" Free Parking Slots Management System "

A PROJECT REPORT SUBMITTED TO

THE NATIONAL INSTITUTE OF ENGINEERING, MYSURU

(An Autonomous Institute under VTU, Belagavi)

In partial fulfillment of the requirements for Project work (Database Laboratory CS5L02), fifth semester

Bachelor of Engineering

in

Computer Science and Engineering

Submitted by

Abhishek A P (4NI19CS004)

Bhargav N (4NI19CS029)

Bhuvan K (4NI19CS032)

Under the Guidance of

Priyanka R V Assistant Professor	Tejaswini N Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

2021-2022

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

THE NATIONAL INSTITUTE OF ENGINEERING

ESTD : 1946

CERTIFICATE

This is to certify that the project work entitled "**Free Parking Slots Management System**" is a work carried out by **Abhishek A P(4NI19CS004), Bhuvan K(4NI19CS032) and Bhargav N (4NI19CS029)** in partial fulfilment for the project work (Database Laboratory – CS5L02), fifth semester, Computer Science & Engineering, The National Institute of Engineering **(**Autonomous Institution under Visvesvaraya Technological University, Belagavi) during the academic year 2021-2022. It is certified that all corrections and suggestions indicated for the Internal Assessment have been incorporated in the report deposited in the department library. The project work report has been approved in partial fulfilment as per academic regulations of The National Institute of Engineering, Mysuru.

Signature of the Internal Guides **Signature of the HoD**

Priyanka R V Assistant Professor Dept. of CS&E NIE, Mysuru	Tejaswini N Assistant Professor Dept. of CS&E NIE, Mysuru	Dr. V K Annapurna Professor and Head Dept. of CS&E NIE, Mysuru

Signature of the Examiners with date

Name: Designation:	Name: Designation:

Acknowledgement

It is a great pleasure for us to express our gratitude to our project supervisors Priyanka R V and Tejaswini N for giving us the opportunity to do a project on ‘Free Parking Slots Management System’, and for their encouragement, appreciation and intellectual zeal which motivated us to venture this project successfully. Their suggestions and feedback have helped us a lot in improving the quality of the project.

- **Abhishek A P**
- **Bhuvan K**
- **Bhargav N**

Table of Contents

List of Figures

Chapter 1

Introduction

The number of vehicles in cities has increased dramatically due to rapid economic development. However, the infrastructure for accommodating these vehicles has grown relatively slow. Alleviating the pressure on the urban transport system and solving the 'parking difficulty' problem have thus become hot topics recently. In this paper, an intelligent parking management website designed using HTML, MYSQL and Python is presented to solve this problem. An algorithm which detects the presence of vehicles in parking spaces in a parking lot is designed and field test results are presented. Our results show that this system has an acceptably high accuracy with low cost, high feasibility, high efficiency and hence is recommended for wide use.

In the current scenario of parking management system, optimal usage of parking space and the abruptness to park vehicles are critical factors. Technology based Parking Management System is an automated and advanced solution that provides management of vehicles right from an entry in the parking area to the exit. Optimizing the parking space for vehicles is still a problematic area for businesses, Government offices, various public places as well as municipalities authorities in multiple cities across India.

1.1 Project Description

This project provides the whereabouts of parking areas in Mysore to users. It also checks for the availability of parking slots in the parking area. The basic requirement for this project is a webpage for interaction between users and the database. The webpage developed is named as **Skie Parking.** We have used HTML for webpage development (Front end), MYSQL for storing and updating data (Back end) and connected the webpage to the database using Python Flask.

The above implementation helps in optimizing parking, reducing traffic, enhanced user experience, integrated payments and POS increased safety.

Overall, this is a complete mini project in order to implement a fully functional and relational database.

1.2 Objective

The aim of implementing Parking Management System is to reduce time and increase efficiency of the current Parking Management System. In overpopulated cosmopolitan zones, parking strategies must be well implemented for management of vehicles. The system provides details of the vacant parking slots in the vicinity and reduces the traffic issues due to illegal parking in the vicinity. It is designed with an objective to meet the requirements of controlled parking that offers effortless parking tactics to the authorities.

1.3 Analysis

Skie Parking is an online website used to reserve a parking slot in a particular parking area. This webpage can be analysed under three views:

1.3.1 Admin
1.3.2 Parking area Manager
1.3.3 User

1.3.1 Admin

The admin refers to the webpage developer. He has to login with his User ID and password. The admin is granted access to add a parking area manager. Once the admin adds a parking manager, a mail is sent to the parking manager which includes the parking managers login credentials. The further process is continued by the Parking area Manager

1.3.2 Parking area Manager

The parking area manager is an employee working in the parking lot i.e., he is the one who knows the number of two-wheeler and four- wheeler slots in the parking lot. After the admin gives access to the parking area manager to add a parking lot, the parking area manager logs in to the website with the given credentials and adds the parking area code, parking area name, no of two-wheeler slots and the number of four-wheeler slots. An update option is available for the manager to make changes in the data he entered if any. This data is then displayed to the user.

1.3.3 User

The user can access the website by signing in. If he is a new user, he has to create an account and then sign in. The user sets his own email address and a password. A user can view the available parking slots in different parking areas and reserve a slot in the required parking area, but he cannot reserve a parking slot without logging in.

1.4 Scope and Relevance

It might be difficult to find reasonable parking, especially if you live in a densely crowded location.

Local drivers have an advantage since they are aware of traffic patterns, parking regulations, construction concerns, meter costs, and other factors. Thankfully, parking-related tension can be eased with the use of mobile apps.

Chapter 2

System Analysis

2.1 Existing System

Today's basic requirement under free parking slots management system is passenger service system which are not available in existing system. As per cutting throat business environment, customers and organizations need to be get information about each and every aspect. Existing parking management system control panel was not so advanced which can make frequent updating by using predefined rules, thus human intervention needed to make information updates every time. There was no comprehensive management tool, which can help 3 tier management architecture to work in co-ordination and gain maximum profit. Because of lacking in analysis, cost effective solutions were not provided to customers, thus not able to make maximum profits.

2.2 Proposed System

It's the parking management system which has been developed keeping in view to provide complete details about the parking slots. To make slot booking process easier and easily understandable to customers, slots and the details will be displayed using proper tables which can be easily understandable. As far security is concern, the accession will be based on concept of network access thus used advanced API to make system flexible. It's the system which is having appropriate distribution channel by which, organization can expand their business region. There are lot more features have been added to make an effective parking management system and some of these are: web based online booking system, proper authentication and validation to use correct data and eliminate the costly searching process. More optimized code with latest technologies including inventory control and booking system to make working process easier.

2.3 System Requirements

2.3.1 Hardware Requirements

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware. A hardware requirements list is often accompanied by a hardware compatibility list (HCL), especially in case of operating systems. An HCL lists tested, compatibility and sometimes incompatible hardware devices for a particular operating system or application. The following sub-sections discuss the various aspects of hardware requirements.

- Windows Platform
- 512 MB DDR2 RAM
- At least 5GB HDD space FREE
- Processor speed 1.7 GHz or higher

2.3.2 Software Requirements

Software Requirements deal with defining software resource requirements and pre-requisites that need to be installed on a computer to provide optimal functioning of an application. These requirements or pre-requisites are generally not included in the in the software installation package and need to be installed separately before the software is installed.

- IDE: Visual Studio Code
- Software: XAMPP
- MySQL Database Server
- Front End: HTML
- Back End: Python, My SQL

Chapter 3

System Design

3.1 System Architecture

A typical three-layer structure is used in this system: the database layer, the application service layer, the user interface layer. System architecture is as shown:

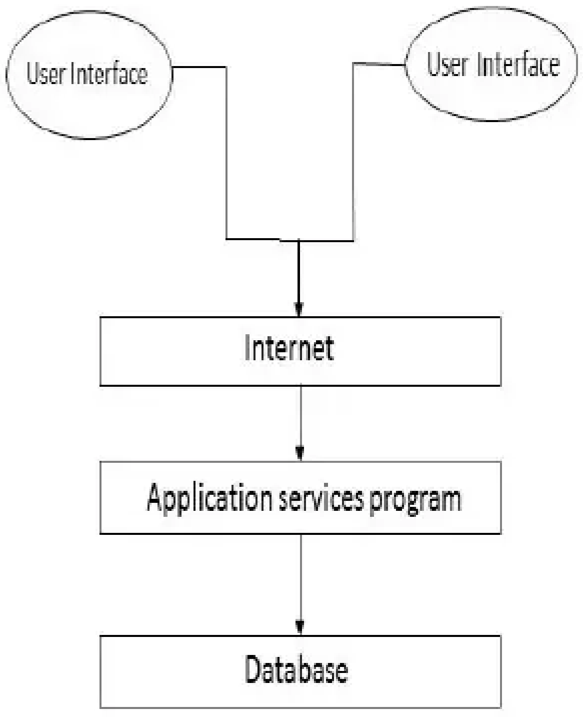


Figure 3.1 System Architecture

3.1.1 The Database Layer:

The database is used to hold data, including user registration information, ticket booking information, ticket information and all of the other information.

3.1.2 The Application Service Layer:

The application service layer is the core of this three-layer structure, the system functions and business logic are handled in this layer. In this layer, the system's business logic is encapsulated, the application service interfaces are provided for the user interface layer and the system modules between the function calls. The application service layer also updates data in the database, according to the service request of the top layer.

3.1.3 The User Interface Layer:

The user interface layer is a program that runs on a remote user computer. It displays the provided services by the server to the user. When the user selects a service, this program sends request to the server. When the server returns the processed result, this program shows it to the user.

Advantages:

- maintenance and understanding are easier.
- compatible with existing systems.

Disadvantages:

- However, this model gives poor performance when there are a large number of users.

3.2TABLE NAME AND ATTRIBUTES:

user	Id, name, email, phone, password
Parking user	Id, pcode, email, password
Pdata	Id, pcode, pname, fwslots, twslots
Parking	Id, vehnum, ptype, pcode, name, phone
trig	

3.3 Entity Relationship Diagram:

An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how entities such as people, objects or concepts relate to each other within a system. ER Diagrams are most often used to design or debug relational databases in the fields of software engineering, business information systems, education and research. Also known as ERDs or ER Models, they use a defined set of symbols such as rectangles, diamonds, ovals and connecting lines to depict the interconnectedness of entities, relationships and their attributes. They mirror grammatical structure, with entities as nouns and relationships as verbs. ER diagrams are related to data structure diagrams (DSDs), which focus on the relationships of elements within entities instead of relationships between entities themselves. ER diagrams also are often used in conjunction with data flow diagrams (DFDs), which map out the flow of information for processes or systems. ER Diagrams are composed of entities, relationships and attributes. They also depict cardinality, which defines relationships in terms of numbers.

- Entity:

A definable thing such as a person, object, concept or event that can have data stored about it. Examples: a customer, student, car or product. Typically shown as a rectangle.

- Relationship:

How entities act upon each other or are associated with each other.
For example, the named student might register for a course. The two entities would be the student and the course, and the relationship depicted is the act of enrolling, connecting the two entities in that way. Relationships are typically shown as diamonds or labels directly on the connecting lines.

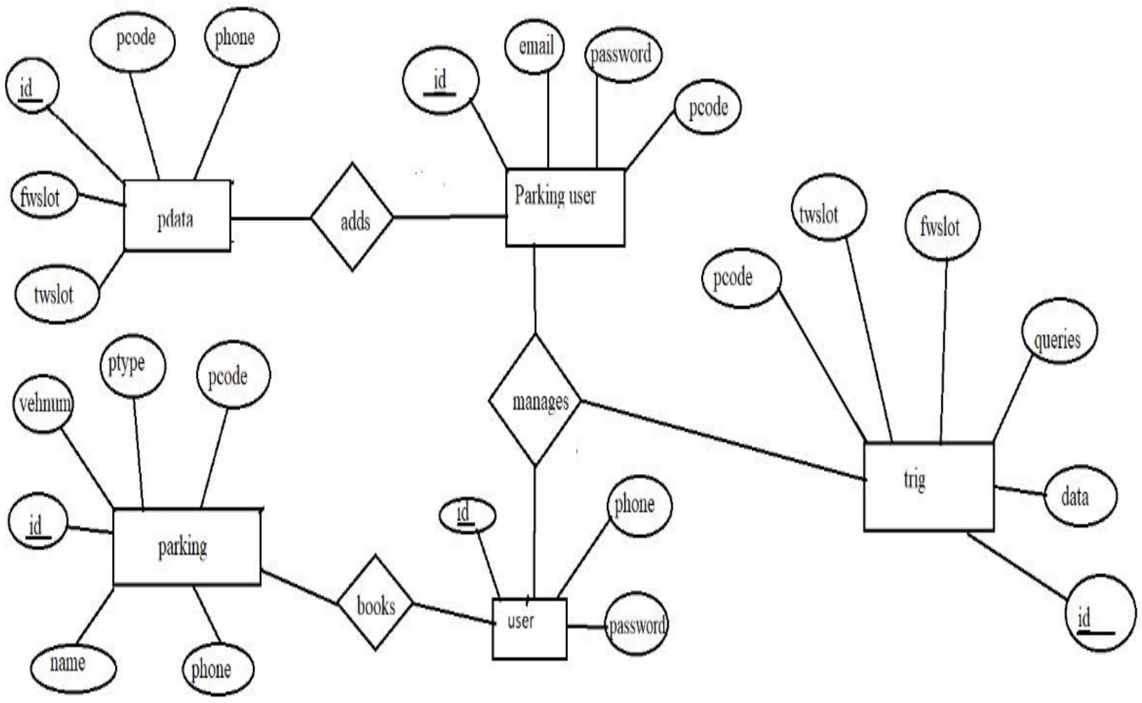
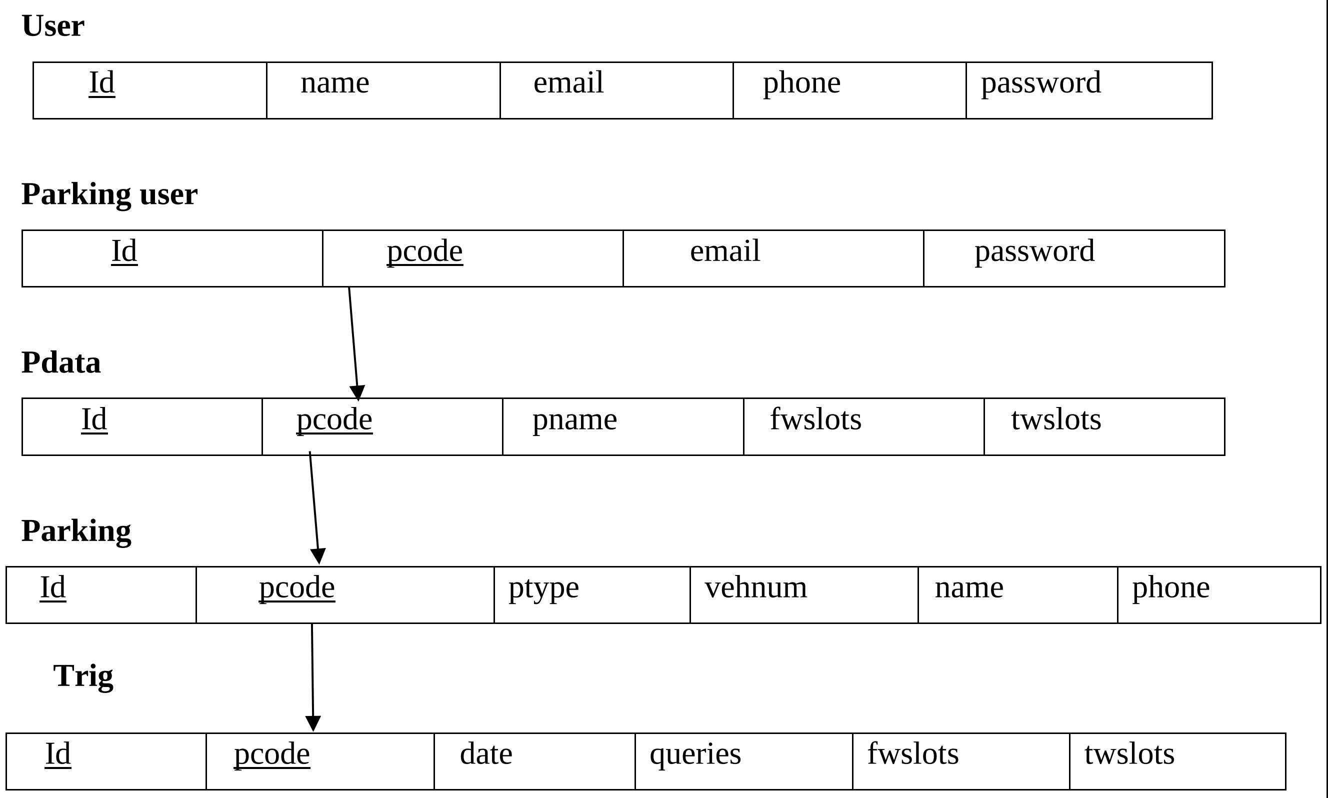


Figure 3.2 ER Diagram

3.4.1 SCHEMA DIAGRAM

A database schema is the skeleton structure that represents the logical view of the entire database. A database schema defines its entities and the relationship among them. It contains a descriptive detail of the database, which can be depicted by means of schema diagrams.



User

Id	name	email	phone	password

Parking user

Id	pcode	email	password

Pdata

Id	pcode	pname	fwslots	twslots

Parking

Id	pcode	ptype	vehnum	name	phone

Trig

Id	pcode	date	queries	fwslots	twslots

Chapter 4

System Implementation

4.1 Tools used for implementation

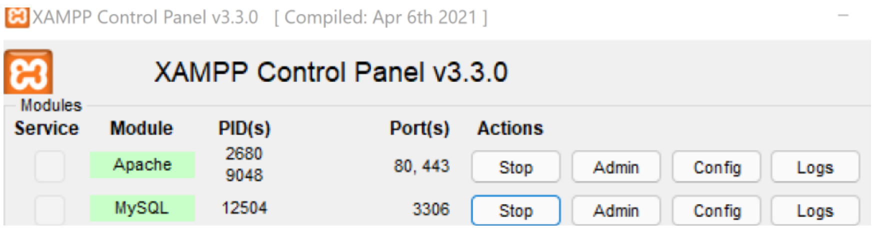
The software tool used for implementation of the system are XAMPP, phpMyAdmin, sublime text editor. This system is implemented or hosted on a localhost server which remotely works on the current computer system to receive and store all the data provided. phpMyAdmin is used to define the structure of the database which also handles the backend of this system. As a code editor sublime text editor is used to operate the frontend while accessing our database.

4.1.1 XAMPP

Figure 4.1 XAMPP Logo

XAMPP is one of the widely used cross-platform web servers, which helps developers to create and test their programs on a local webserver. It consists of **Apache HTTP Server, MariaDB, and interpreter** for the different programming languages like PHP and Perl. XAMPP is an abbreviation where *X stands for Cross-Platform, A stands for Apache, M stands for MYSQL and the Ps stand for PHP and Perl.* XAMPP helps a local host or server to test its website and clients via computers and laptops before releasing it to the main server. It is a platform that furnishes a suitable environment to test and verify the working of projects based on Apache, Perl, MySQL database, and PHP through the system of the host itself. Where PHP is a backend scripting language, works on different platforms such as Windows, Linux and macOS.

PHP: It is the backend scripting language primarily used for web development. PHP allows users to create dynamic websites and applications. It can be installed on every platform and supports a variety of database management systems. It was implemented using C language. PHP stands for Hypertext Processor. It is said to be derived from Personal Home Page tools, which explains its simplicity and functionality.



XAMPP Control Panel v3.3.0 [Compiled: Apr 6th 2021]

XAMPP Control Panel v3.3.0

Modules

Service	Module	PID(s)	Port(s)	Actions			
	Apache	2680 9048	80, 443	Stop	Admin	Config	Logs
	MySQL	12504	3306	Stop	Admin	Config	Logs

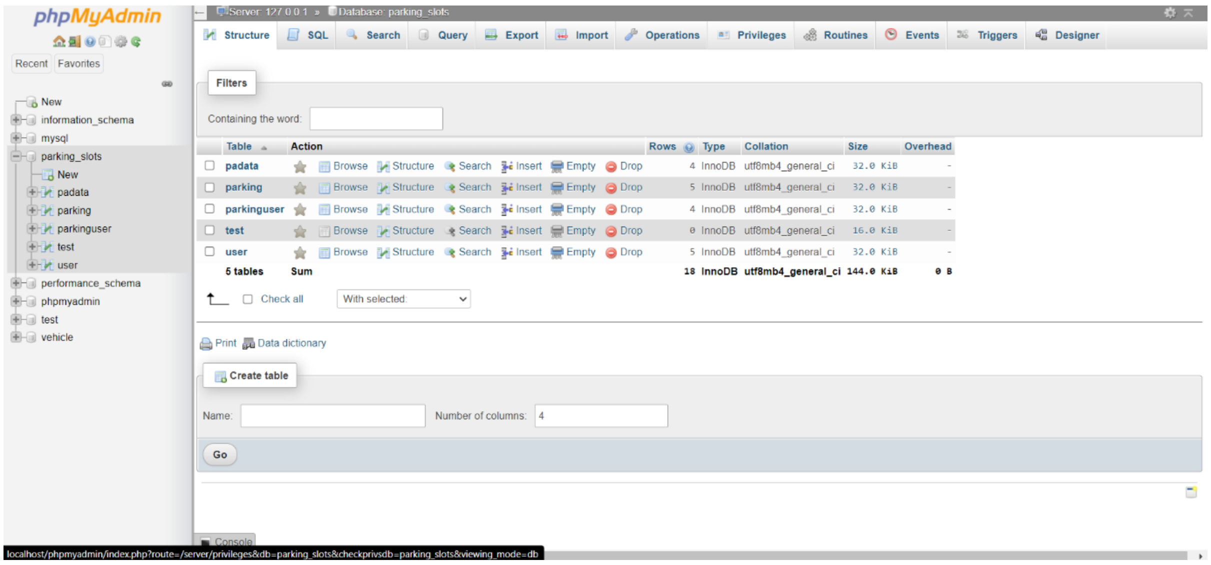
Figure 4.2 XAMPP Control Panel

4.1.2 phpMyAdmin

phpMyAdmin is an open-source software tool which is written in PHP. It is a tool to manage the tables and data inside the database. phpMyAdmin supports various type of operations on **MariaDB** and **MySQL.** The main purpose of phpMyAdmin is to handle the administration of MySQL over the web.

We can create, update, drop, alter, delete, import, and export MySQL database tables by using this software. phpMyAdmin also supports a wide range of operation like **managing databases, relations, tables, columns, indexes, permissions, and users,** etc., on MySQL and MariaDB. These operations can be performed via user interface, while we still have the ability to execute any SQL statement. phpMyAdmin can also be used to perform administrative tasks such as **database creation, query execution.** We can manually create database and table and execute the query on them.

It provides a web-based interface and can run on any server. Since it is web-based, so we can access it from any computer. It supports foreign keys and InnoDB table, can track the changes done on databases; views; and tables, also supports mysqli, which is the improved MySQL extension. It provides the facility to back up the database into different forms.



Containing the word:

Table	Action	Rows	Type	Collation	Size	Overhead
padata	Browse Structure Search Insert Empty Drop	4	InnoDB	utf8mb4_general_ci	32.0 KiB	-
parking	Browse Structure Search Insert Empty Drop	5	InnoDB	utf8mb4_general_ci	32.0 KiB	-
parkinguser	Browse Structure Search Insert Empty Drop	4	InnoDB	utf8mb4_general_ci	32.0 KiB	-
test	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
user	Browse Structure Search Insert Empty Drop	5	InnoDB	utf8mb4_general_ci	32.0 KiB	-
5 tables	Sum	18	InnoDB	utf8mb4_general_ci	144.0 KiB	0 B

Check all With selected:

Name: Number of columns: 4

localhost/phpmyadmin/index.php?route=/server/privileges&db=parking_slots&checkprivsdb=parking_slots&viewing_mode=db

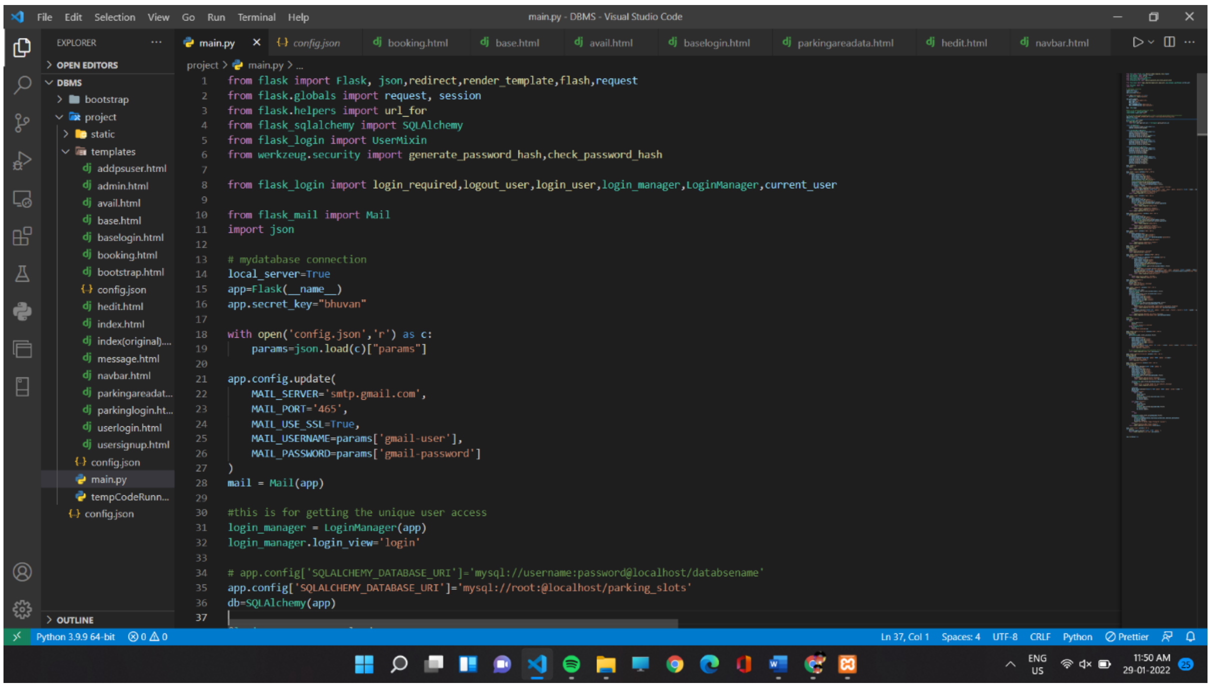
Figure 4.3 phpMyAdmin

4.1.3 Visual Studio Code

With support for hundreds of languages, VS Code helps us be instantly productive with syntax highlighting, bracket-matching, auto-indentation, box-selection, snippets, and more. Intuitive keyboard shortcuts, easy customization and community-contributed keyboard shortcut mappings let you navigate your code with ease.

For serious coding, we'll often benefit from tools with more code understanding than just blocks of text. Visual Studio Code includes built-in support for IntelliSense code completion, rich semantic code understanding and navigation, and code refactoring.

Assistance is provided to the user at each and every step so that no problem is faced during using it. Further the details of every process and the user manuals attached in the report make it very easy to understand. Every possible care has been taken to make the software and the report clear, simple and error free which makes it so special and one of its kind.



```python
from flask import Flask, json,redirect,render_template,flash,request
from flask.globals import request, session
from flask.helpers import url_for
from flask_sqlalchemy import SQLAlchemy
from flask_login import UserMixin
from werkzeug.security import generate_password_hash,check_password_hash

from flask_login import login_required,logout_user,login_user,login_manager,LoginManager,current_user

from flask_mail import Mail
import json

# mydatabase connection
local_server=True
app=Flask(__name__)
app.secret_key="bhuvan"

with open('config.json','r') as c:
    params=json.load(c)["params"]

app.config.update(
    MAIL_SERVER='smtp.gmail.com',
    MAIL_PORT='465',
    MAIL_USE_SSL=True,
    MAIL_USERNAME=params['gmail-user'],
    MAIL_PASSWORD=params['gmail-password']
)
mail = Mail(app)

#this is for getting the unique user access
login_manager = LoginManager(app)
login_manager.login_view='login'

# app.config['SQLALCHEMY_DATABASE_URI']='mysql://username:password@localhost/databsename'
app.config['SQLALCHEMY_DATABASE_URI']='mysql://root:@localhost/parking_slots'
db=SQLAlchemy(app)
```

Figure 4.4 VS Code

4.2 Database Connectivity

local_server=True

app=Flask(__name__)

app.secret_key="bhuvan"

4.3 Table Creation

User Table

```
class User(UserMixin,db.Model):
 id=db.Column(db.Integer,primary_key=True)
 name=db.Column(db.String(15))
 email=db.Column(db.String(50))
 phone=db.Column(db.String(13),unique=True)
 password=db.Column(db.String(1000))
```

Parkinguser Table

```
class Parkinguser(UserMixin,db.Model):
 id=db.Column(db.Integer,primary_key=True)
 pcode=db.Column(db.String(20),unique=True)
 email=db.Column(db.String(50))
 password=db.Column(db.String(1000))
```

Pdata Table

```
class padata(UserMixin,db.Model):
id=db.Column(db.Integer,primary_key=True)
pcode=db.Column(db.String(20),unique=True)
pname=db.Column(db.String(100))
fwslots=db.Column(db.Integer)
twslots=db.Column(db.Integer)
```

Parking Table

```
class parking(UserMixin,db.Model):
id=db.Column(db.Integer,primary_key=True)
 vehnum=db.Column(db.String(20),unique=True)
 ptype=db.Column(db.String(20))
 pcode=db.Column(db.String(20))
 name=db.Column(db.String(50))
 phone=db.Column(db.String(12))
```

Trigger Table

```
class trig(UserMixin,db.Model):
id=db.Column(db.Integer,primary_key=True)
pcode=db.Column(db.String(20),unique=True)
date=db.Column(db.String(100))
fwslots=db.Column(db.Integer)
twslots=db.Column(db.Integer)
queries=db.Column(db.String(100))
```

4.4 Python Flask Implementation

4.4.1 Signup Page

```
@app.route('/signup',methods=['POST','GET'])
def signup():
    if request.method=="POST":
        name=request.form.get('name')
        email=request.form.get('email')
        phone=request.form.get('phone_no')
        password=request.form.get('password')
        #print(name,email,phone, password)
        encpassword=generate_password_hash(password)
        userph=User.query.filter_by(phone=phone).first()
        usermail=User.query.filter_by(email=email).first()
        if usermail or userph:
            flash("Email or Phone number is already present","warning")
            return render_template("usersignup.html")
        new_user=db.engine.execute(f"INSERTINTO
`user`(`name`,`email`,`phone`,`password`)VALUES
('{name}','{email}','{phone}','{encpassword}') ")
        flash("Successfully Registered, please login","success")
        return render_template("userlogin.html")
    return render_template("usersignup.html")
```

4.4.2 Login Page

```
@app.route('/login',methods=['POST','GET'])
def login():
    if request.method=="POST":
        email=request.form.get('email')
        password=request.form.get('password')
        user=User.query.filter_by(email=email).first()
        if user and check_password_hash(user.password,password):
            login_user(user)
            return render_template("index.html")
        else:
            flash("Invalid Credentials","danger")
            return render_template("userlogin.html")
    return render_template("userlogin.html")
```

4.4.3 Parking Login

```
@app.route('/parkinglogin',methods=['POST','GET'])
def parkinglogin():
    if request.method=="POST":
        email=request.form.get('email')
        password=request.form.get('password')
        user=Parkinguser.query.filter_by(email=email).first()
        if user and check_password_hash(user.password,password):
            login_user(user)
            return render_template("index.html")
        else:
            flash("Invalid Credentials","danger")
            return render_template("parkinglogin.html")
    return render_template("parkinglogin.html")
```

4.4.4 Admin login

```
@app.route('/admin',methods=['POST','GET'])
def admin():
    if request.method=="POST":
        username=request.form.get('username')
        password=request.form.get('password')
        if(username==params['user'] and password==params['password']):
            session['user']=username
            return render_template("addpsuser.html")
        else:
            flash("Invalid Credentials","danger")
    return render_template("admin.html")
```

4.4.5 Logout:

```
@app.route('/logout')
@login_required
def logout():
    logout_user()
    flash("Logout Successful","warning")
    return redirect(url_for('login'))
```

4.4.6 Admin Logout:

```
@app.route('/logoutadmin')
def logoutadmin():
    session.pop('user')
    flash("Logout Successful","warning")
    return redirect('/admin')
```

Chapter 5

System Testing

5.1 System Testing as a Whole

Testing is the major quality measure employed during the software engineering development. Its basic function is to detect error in the software. Testing is necessary for the proper functioning of the system. Testing is done at four levels.

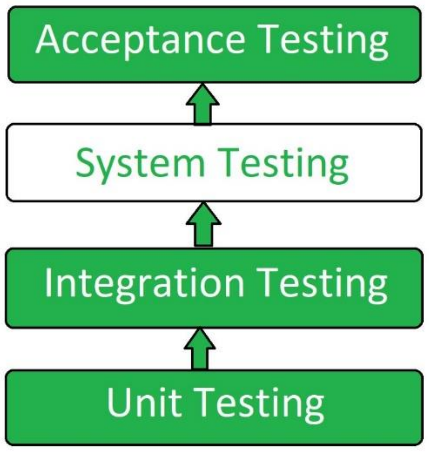


Figure 5.1 System Testing Levels

5.1.1 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.

5.1.2 Integration Testing

Integration testing is the process of testing the interface between two software units or module. It's focus on determining the correctness of the interface. The purpose of the integration testing is to expose faults in the interaction between integrated units. Once all the modules have been unit tested, integration testing is performed.

5.1.3 System Testing

Software testing can be stated as the process of verifying and validating that software or application is bug-free, meets the technical requirements as guided by its design and development, and meets the user requirements effectively and efficiently with handling all the exceptional and boundary cases.

5.1.4 Standard Definition of Acceptance Testing

It is a formal testing according to user needs, requirements and business processes conducted to determine whether a system satisfies the acceptance criteria or not and to enable the users, customers or other authorized entities to determine whether to accept the system or not.

5.2 Valid Inputs

System is tested for correct inputs and displayed message for errors if any during each time the input is taken through a form for processing.

One of the main constrains is that in any case, user cannot submit empty fields.

The following table illustrates the messages displayed for the corresponding labels if any constraint is violated when they are submitted.

Input field label	**Constraint and message displayed during error**
If any field is submitted as empty	Error message saying empty fields appear
If incorrect password is entered	Error message saying incorrect password appears
If proper inputs are not entered while signing up	Corresponding error message appears

The error messages are displayed on the screen of the particular page if constraints are not satisfied.

If data in only few of the fields are entered in correct format and if the other fields are left empty or in a wrong format, the error message is displayed only for those fields for which the constraints are not satisfied and the data which is entered in correct format in other fields is retained as it is.

Unless and until the user enters the correct data for all the input fields and unless form has no error messages, no further instruction or action is executed and it is not stored as input and user cannot be taken to the next web page and remain in the current web page to enter data in correct format satisfying all the constraints.

Chapter 6

Results

6.1 Admin Login Page

The admin login page is for the admin to log in with his username and password. The admin can login and carry out the operations required.

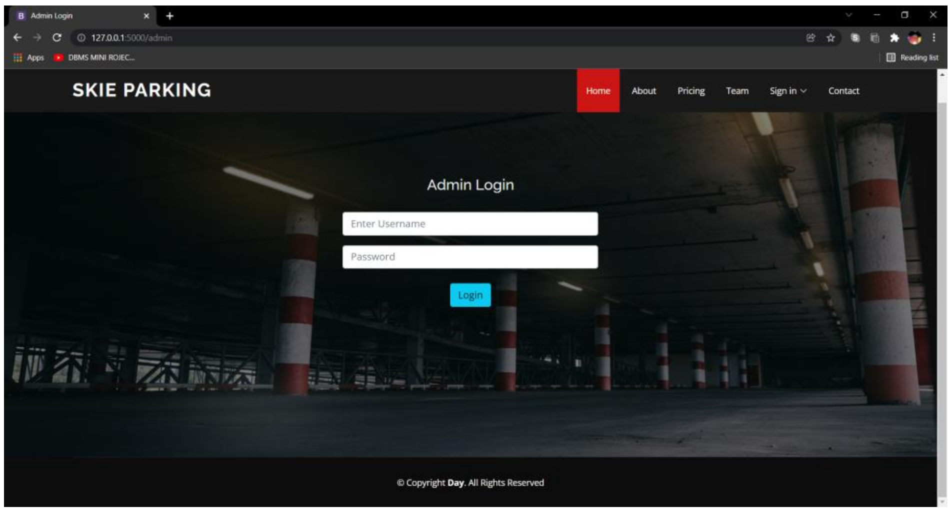


Figure 6.1 Admin Login Page

6.2 Parking area Manager Login Page

The parking area manager login page is for an employee working in the parking area to log in and update the information of the parking area.

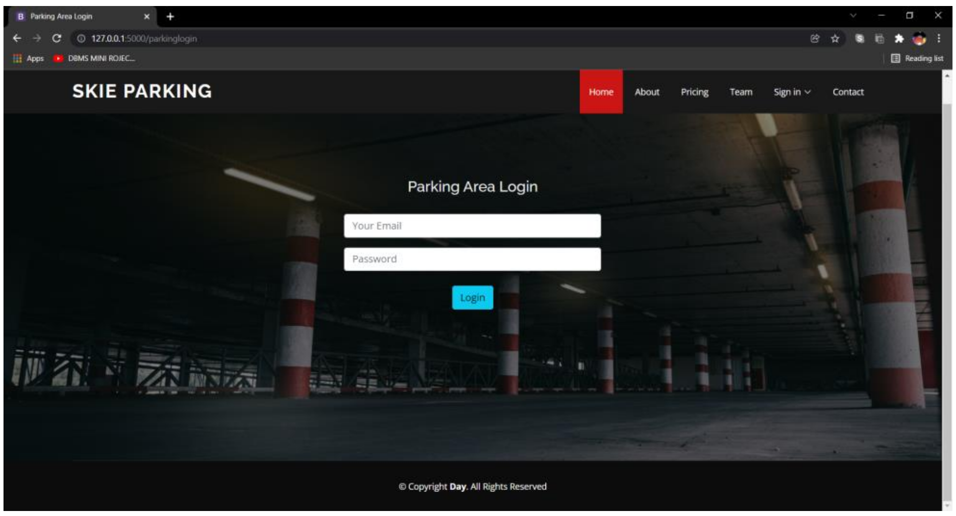


Figure 6.2 Parking area manager Login Page

6.3 User Login Page

Any user can view the data of parking areas by logging in. If he is a new user, he can sign up.

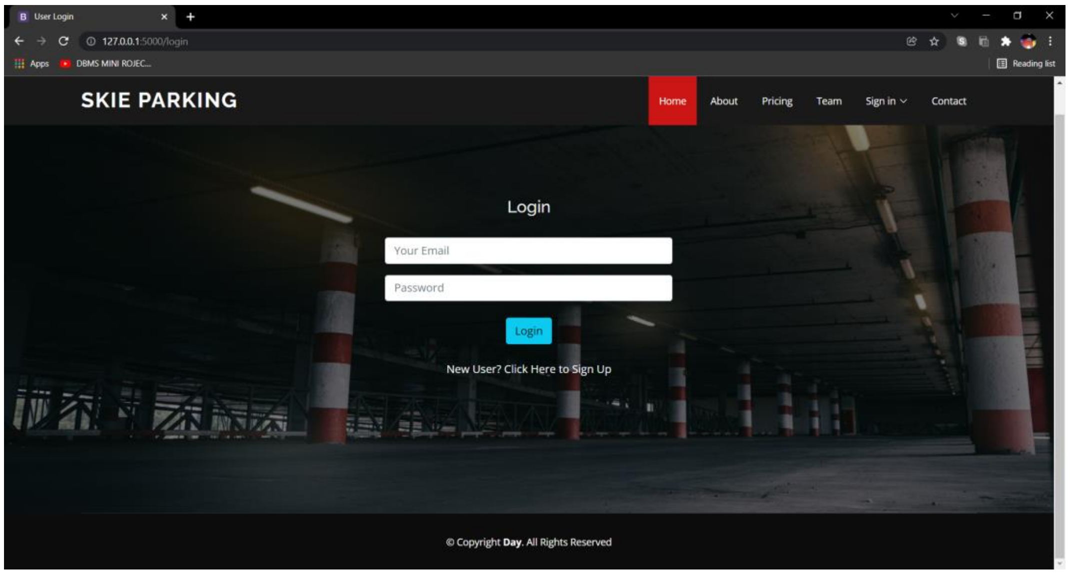


Figure 6.3 User Login Page

6.4 User Sign Up Page

Any new user to the website can register using the sign up option and then log in.

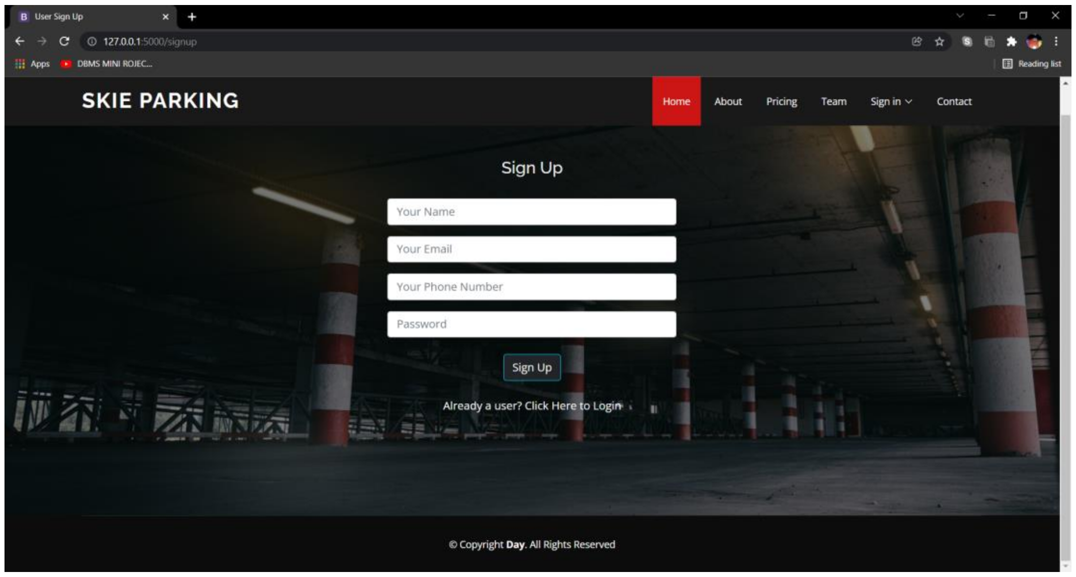


Figure 6.4 User Signup Page

6.5 Admin Dashboard

This page can be accessed only by the admin. The admin adds parking area managers in this page.

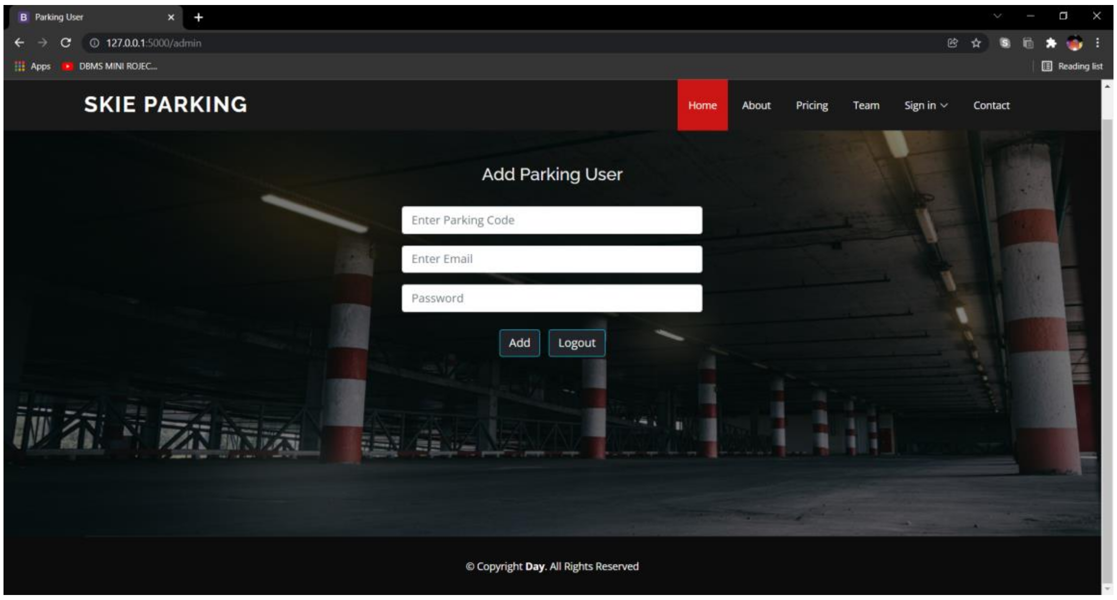


Figure 6.5 Admin Dashboard

6.6 Parking Area Manager Dashboard

This page can be accessed only by the parking area manager (employee working in the parking area). The parking area manager adds the parking areas data in this page.

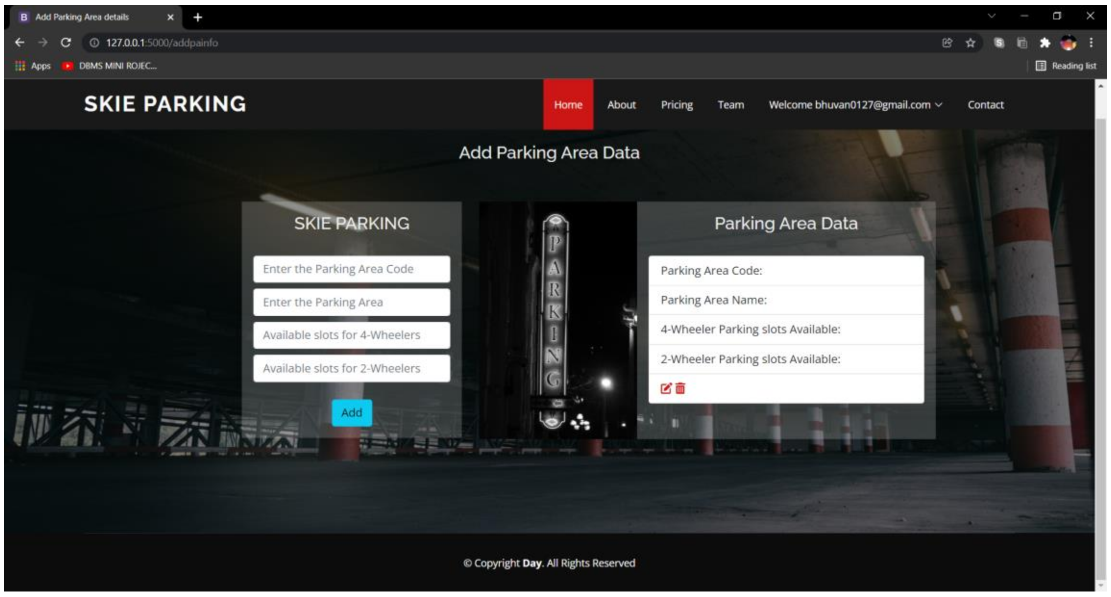


Figure 6.6 Parking Area Manager Dashboard

6.7 User Dashboard

The user can view this page once he logs in to the website.

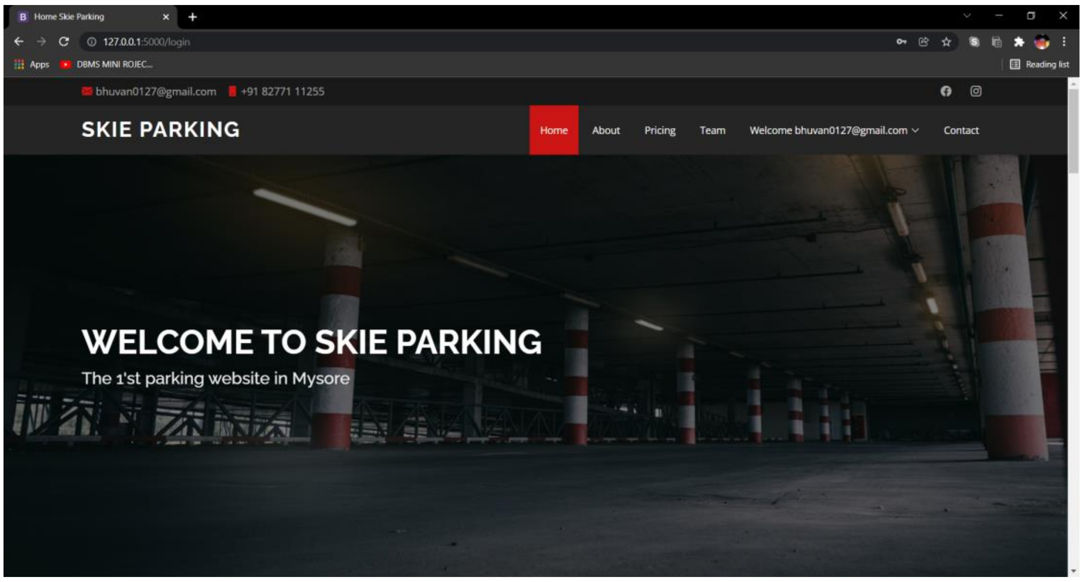
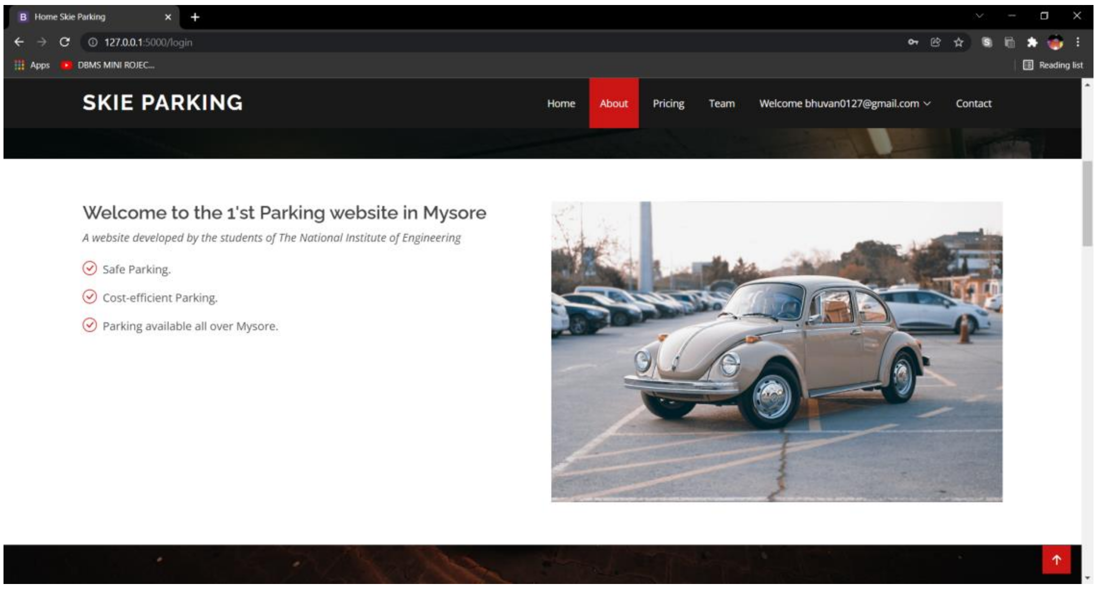


Figure 6.7 User Dashboard – Front Page

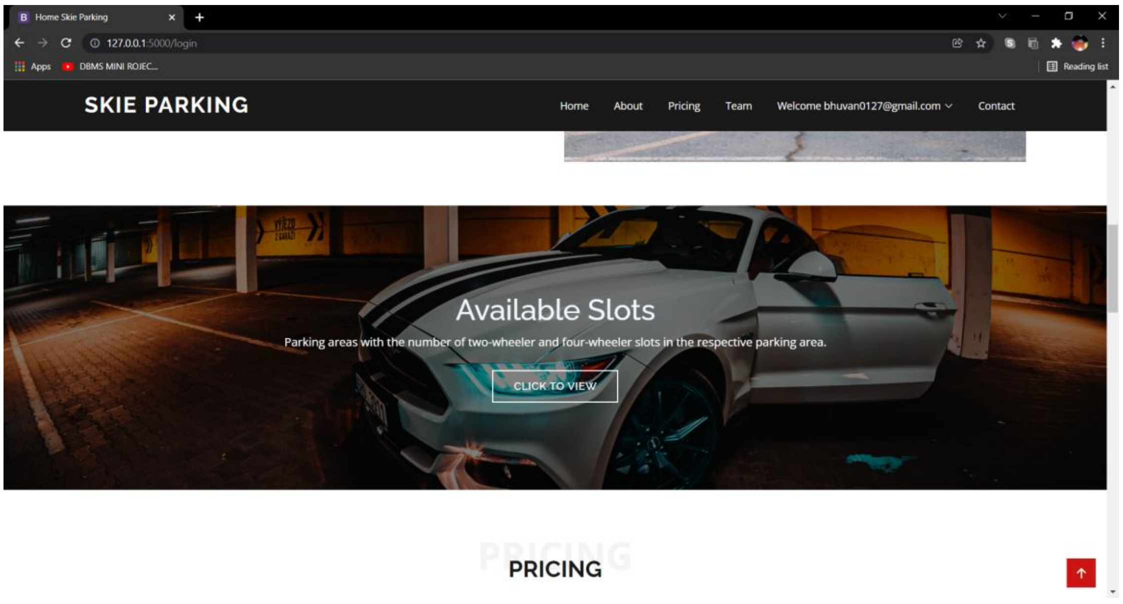


Welcome to the 1'st Parking website in Mysore

A website developed by the students of The National Institute of Engineering

- Safe Parking.
- Cost-efficient Parking.
- Parking available all over Mysore.

Figure 6.8 User Dashboard – About



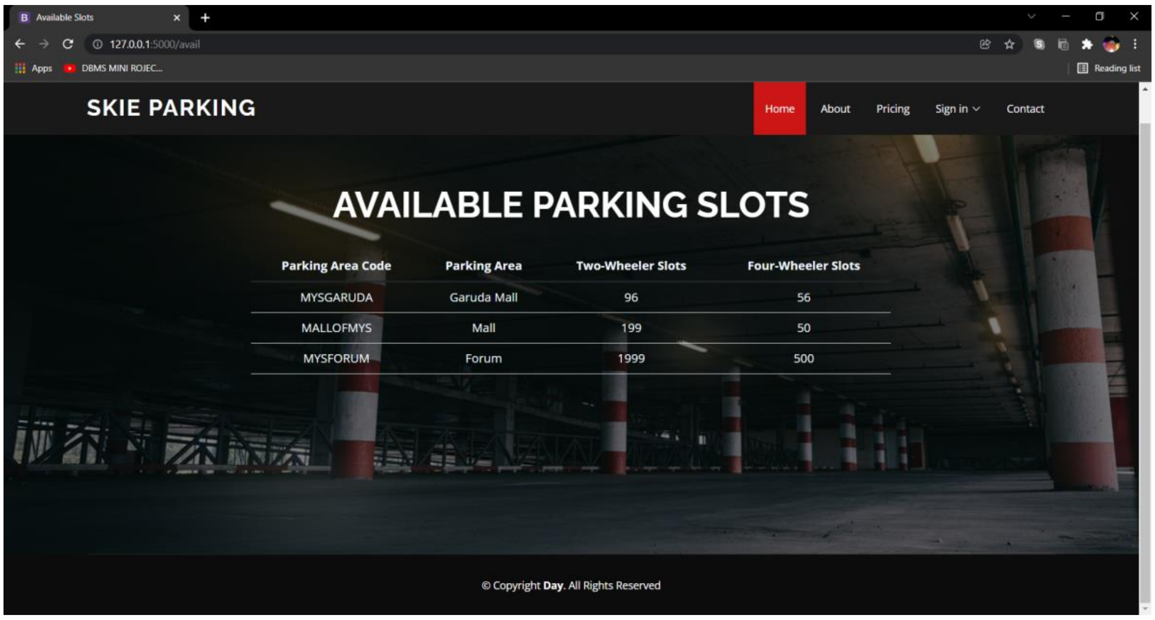
Available Slots

Parking areas with the number of two-wheeler and four-wheeler slots in the respective parking area.

Figure 6.9 User Dashboard – Available Slots

6.7.1 Available Slots

This page can be viewed by the user without logging in.



AVAILABLE PARKING SLOTS

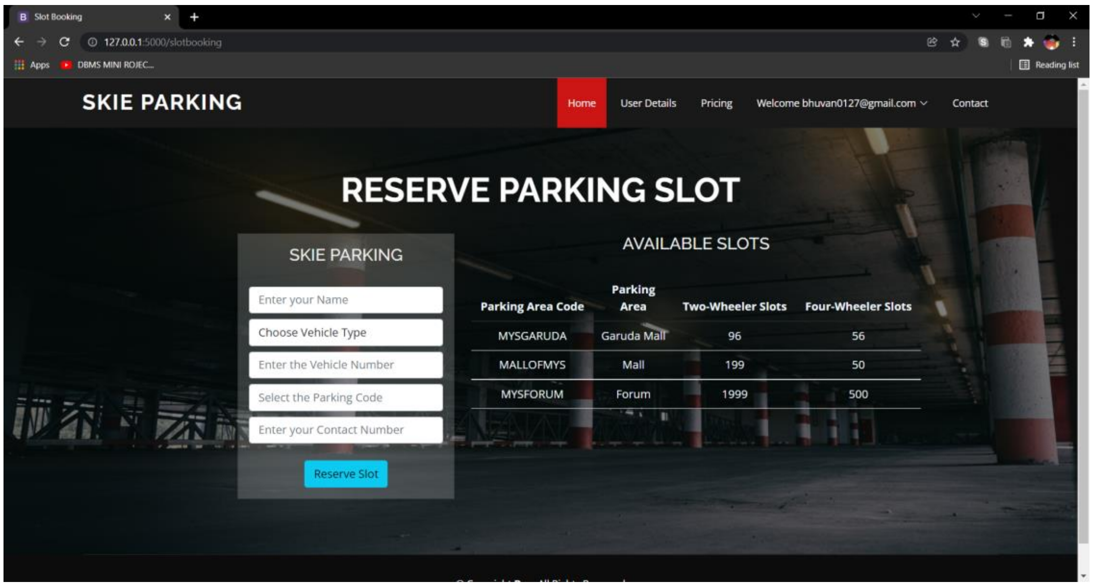
Parking Area Code	Parking Area	Two-Wheeler Slots	Four-Wheeler Slots
MYSGARUDA	Garuda Mall	96	56
MALLOFMYS	Mall	199	50
MYSFORUM	Forum	1999	500

© Copyright **Day**. All Rights Reserved

Figure 6.10 User Dashboard – Available Slots (Without Login)

6.7.1 Reserve slot

The user can reserve a parking slot once he logs in with the correct email and password.



RESERVE PARKING SLOT

SKIE PARKING

Enter your Name

Choose Vehicle Type

Enter the Vehicle Number

Select the Parking Code

Enter your Contact Number

Reserve Slot

AVAILABLE SLOTS

Parking Area Code	Parking Area	Two-Wheeler Slots	Four-Wheeler Slots
MYSGARUDA	Garuda Mall	96	56
MALLOFMYS	Mall	199	50
MYSFORUM	Forum	1999	500

Figure 6.11 Reserve slot

Conclusion and Future Enhancements

In this project, we have designed and achieved a free parking slot management. We have implemented admin login, parking area manager login, user login, slot booking, parking slot enquiry in this system. Business process design and database design is the focus of this system which are clearly and effectively designed by the business process diagrams and database ER diagram. In this project, we have stored all the information about the parking lots, parking slots, user details, reservation details and parking area manager details. Local drivers have an advantage since they are aware of traffic patterns, parking regulations, construction concerns, meter costs, and other factors. Thankfully, parking-related tension can be eased with the use of mobile apps. We have defined the problem on which we worked on, understood the problem domain and produced a model of the system, which describes operations that can be performed on the system. We designed user interface and security issues related to the system. The efficiency of reservation is improved, manual reservation errors is reduced, the management of free parking slots and slot reservation is facilitated. We have considered the most important requirements only; many more features and details can be added to our project in order to obtain even more user-friendly applications.

Drawback:

Drawback is we need to implement a sensor or any physical devices to get the data of number of free slots in different places. To overcome the drawback, we can use a person to get a data which also helps in employment.

Future Enhancements:

This project can be implemented in a wide area as there are no such websites for reserving a parking slot. In the future, the drawback mentioned above can be overcome with an appropriate algorithm. Here, a user can reserve a parking slot for a whole day. This feature can be updated such that the user can reserve according to his time. A billing page can be added so that the user can view the amount payable.

References

- The Database Book: Principles and Practice using MySQL, NarainGehani, Universities Press(India) Private Limited 2008.

- Database Management Systems, Raghu Ramakrishnan and Johannes Gehrke McGrawHill, 3rd Edition, 2003.

- W3schools: https://www.w3schools.com/

- YouTube: https://www.youtube.com/watch?v=7Ge8JQcluoI&t=6639s

- Udemy