|Yamaha|MT-15V2|1.45LAHKS EX|2.08LAHKS ONROAD|Metallic Black|InStock|ABS|155 CC|45 KMPL|10L|#  

27 285101|Yamaha|MT-15V2|1.45LAHKS EX|2.08LAHKS ONROAD|Cyan Strom|Out of Stock|ABS|155 CC|45 KMPL|10L|#  

28 295301|Yamaha|MT-15V2|1.45LAHKS EX|2.08LAHKS ONROAD|Matte Blue|Out of Stock|ABS|155 CC|45 KMPL|10L|#  

29 305501|Yamaha|FZ V3|1.20LAHKS EX|1.65LAHKS ONROAD|Metallic Grey|InStock|ABS|149 CC|50 KMPL|13L|#  

30 315301|Yamaha|FZ V3|1.22LAHKS EX|1.67LAHKS ONROAD|Metallic Red|Out of Stock|ABS|149 CC|50 KMPL|13L|#  

31 325501|Yamaha|Ray ZR|77,410 EX|96,863 ONROAD|Rally Red|InStock|UBS|125 CC|50 KMPL|4L|#  

32 334301|Yamaha|Ray ZR|77,410 EX|96,863 ONROAD|Sparkle Green|Out of Stock|UBS|125 CC|50 KMPL|4L|#

```

Fig 4.6.4 After modification can be viewed in bikes.txt file

4.6.5 Add new Bikes

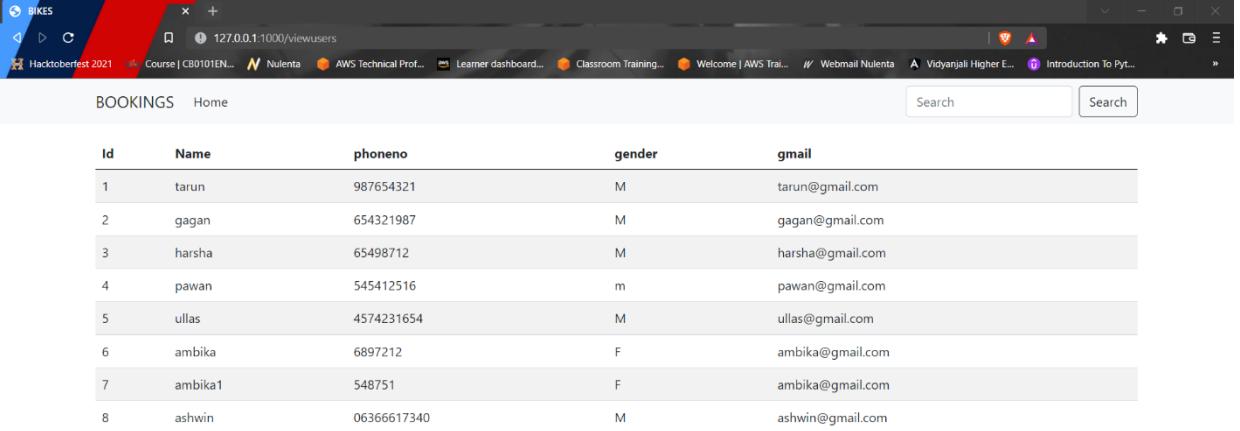
modelno	company	Model	X showroom	OnRoad	colour	Availability	
001	Royal Enfield	Himalayan	2.80LAKHS EX	3.20LAKHS ONROAD	Gravel Grey	InStock	<button>Delete</button>
10501	Royal Enfield	Himalayan	2.80LAKHS EX	3.40LAKHS ONROAD	Rock Red	InStock	<button>Delete</button>
20801	Royal Enfield	Himalayan	2.80LAKHS EX	3.15LAKHS ONROAD	Lake Blue	Out Of Stock	<button>Delete</button>
32101	Royal Enfield	Himalayan	2.80LAKHS EX	3.2LAKHS ONROAD	Snow White	Out of Stock	<button>Delete</button>
43001	Royal Enfield	Meteor 350	2.0LAKHS EX	2.45LAKHS ONROAD	Fireball Red	InStock	<button>Delete</button>
53701	Royal Enfield	Meteor 350	2.0LAKHS EX	2.60LAKHS ONROAD	Stellar Black	InStock	<button>Delete</button>
64501	Royal Enfield	Meteor 350	2LAKHS EX	2.60LAKHS ONROAD	Stellar Blue	Out of Stock	<button>Delete</button>

Fig 4.6.5 add new bikes

4.6.6 New added Bikes can be viewed by admin

modelno	company	Model	X showroom	OnRoad	colour	Availability	
743301	HONDA	Dio	75,897 EX	98,856 ONROAD	Silver	Out of Stock	<button>Delete</button>
751901	BAJAJ	Dominor 400	1.95LAKHS EX	2.95LAKHS ONROAD	Green	InStock	<button>Delete</button>
761401	BAJAJ	Dominor 400	1.95LAKHS EX	2.95LAKHS ONROAD	Black	InStock	<button>Delete</button>
770901	BAJAJ	Dominor 400	1.95LAKHS EX	2.95LAKHS ONROAD	Green	Out of Stock	<button>Delete</button>
781301	BAJAJ	Dominor 400	1.95LAKHS EX	2.95LAKHS ONROAD	Blue	Out of Stock	<button>Delete</button>
791601	BAJAJ	Dominor 250	1.71LAKHS EX	2.10LAKHS ONROAD	Red	InStock	<button>Delete</button>
801301	BAJAJ	Dominor 250	1.71LAKHS EX	2.10LAKHS ONROAD	Black	InStock	<button>Delete</button>
810801	BAJAJ	Dominor 250	1.71LAKHS EX	2.10LAKHS ONROAD	Yellow	InStock	<button>Delete</button>
820401	BAJAJ	Dominor 250	1.71LAKHS EX	2.10LAKHS ONROAD	Silver	InStock	<button>Delete</button>
830401	BAJAJ	RS 200	1.53LAKHS EX	2.00LAKHS ONROAD	Racing Blue	InStock	<button>Delete</button>
840001	BAJAJ	RS 200	1.53LAKHS EX	2.00LAKHS ONROAD	Graphite Black	InStock	<button>Delete</button>
849901	BAJAJ	RS 200	1.53LAKHS EX	2.00LAKHS ONROAD	Red	Out Of Stock	<button>Delete</button>
859201	BAJAJ	RS 200	1.53LAKHS EX	2.00LAKHS ONROAD	Silver	Out Of Stock	<button>Delete</button>
869201	BAJAJ	NS 200	1.40LAKHS EX	1.90LAKHS ONROAD	Matte Black	InStock	<button>Delete</button>
878801	BAJAJ	NS 200	1.40LAKHS EX	1.90LAKHS ONROAD	Mirage White	Out Of Stock	<button>Delete</button>

Fig 4.6.6 New added bikes



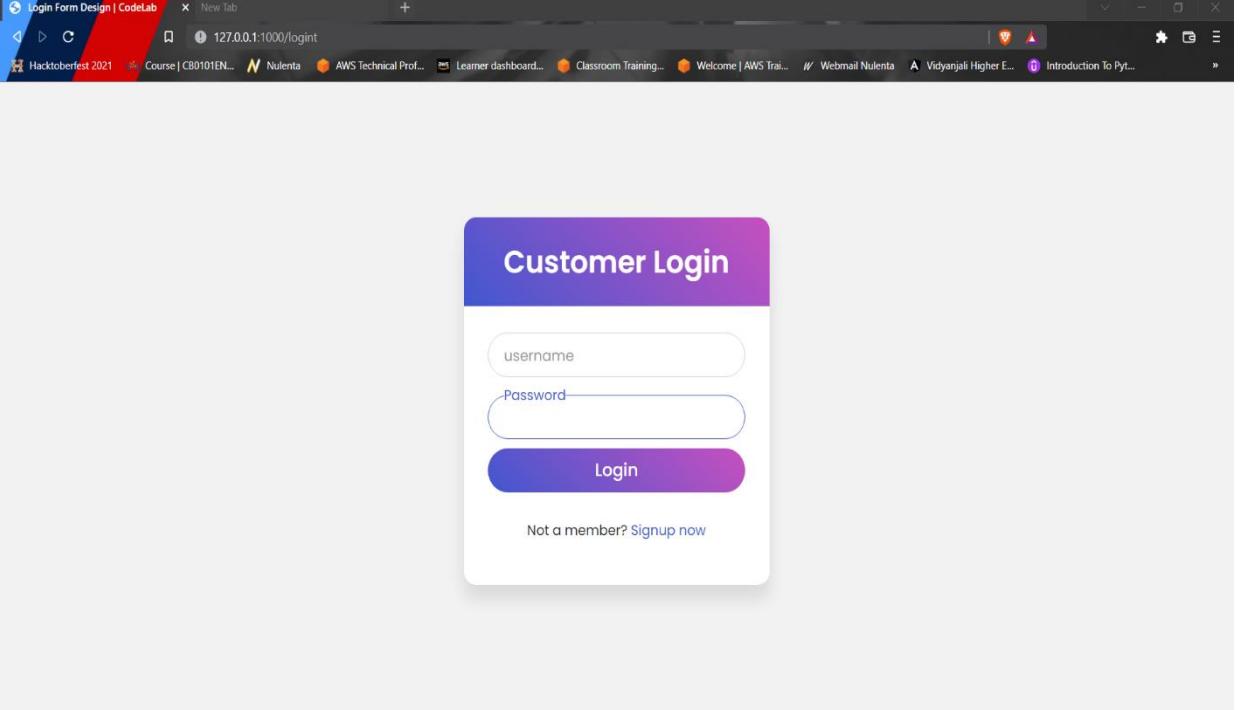
A screenshot of a web browser window titled "BOOKINGS Home". The URL in the address bar is "127.0.0.1:1000/viewusers". The page displays a table with columns: Id, Name, phoneno, gender, and gmail. The data is as follows:

Id	Name	phoneno	gender	gmail
1	tarun	987654321	M	tarun@gmail.com
2	gagan	654321987	M	gagan@gmail.com
3	harsha	65498712	M	harsha@gmail.com
4	pawan	545412516	m	pawan@gmail.com
5	ullas	4574231654	M	ullas@gmail.com
6	ambika	6897212	F	ambika@gmail.com
7	ambika1	548751	F	ambika@gmail.com
8	ashwin	06366617340	M	ashwin@gmail.com

Fig 4.6.6 Registered Users for booking a bike.

4.6.7 User Dashboard

User logs in with all his credentials.



A screenshot of a web browser window titled "Customer Login". The URL in the address bar is "127.0.0.1:1000/login". The page features a purple header with the text "Customer Login". Below it is a form with two input fields: "username" and "Password", and a "Login" button. At the bottom of the form, there is a link "Not a member? Signup now".

Fig 4.6.7 User login page

4.6.8 Choose the bike model

BOOK YOUR BIKE

Email

name

Address 1234 Main St

city

bike

model Choose...

gender

ph no

Book Now

Fig 4.6.8 Booking a Bike

4.6.9 Bike Booked details

BOOKINGS Home

Booking Id	Name	Bike model	Phno	gender	gmail	city	address
020	tarun	continental gt	987654321	M	tarun@gmail.com	BANGALORE	LALBAGH
7220	gagan	continental gt	987654321	M	gagan@gmail.com	BANGALORE	8th mile
14520	harsha	Dominor 400	987654321	M	harsha@gmail.com	BANGALORE	sunkad
21720	pawan	continental gt	54545	m	pawab	benagaluru	sunkadkatte
28320	ullas	mt-15	987654321	M	ullas@gmail.com	BANGALORE	kengeri
34820	ambika	himalayan	548787	F	ambika@gmail.com	BANGALORE	basaveshwaranagar
42620	fedfd	ff	fff	ff	dfasdfsadsadsadsds	fdffff	erewrer
48820	ashwin	Interceptor	6366617340	M	ashwin@gmail.com	Bangalore	chikkabanavra
57020	pranam	continental gt	6366617340	M	pranam@gmail.com	Bangalore	balagunte

Fig 4.6.9 Bike Booking details

4.6.10 Snapshots

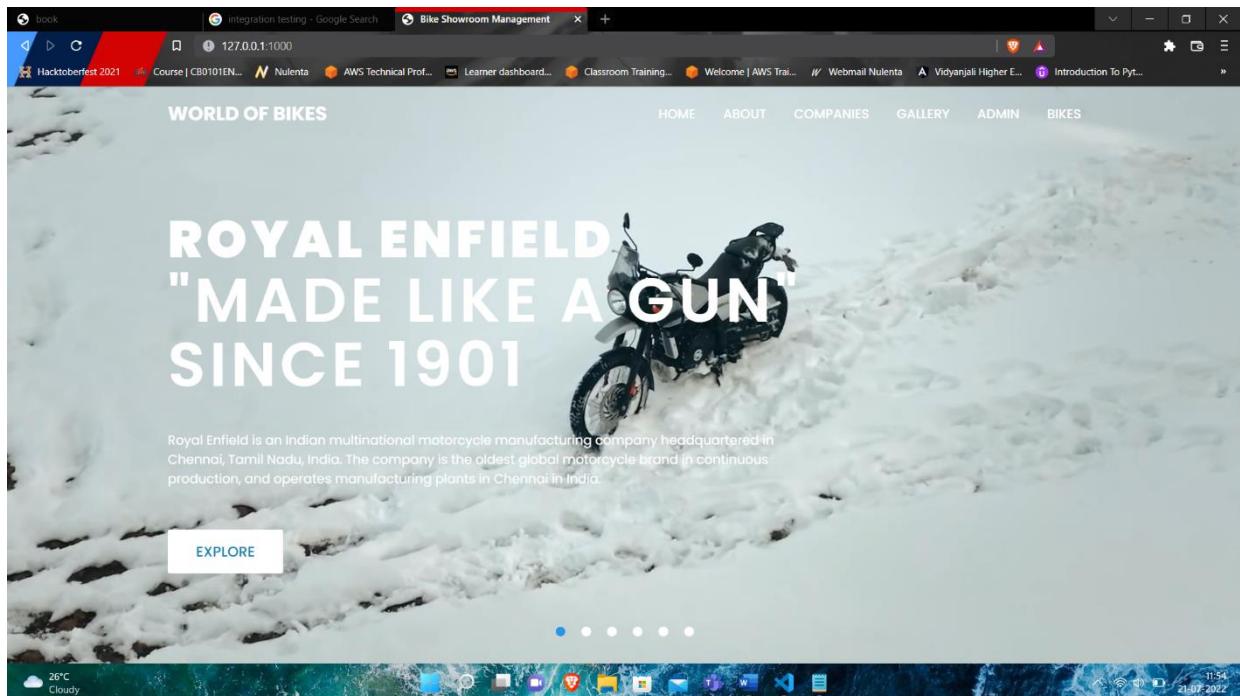


Fig 4.6.10 Home Page

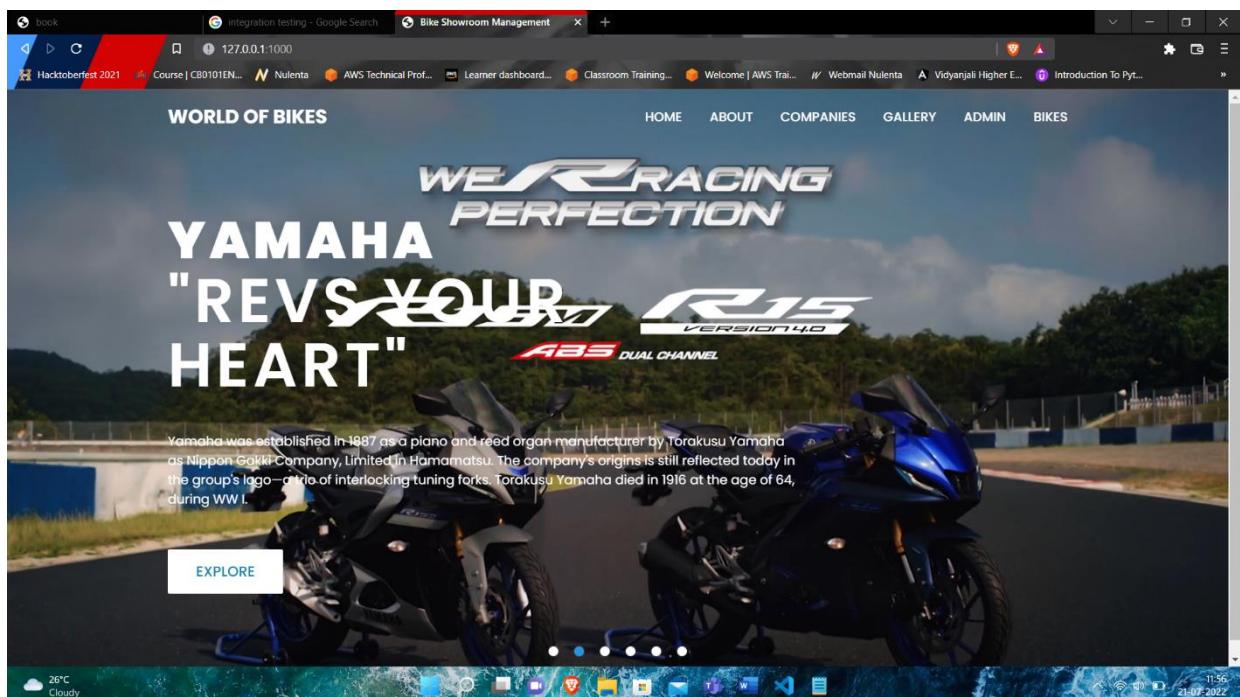


Fig 4.6.10 Home Page

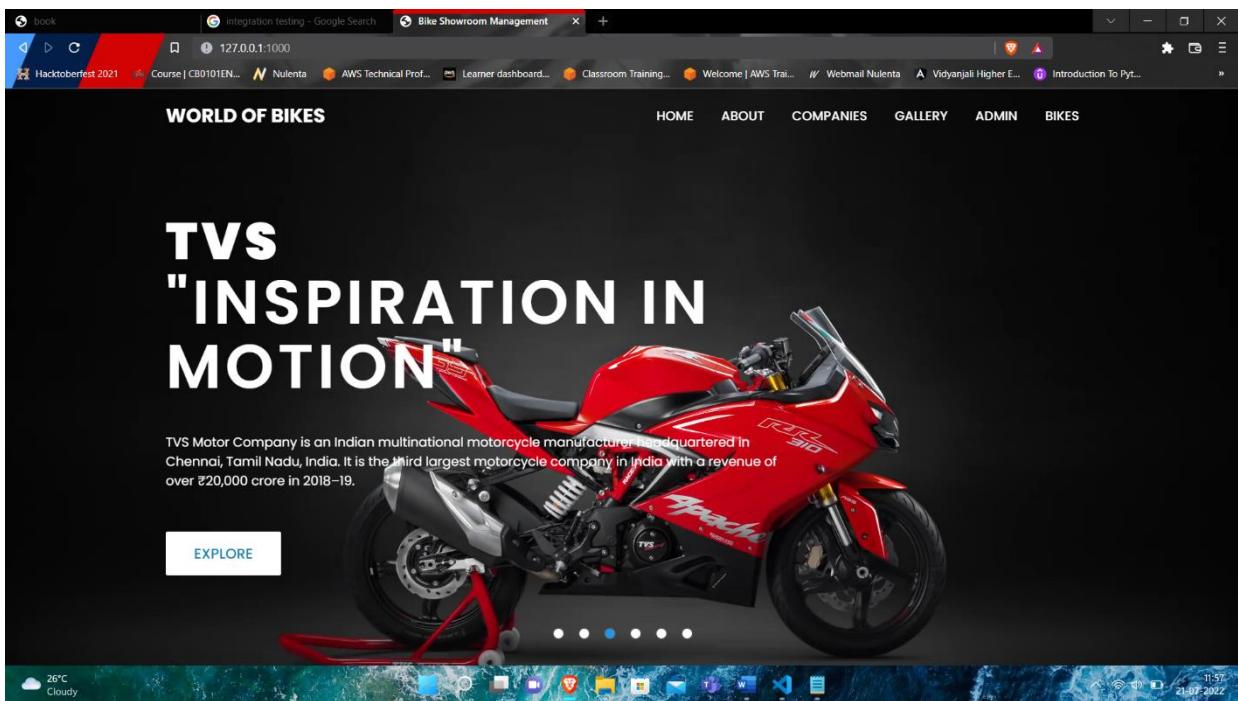


Fig 4.6.10 Home Page

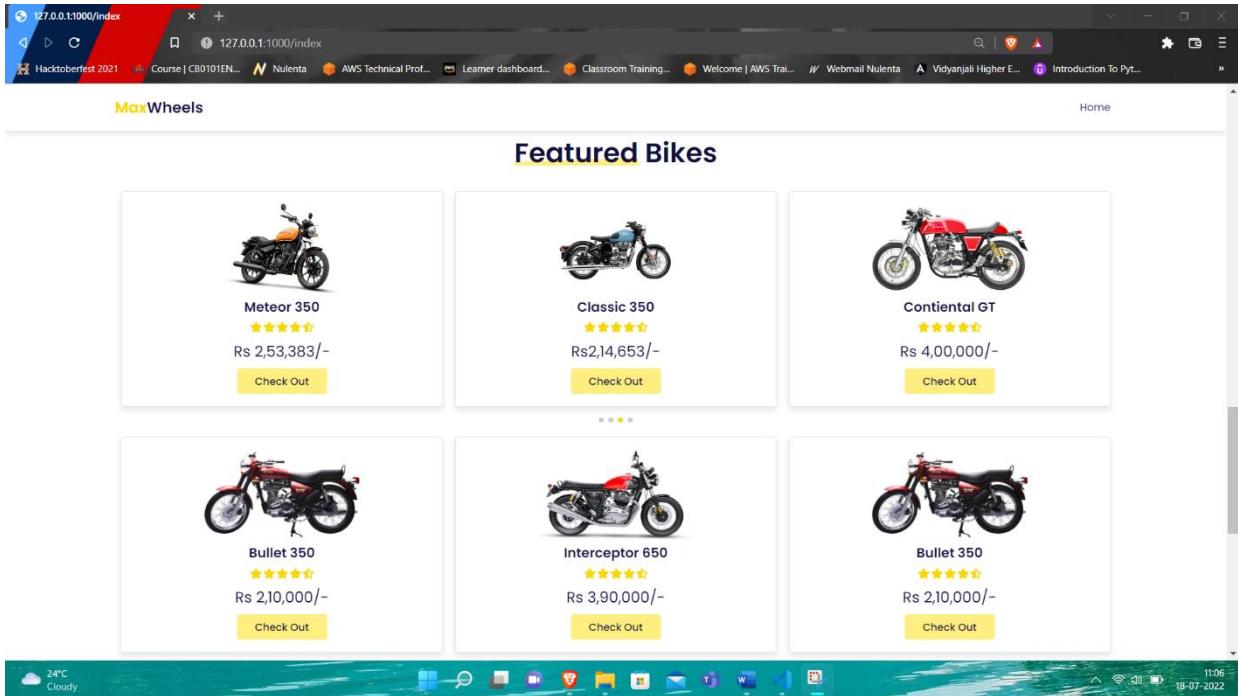


Fig 4.6.10 Checkout Page

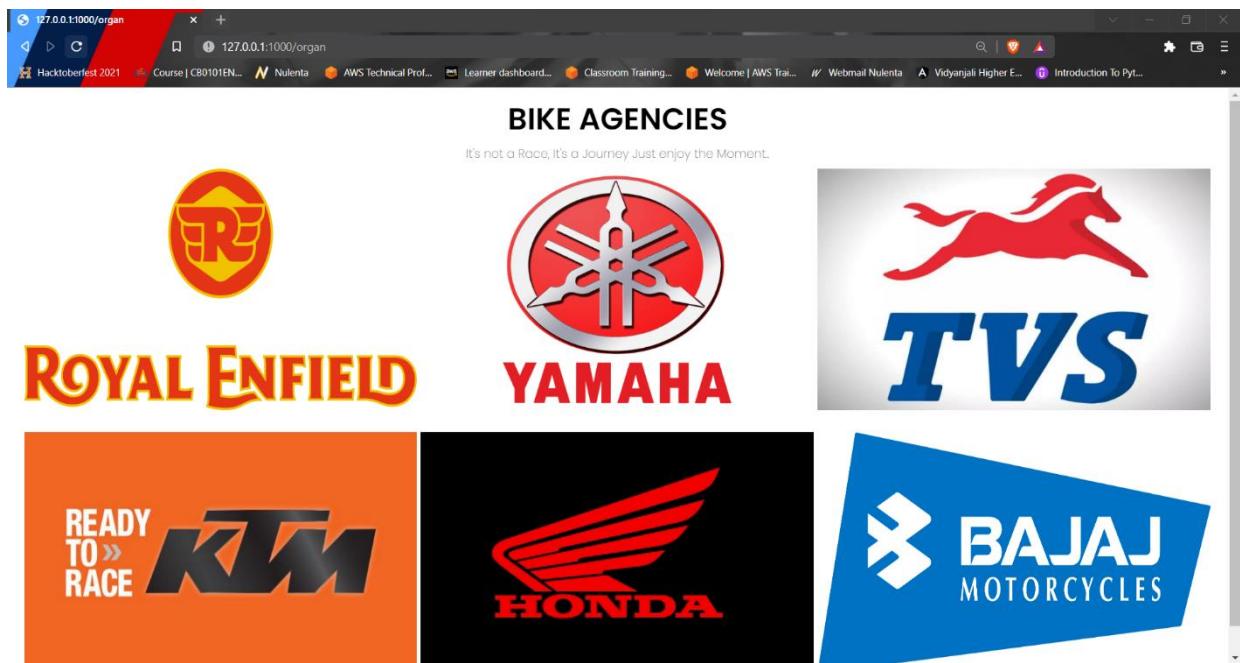
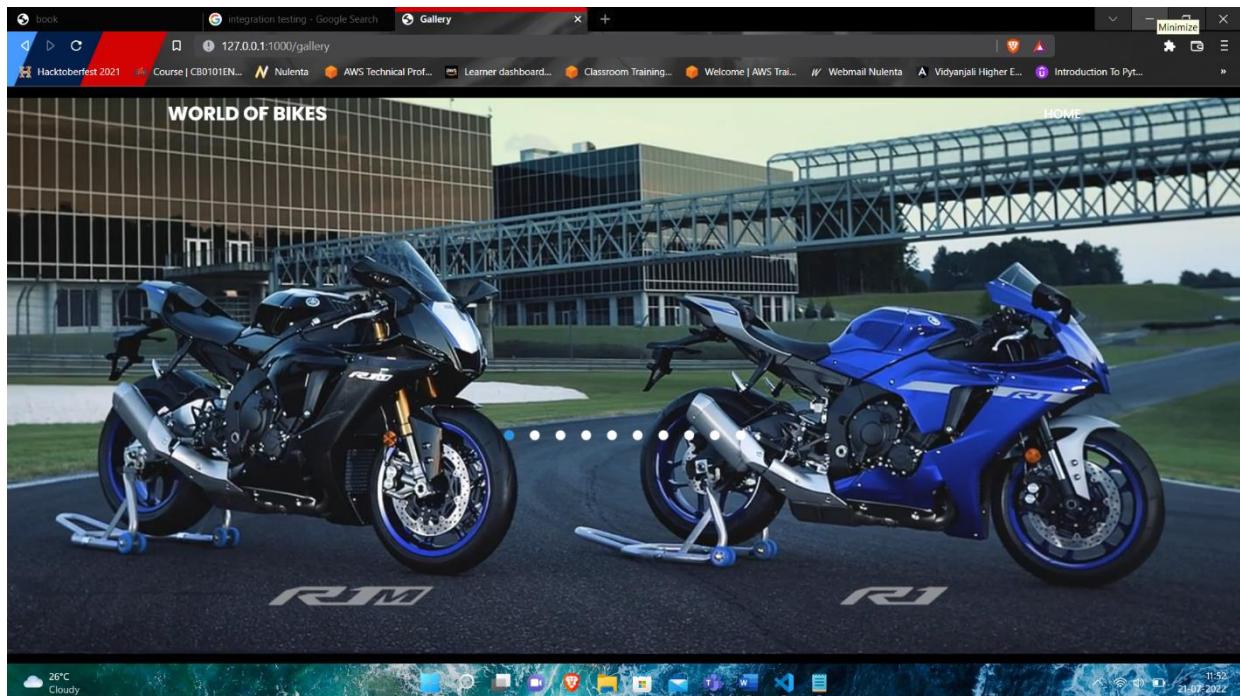
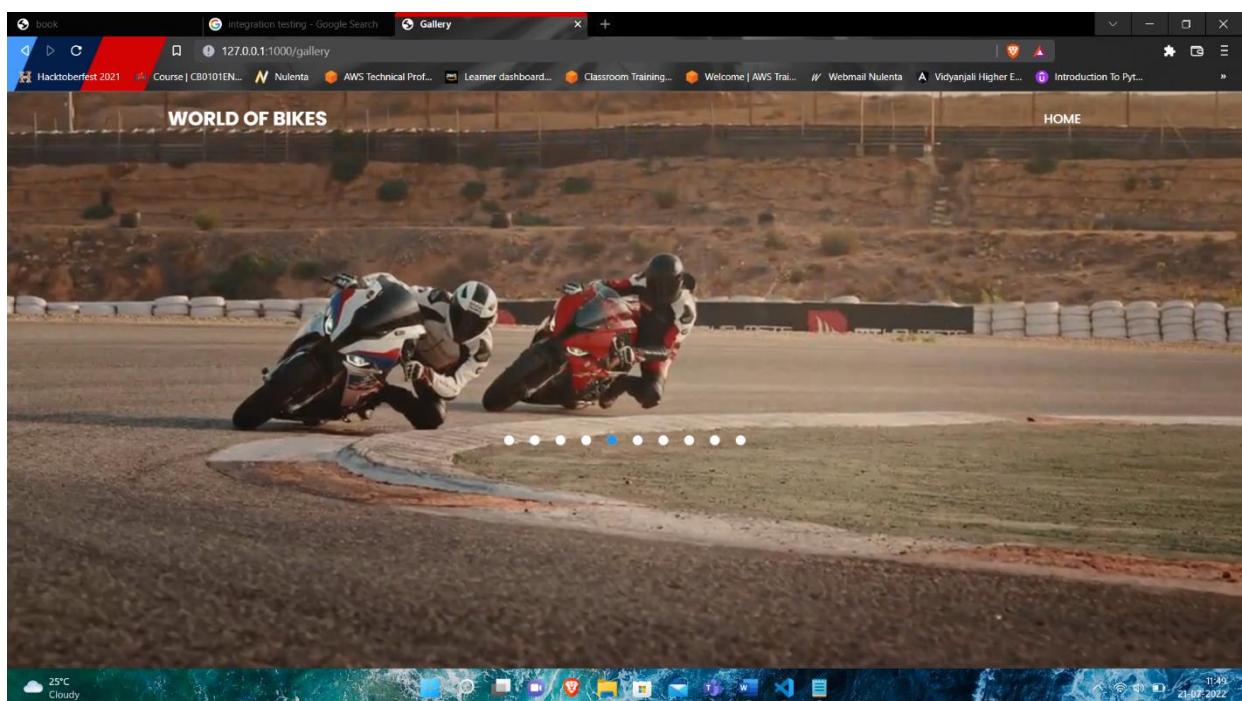
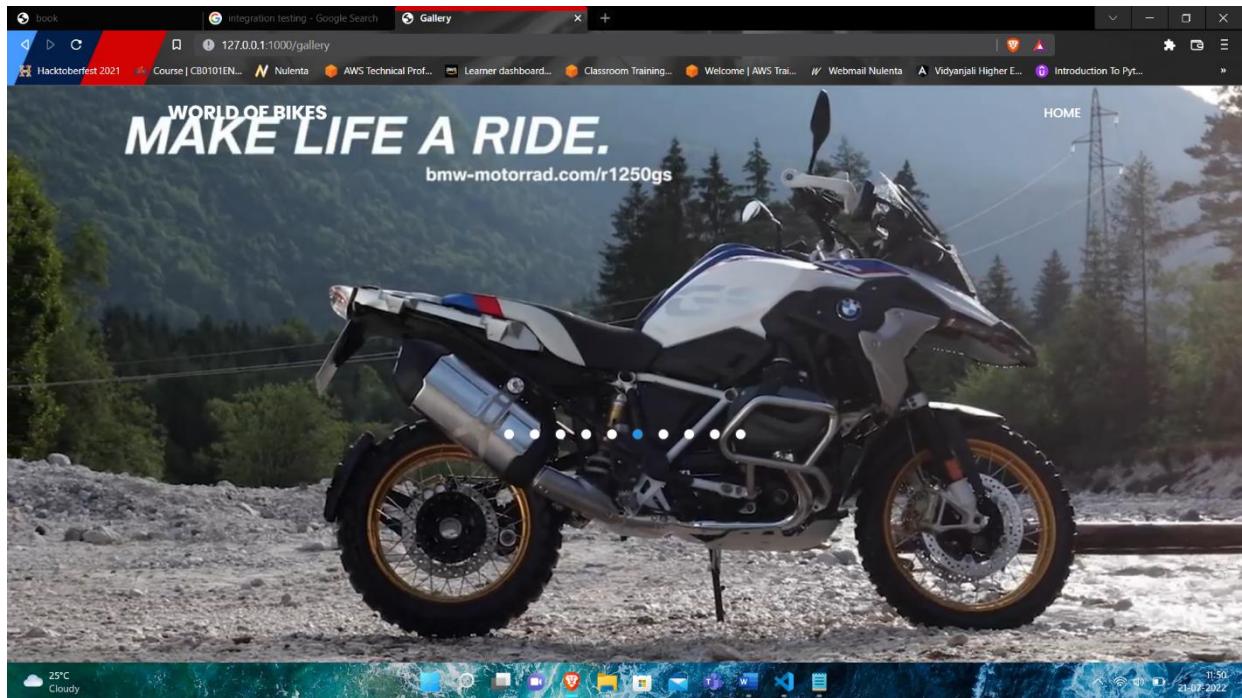
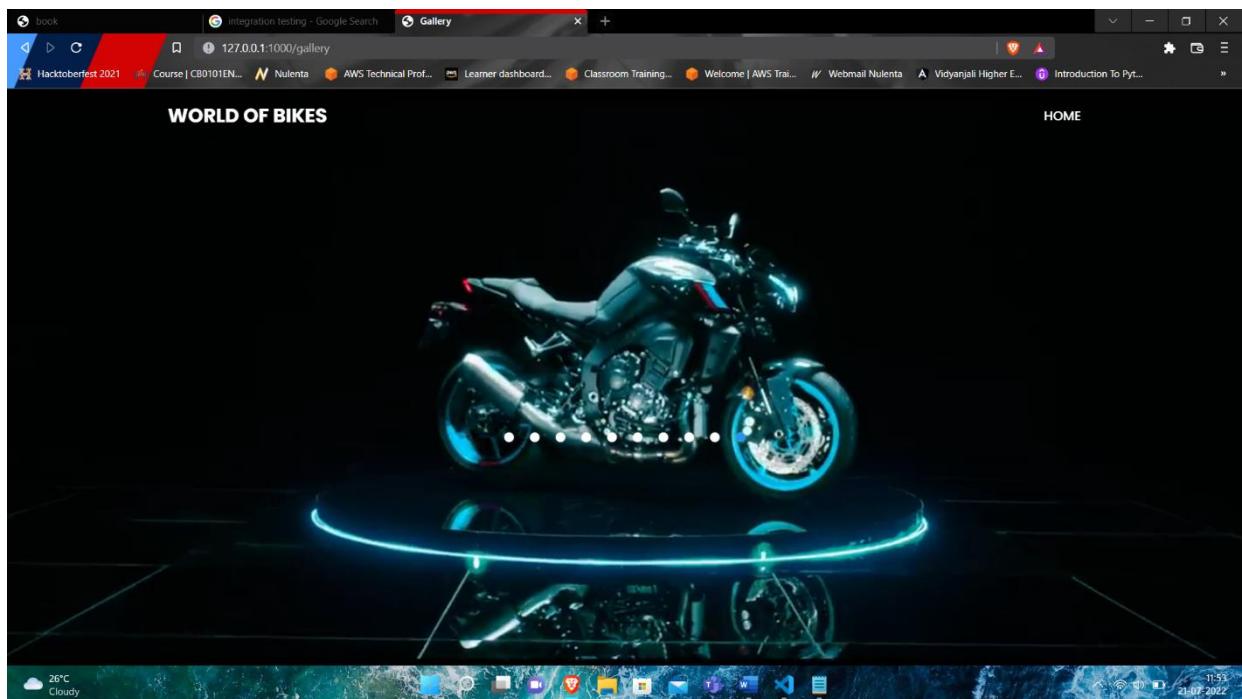
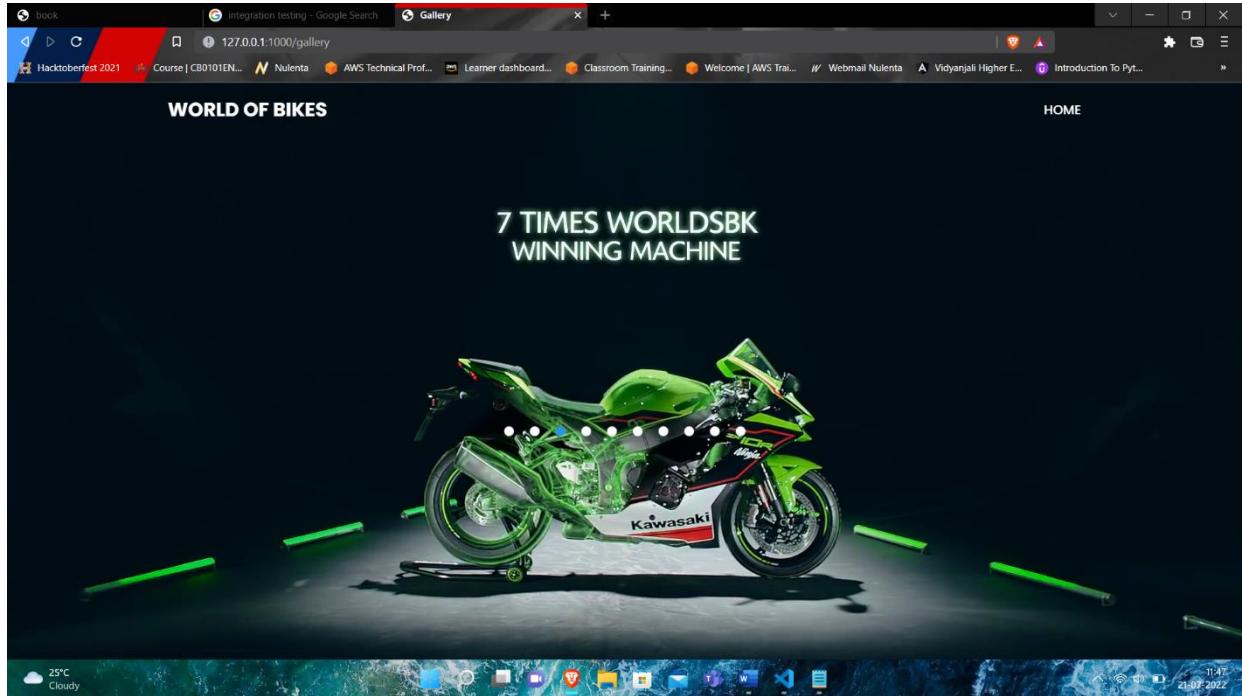


Fig 4.6.10 Bikes Agencies.

Fig 4.6.11 Bike Gallery







4.6.12 File contents

```

File Edit Selection View Go Run Terminal Help users.txt - FS Mini Project - Visual Studio Code
EXPLORER admin.html admin.txt users.txt
1 |tarun|ec9cdb815d0f8eedb7a29f478418f95dec86a7afe3ffa6fcbb393782bf036ff33ae393db32c3c059172c070e36efb9b767441b87750518b68d7f3d82c4a99a55de53c
2 |gagan|8760ea6c23fde3dc601f0a36ca0261b0e643aa1ed4209020890a85556181cac13efaa035da891c9df2a7ba27adcaf805dfb84684c66bf937d15504e4279a9cf7c8
3 |harsha|b867dd64aaee29caa8fb0d91375c0334d49e1a2100b679a0129fc5fdde3afef1bf1df77485d523866ced9fcf9f082d16630fd426b136d76a7030880d9e10b93cf659
4 |pawan|3ddc6961a2bd3fd170c94b11cd04bf70085631c7fa4da6198f78eae5d5734bcd21fe4d8108742499debe3a315c8d4ad94c4dc31cc554d9c2f0a4b1c2d42964d5c
5 |ujlas|fabf3bea31cf0515679c7cd7fc8166239d8f04e6a8641ac8f94e4a5d7ffeb85ed4f81315a4c0e8050d34cf869a7f5e90076544d3c60002def64acf9de00a48f853
6 |ambika|701f4718b41e66ba4d407cb9b04227b67e647a282164d3dcc98f6443a0d66f796720b26b8d013237ada10d25c69eb8597442bf735da4a45e16c0639361cd920e
7 |ambika1|64ae55f355a8971bfac7a1857b73f230adc138347d02d23df598133746f0e730a03b5bc100c2197632676690bf1fae379bc169dec87ae0a2076abbedfafb1ca
8 |ashwin|274711a150ffc31a45a139c8eb70f397f8d80e9c01d44edc61a88956d1e908f1111e680533e17cc2409df4856e8cdee4c3668ef8de88fe131e4aabe6881115d8
9 |pranam|d705807fd3b602e13c401822bd7700103110a208efbd3de8e169f9ef8674d19a3989a1b52a23deaaeb3b6b5718c6cf3107d90c78a63aac55ddc7033663b606378
10 |ananya|b7bc09087d129ca154341d1a17437b9d7f57d61dff56fabef62295968a4f7e3cef0a7a786d822f51451da327320e2101986dd430b600163bdb401813f848fc8b2
11

```

Fig 4.6.12 users.txt

```

File Edit Selection View Go Run Terminal Help admin.txt - FS Mini Project - Visual Studio Code
EXPLORER admin.html admin.txt users.txt
1 |rohit|9bb1eeb99a97cda3eb5713716898bb5a9fd0abcf794cf8505f3877a19b5e8f424145b55866e6c5dc843857475b0c4aaed2ad88632f37b131b9fe7e6c14b5548a262
2 |rohan|61aa203aaa2443f2bd8ed9a97263dd4b585a9976a509acf1185dd4514e26134eb4f6b2bcf763e2efbcfb0d52b537ec68cc7b50e0251040861ea2
3
4

```

Fig 4.6.12 admin.txt

BIKE SHOWROOM MANAGEMENT SYSTEM

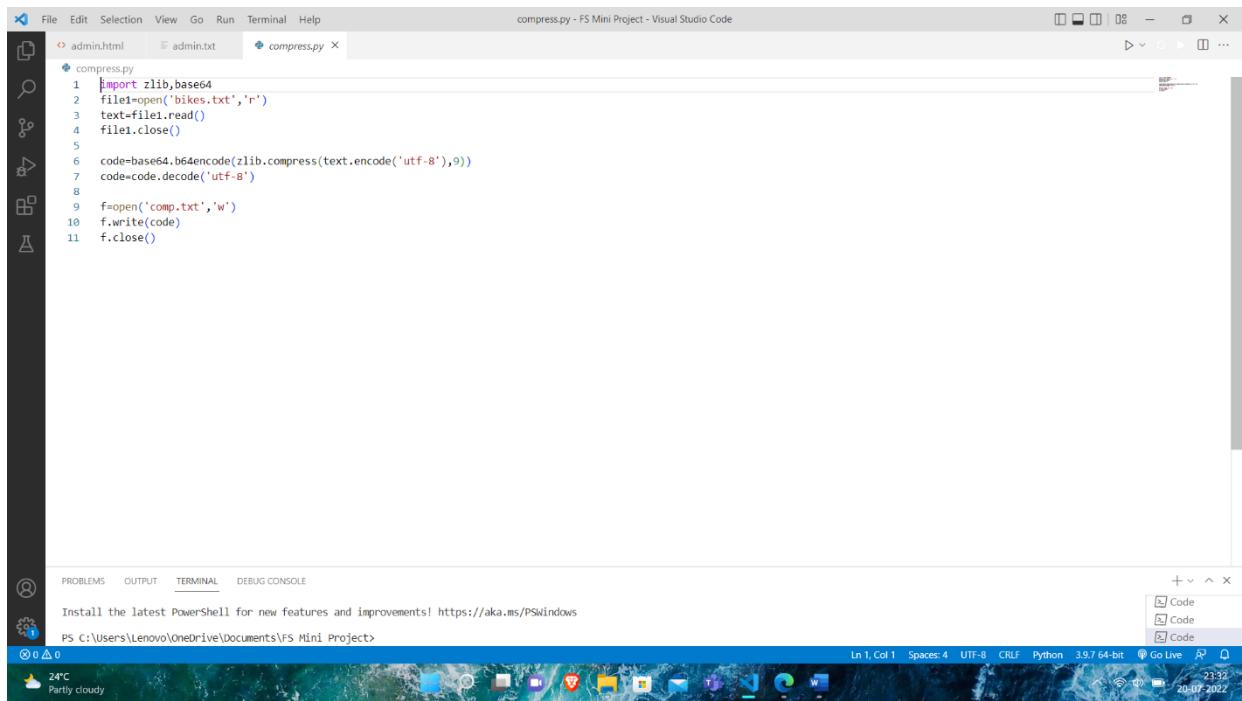
Fig 4.6.12 bikes.txt

The screenshot shows a Visual Studio Code window with the title "bookings.txt - FS Mini Project - Visual Studio Code". The left sidebar contains icons for file operations like Open, Save, Find, and Delete. The main editor area displays the contents of the "bookings.txt" file, which lists 11 entries, each consisting of a number, name, location, contact information, and email address. The bottom status bar shows the file path "PS C:\Users\Lenovo\OneDrive\Documents\FS Mini Project>" and the current date and time "20-07-2022 23:30".

```
1 020|tarun|continental gt|987654321|M|tarun@gmail.com|BANGALORE |LALBAGH
2 7220|gagan|continental gt|987654321|M|gagan@gmail.com|BANGALORE |8th mile
3 14520|harsha|domino|400|987654321|M|harsha@gmail.com|BANGALORE |sunkad
4 21720|pawan|continental gt|54545|M|pawan@benagaluru|sunkadkatte
5 28320|ullas|mt-15|987654321|M|ullas@mail.com|BANGALORE |kengeri
6 34820|amika|himalayan|548787|F|amika@gmail.com|BANGALORE |basaveshwaranagar
7 42620|feddf|ffff|ff|dfafasdfsadsadsads|ffff|erewerer|$
8 48820|ashwin|Interceptor|6366617340|M|ashwin@gmail.com|Bangalore|chikkabanavara|$
9 57020|pranam|continental gt|6366617340|M|pranam@gmail.com|Bangalore|balagunte|$
10 65120|ananya|continental gt|6366617340|F|ananya@gmail.com|Bangalore|balagunte|$
11
```

Fig 4.6.12 bookings.txt

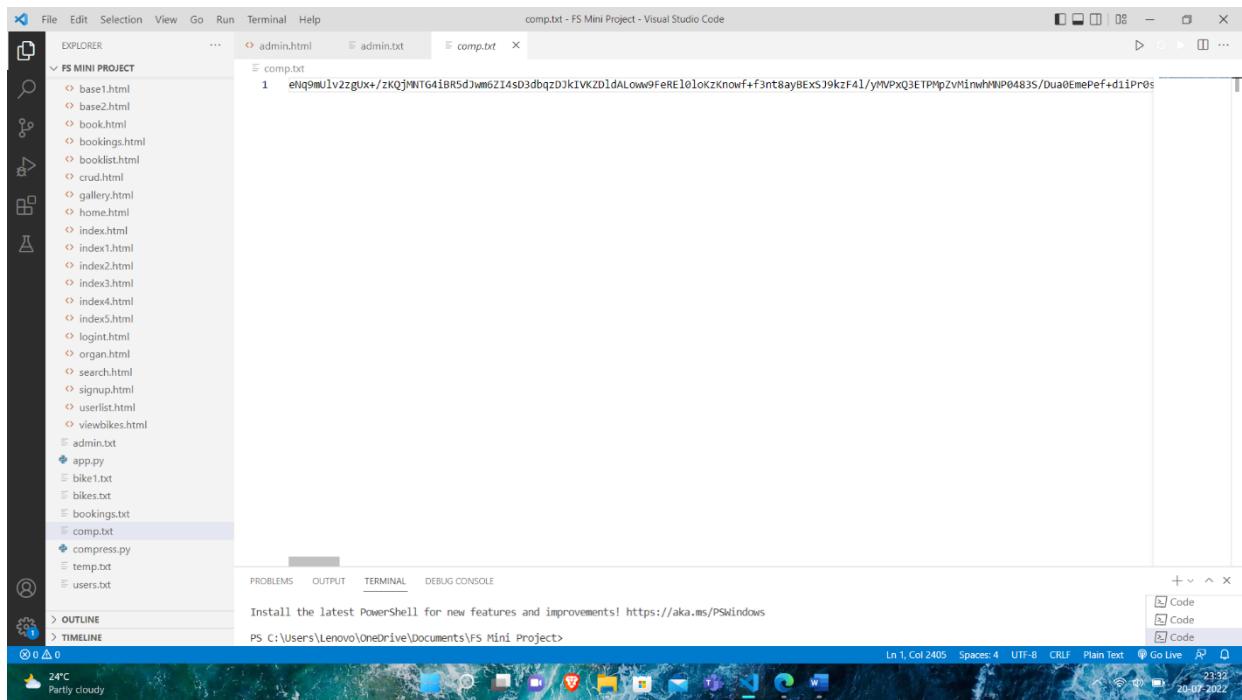
BIKE SHOWROOM MANAGEMENT SYSTEM



The screenshot shows a Visual Studio Code interface with the following details:

- File Explorer:** Shows files like admin.html, admin.txt, and compress.py.
- Code Editor:** Displays the Python script `compress.py` which reads a file, compresses its content, and writes it to another file.
- Terminal:** Shows the command `PS C:\Users\Lenovo\OneDrive\Documents\FS Mini Project>`.
- Bottom Status Bar:** Shows the current file is `compress.py`, line 1, column 1, spaces: 4, encoding: UTF-8, and file size: 39.7 64-bit.

Fig 4.6.12 file to be compressed



The screenshot shows a Visual Studio Code interface with the following details:

- File Explorer:** Shows a large number of HTML files (base1.html, base2.html, bookings.html, booklist.html, crud.html, gallery.html, home.html, index.html, index1.html, index2.html, index3.html, index4.html, index5.html, login.html, organ.html, search.html, signup.html, userlist.html, viewbikes.html) and other files (app.py, bike1.txt, bikes.txt, bookings.txt, comp.txt, compress.py, temp.txt, users.txt).
- Code Editor:** Displays the file `comp.txt` containing a single line of highly compressed binary data.
- Bottom Status Bar:** Shows the current file is `comp.txt`, line 1, column 2405, spaces: 4, encoding: UTF-8, and file size: 23.32 KB.

Fig 4.6.12 compress.txt

CHAPTER 5

CONCLUSION AND FUTURE ENHANCEMENT

The web-based **Bike Showroom** is an attempt to book bikes for users. Our project “**Bike Showroom Management System**” is only a humble venture to satisfy the needs to manage their project work. Several User friendly coding have been adopted. This package shall prove to be a powerful package in satisfying all the requirements of the users. The Objective of software planning is to provide a frame work that enables the manager to make reasonable estimates made within a limited time frame at the beginning of the software project and should be updated regularly as the project progresses.

In Future, We have planned to Create the master and slave database structure to reduce the overload of the database queries and also Implement the backup mechanism for taking backup of codebase and database on regular basis on different servers. In Future the “**Bike Showroom Management System**” will be keener to use, more user interactive and more dependable.

REFERENCES

- [1] www.google.com
- [2] www.stackoverflow.com
- [3] www.youtube.com
- [4] <https://palletsprojects.com/p/flask>
- [5] www.geeksforgeeks.org
- [6] www.python.org