

## **ONLINE RESTAURANT TABLE RESERVATION**

**Under Guidance Of  
Mr.JoyjitGuhaBiswas  
Subject Matter Expert(PYTHON)**

Ardent Computech Pvt Ltd(An ISO 9001:2008 Certified) Module 132 SDS Building Salt Lake Sector - 5 , Kolkata - 700091

A

Project Report Submitted In Partial Fulfillment Of The Requirements  
Partial Fulfillment Of The Requirements For The Award Of the Bachelor of Technology  
**Project Carried Out At**



**Ardent Computech Pvt Ltd(An ISO 9001:2008 Certified)**  
Module 132 SDS Building Salt Lake Sector - 5 , Kolkata - 700 091

### **SUBMITTED BY:**

**MOHIT VORA  
YOGESH KUMARSHAW  
RISHAP KUMAR  
SHIVAM GUPTA  
NABIL MALLICK**

### **DEPARTMENT OF BACHELOR OF COMPUTER APPLICATION(BCA)**

**THE CALCUTTA ANGLO GUJARATI COLLEGE  
KOLKATA - 700 001 WEST  
BENGAL | INDIA |**

*(Note: All entries of the proforma of approval should be filled up with appropriate and complete information of approval in any respect will be summarily rejected.)*

1. Name of the Student With Group : MOHIT VORA  
YOGESH KUMAR SHAW  
RISHAP KUMAR  
SHIVAM GUPTA  
NABIL MALLICK

2. Title of the Project : ONLINE RESTAURANT TABLE RESERVATION

3. Name and Address of the Guide: Mr. Joyjit Guha Biswas

Sr. Subject Matter Expert & Technical Head(PYTHON) Ardent Computech Pvt Ltd(An ISO 9001:2008 Certified)  
Module 132 SDS Building Salt Lake Sector - 5 , Kolkata - 700 091

4. Educational Qualification of the Guide : Ph.d\* M.tech\* B.E\*/B.Tech \* MCA\* M.Sc\*

5. Working and Teaching experience of the Guide : ..... years

6. Software used in the Project :  
a. Python 3.8  
b. Xampp Control Panel  
c. Notepad ++

Signature of the Student

Date:

For Office Use Only

Signature of the Guide

Date:

Name: Mr. Joyjit Guha Biswas

Subject Matter Expert. Signature,  
Designation, Stamp of the Project  
Proposal Evaluator

APPROVED

NOT APPROVED

**PROJECT RESPONSIBILITY FORM**  
**ONLINE RESTAURANT TABLE RESERVATION**

<b>Sl.No.</b>	<b>Name of the Members</b>	<b>Responsibility</b>
1.	<b>MOHIT VORA</b>	<b>PROJECT LEADER AND CODING</b>
2.	<b>YOGESH KUMAR SHAW</b>	<b>DESIGNING &amp; CODING</b>
3.	<b>RISHAP KUMAR</b>	<b>DOCUMENTATION</b>
4.	<b>SHIVAM GUPTA</b>	<b>DOCUMENTATION</b>
5.	<b>NABIL MALLICK</b>	<b>DOCUMENTATION</b>

**SELF CERTIFICATE**

This is to certify that the dissertation/project proposal entitled "**ONLINE RESTAURANT TABLE RESERVATION**" is done by us, is an Authentic work carried out for the partial fulfillment of the requirements for the award of the certificate of **Bachelor of Technology** under the guidance of **Mr. Joyjit Guha Biswas**. The matter embodied in this project work has not been submitted earlier for award of any certificate to the best of our knowledge and belief.

**NAME OF THE STUDENTS**

- MOHIT VORA
- YOGESH KUMAR SHAW
- RISHAP KUMAR
- SHIVAM GUPTA
- NABIL MALLICK

### **CERTIFICATE BY GUIDE**

This is to certify that this project entitled "**Online Restaurant Table Reservation**" submitted in partial fulfillment of the certificate of Bachelor of Computer Application through **Ardent Computech Pvt Ltd**, done by the Group Members

- MOHIT VORA
- YOGESH KUMAR SHAW
- RISHAP KUMAR
- SHIVAM GUPTA
- NABIL MALLICK

is an authentic work carried out under my guidance & best of our knowledge and belief.

Signature of the students

Date:

Signature of the Guide

Date:

### **CERTIFICATE OF APPROVAL**

This is to certify that this proposal of Minor project, entitled "**Online Restaurant Table Reservation**" is a record of bona-fide work, carried out by: 1. Mohit Vora, 2. Yogesh Kumar Shaw, 3. Rishabh Kumar, 4. Shivam Gupta, 5. Nabil Mallick under my supervision and guidance through the Ardent Computech Pvt Ltd. In my opinion, the report in its present form is in partial fulfillment of all the requirements, as specified by The Calcutta Anglo-Gujarati College as per regulations of the **Ardent®**. In fact, it has attained the standard, necessary for submission. To the best of my knowledge, the results embodied in this report, are original in nature and worthy of incorporation in the present version of the report for Bachelor of Technology.

---

#### **GUIDE/SUPERVISOR**

**Mr. Joyjit Guha Biswas**

Subject Matter Expert & Technical Head (PYTHON) Ardent  
Computech Pvt Ltd (An ISO 9001:2008 Certified)  
Module 132 SDS Building Salt Lake Sector - 5 , Kolkata - 700 091

#### **External Examiner(s)**

**Head of the Department** Department of  
Computer Science **Kolkata, W.B**  
(Affiliated to WBUT, WB)

## TABLE OF CONTENTS

<b>SL.NO.</b>	<b>NAME OF THE TOPIC</b>	<b>PAGE NO.</b>
<b>1.</b>	<b>COMPANY PROFILE</b>	<b>8</b>
<b>2.</b>	<b>INTRODUCTION</b>	<b>10</b>
	Objective	11
	Scope	12
<b>3.</b>	<b>SYSTEM ANALYSIS</b>	<b>13</b>
	Identification of Need	13
	Feasibility Study	14
	Workflow	15
	Study of the Systems	18
	Input & Output	19
	Software Requirement Specifications(SRS)	20
	Software Engineering Paradigm Applied	21
<b>4.</b>	<b>SYSTEM DESIGN</b>	<b>23</b>
	Data Flow Diagram	23
	Data Sequence	29
	Entity Relationship Diagram	30
	Modularization Diagram	34
	Database Design	36
	Data Dictionary	
<b>5.</b>	<b>OUTPUT SCREEN</b>	
	UserInterfaceDesign Snapshots	
<b>6.</b>	<b>IMPLEMENTING &amp; TESTING</b>	
	Introduction	

Objective of Testing Test Cases  
White Box Testing Black  
Box Testing Output Testing  
Goal Of Testing  
Integration Test Reports

**7. GANTT CHART**

**8. SYSTEM SECURITY MEASURES**

Database Security Measures System  
Security Measures Limitations

**9. CONCLUSION**

**10. FUTURESCOPE & FURTHER  
ENHANCEMENTS**

**11. BIBLIOGRAPHY/REFERENCES**

#### **ARDENT COMPUTECH PVT.LTD.**

Ardent Computech Private Limited is an ISO 9001-2008 certified Software Development Company in India. It has been operating independently since 2003. It was recently merged with **ARDENT TECHNOLOGIES**.

---

#### **Ardent Technologies**

**ARDENT TECHNOLOGIES** is a Company successfully providing its services currently in the UK, USA, Canada and India. The core line of activity at **ARDENT TECHNOLOGIES** is to develop customized application software covering the entire responsibility of performing the initial system study, design, development, implementation and training. It also deals with consultancy services and Electronic Security systems. Its primary clientele includes educational institutes, entertainment industries, resorts, theme parks, service industry, telecom operators, media and other business houses working in various capacities.

---

#### **Ardent Collaborations**

**ARDENT COLLABORATIONS**, the Research Training and Development Department of **ARDENT COMPUTECH PVT LTD** is a professional training Company offering IT enabled services & industrial trainings for B-Tech, MCA, BCA, MSc and MBA freshers and experienced developers/programmers in various platforms. Summer Training / Winter Training / Industrial training will be provided for the students of B.TECH, M.TECH, MBA and MCA only. Deserving candidates may be awarded stipends, scholarships and other benefits, depending on their performance and recommendations of the mentors.

---

#### **Associations**

Ardent is an ISO 9001:2008 company. It is affiliated to National Council of Vocational Training (NCVT), Directorate General of Employment & Training (DGET), Ministry of Labor & Employment, and the Government of India.

**ORTR**  
**(ONLINE RESTAURANT  
TABLE RESERVATION)**

## INTRODUCTION

A **table reservation** is an arrangement made in advance to have a table available at a later time. While most restaurants in the vast majority of the world do not require a reservation, and some do not have a policy or simply any channel for making one, so-called higher-end restaurants mainly in overcrowded cities often require a reservation, and some may have tables booked for so-called higher-end restaurants mainly in overcrowded cities often require a reservation, and some may have tables booked for reservation, and some may have tables booked for weeks in advance. At particularly exclusive reservation systems evolved from the practice of arranging a table at a restaurant. Today, at such venues, observes Joy Smith, author of *Kitchen Afloat: Galley Management and Meal Preparation* (2002): "It's always smart to inquire about a restaurant's reservation policy. Some will only reserve for large parties of six or more".

In recent times, many restaurants have replaced the traditional pencil and notebook with an online reservation system. Some websites exist which provide this service for multiple venues, such as Tock, Bookatable, Chope, EZTABLE, Dimmi, Killer Rezzy (which also offers "rogue" reservations, which members can sell), OpenTable, Resy, Yelp Reservations (formerly known as SeatMe), Shout (which allows users to either make a reservation or put the one they have up for sale), Zomato, ResDiary.

Reservations for late dining times may prove problematic, as a restaurant may have a backlog which will require the reservation-holders to wait beyond their stated arrival time. In addition, diners with a late reservation face a higher chance that the restaurant will run out of necessary ingredients for a particularly popular dish. Reservation once it is known one will not use it.

Most restaurants do not charge a customer who fails to honor their reservations, and courts have tended not to impose substantial penalties on restaurants that fail to honor reservations. Nonetheless, it is generally considered polite to call and cancel a

## **Objectives**

Restaurant reservation system helps to reduce one of the direst problems of the industry – no-show clients – empty seating is always a sorry sight, especially for eateries operating on small profit margins. It can also be harrowing for these businesses to turn away customers because of the wrong reservations or unavailability of tables.

That's why more than 80% of the restaurants are turning to technology like online ordering and reservation software to streamline their business processes – 95% of these restaurants have reported growth by employing this technology.

Restaurant reservation systems help manage the constant influx of reservations and customers – allowing the customers to book their tables remotely so managers can schedule resources according to the number of bookings.

Customers also want a more convenient dining experience, and restaurants want to deliver the same to boost their business. A powerful table booking system for restaurants can help them do just that.

## **Scope**

Traditionally, restaurants have managed their reservation systems with a reservation book, which means they received the reservations via telephone calls and wrote them down in a book. Nowadays, as a consequence of the massive use of the Internet and its benefits, experts have seen the opportunity and great added value of creating online reservation systems, and already many restaurants have replaced the traditional format with these new systems.

An important advantage of online reservation systems is the flexibility they offer when making a reservation. When reservations are managed in the traditional way, patrons will only be able to call a restaurant to make a reservation during operational hours. On the contrary, when reservations are managed through an online reservation system, customers will be able to make their reservation at any time and from any place they choose. In general, patrons will have a better experience when making an online reservation, because it will be a quick process, the service will be available 24/7, and the system will provide all the necessary information in order to make the desired reservation with tranquility.

Restaurants will experience a great number of benefits when using an online reservation system. Some of these benefits translate into a decline in incoming phone calls, a better control of the capacity of the restaurant and the number of reservations one will be able to accept, and a number of handy statistics and reports that will help to analyze the business in interesting ways.

These benefits arise from a wide range of management tools provided by online reservation systems, like operational reports, floor management software, customer reservation histories, and customer databases that include customer data and preferences, and grow with each new table booking. Restaurants will also be able to track cancellations, and manage walk-in and waitlists in a better way, eliminate overbookings, and create target email and postal mailings with the information from the customer database. Some online reservation systems include integrated email marketing tools.

## **SYSTEM ANALYSIS**

### **Identification Of Need**

The traditional examination system has been in place for a long time. Over the years, tweaks in exam duration, question format, evaluation methods, and feedback have evolved, but generally speaking this pen and paper model has persisted over decades of education. So why the change to an online examination system now?

Simply put, with innovations in technology, an increasingly globalized world, and data-driven insights into teaching and learning methodologies, an online examination system might provide a better perspective into the scholarly journey of a student. Examinations are not just a part and parcel of student life. They are an evaluative tool that allows students, teachers, and parents to track and measure the student's scholarly progress. It also serves as a concrete metric on which to base learning and teaching decisions for the future.

Unlike a traditional examination system, an online examination system is more enhanced in numerous ways. From easier setup to remote access, to faster evaluations, an online exam system is better at fulfilling its primary role of

being an evaluative tool.

### **Feasibility Study**

**ECONOMIC FEASIBILITY :** Economic analysis is most frequently used for evaluation of the effectiveness of the system. More commonly known as cost/benefit analysis the procedure is to determine the benefit and saving that are expected from a system and compare them with costs, decisions are made to design and implement the system .This part of feasibility study gives the top management the economic justification for the new system. This is an important input to the management, because very often the top management doesn't like to get confused by the various technicalities that are bound to be associated with a project of this kind. A simple economic analysis that gives the actual comparison of costs and benefits is much more meaningful in such cases. In the system, the organization is most satisfied by economic feasibility .Because, if the organization implements this system, it need not require any additional hardware resources as well as it will be saving a lot of time.

**TECHNICAL FEASIBILITY :** Technical feasibility centers on the existing manual system of the test management process and to what extent it can support the system According to feasibility analysis procedure the technical feasibility of the system is analyzed and the technical requirements such as software facilities, procedure, inputs are identified. It is also one of the important phases of the system development activities. The system offers greater levels of user friendliness combined with greater processing speed. Therefore, the cost of maintenance can be reduced. Since, processing speed is very high and the work is reduced from the maintenance point of view, management is convinced that the project is operationally feasible.

**BEHAVIOURAL FEASIBILITY :** People are inherently resistant to change and computers have been known to facilitate changes. An estimate should be made of how strong the user is likely to move towards the development of computerized systems. These are various levels of users in order to ensure proper authentication and authorization and security of sensitive data of the organization.

## **Workflow**

This Document plays a vital role in the development life cycle (SDLC) as it describes the complete requirement of the system. It is meant for use by the developers and will be the basics during the testing phase. Any changes made to the requirements in the future will have to go through a formal change approval process.

The Waterfall Model was the first Process Model to be introduced. It is also referred to as a linear-sequential life cycle model. It is very simple to understand and use. In a waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases.

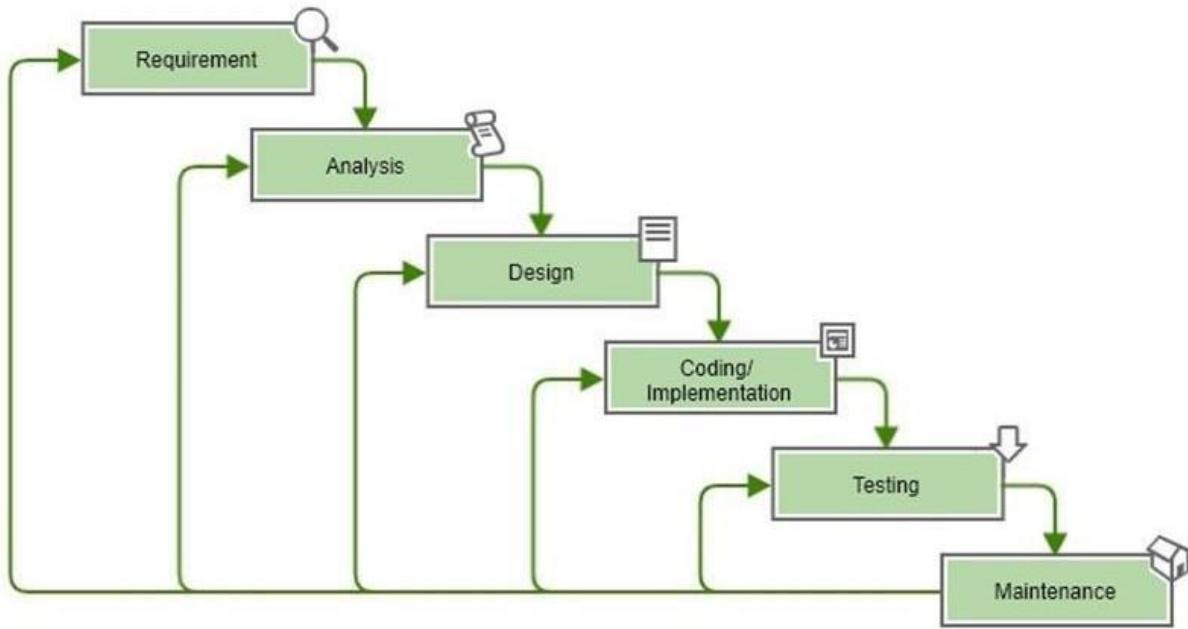
The Waterfall model is the earliest SDLC approach that was used for software development .

The waterfall Model illustrates the software development process in a linear sequential flow; hence it is also referred to as a linear-sequential life cycle model. This means that any phase in the development process begins only if the previous phase is complete. In waterfall model phases do not overlap.

### **Waterfall Model design:-**

Waterfall approach was the first SDLC Model to be used widely in Software Engineering to ensure success of the project. In "The Waterfall" approach, the whole process of software development is divided into separate phases. In the Waterfall model, typically, the outcome of one phase acts as the input for the next phase sequentially.

Following is a diagrammatic representation of different phases of the waterfall model.



The sequential phases in Waterfall model are:

- **Requirement Gathering and analysis:** All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification doc.
- **System Design:** The requirement specifications from the first phase are studied in this phase and system design is prepared. System Design helps in specifying hardware and system requirements and also helps in defining overall system architecture.
- **Implementation:** With inputs from system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality which is referred to as Unit Testing.
- **Integration and Testing:** All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.
- **Deployment of system:** Once the functional and non functional testing is done, the product is deployed in the customer environment or released into the market.
- **Maintenance:** There are some issues which come up in the client environment. To fix those issues patches are released. Also to enhance the

product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

All these phases are cascaded to each other in which progress is seen as flowing steadily downwards (like a waterfall) through the phases. The next phase is started only after the defined set of goals are achieved for the previous phase and it is signed off, so the name "Waterfall Model". These model phases do not overlap.

#### **Waterfall Model Application:-**

Every software developed is different and requires a suitable SDLC approach to be followed based on the internal and external factors. Some situations where the use of Waterfall model is most appropriate are:

- Requirements are very well documented, clear and fixed.
- Product definition is stable.
- Technology is understood and is not dynamic.
- There are no ambiguous requirements.
- Ample resources with required expertise are available to support the product.
- The project is short.

The advantage of waterfall development is that it allows for departmentalization and control. A schedule can be set with deadlines for each stage of development and a product can proceed through the development process model phases one by one. Development moves from concept, through design, implementation, testing, installation, troubleshooting, and ends up at operation and maintenance. Each phase of development proceeds in strict order.

## **Study Of The System**

### **Modules:-**

The modules used in this software are as follows:

- **Login:** This module is for ADMIN and COUNCILLOR. ADMIN has the authority to Insert, Update and Delete councillor, branch, courses, faculty, payment details, batch, installment. ADMINISTRATOR has the authority to insert, Update,Delete student details.
- **Home:** This page contains an overview of highlights for other pages.
- **About Us:** This page contains information about SMS.
- **Faculty:** This page contains the information about the faculties for specified courses.
- **Branch:** This page contains the information of each branch in detail.
- **Courses:** This page contains the updated information about the courses that are available .
- **Installment:** This page contains the information about the installment part by part.
- **Student:** This page contains minute details of the each and every details of student from the time of admission till time of certification .

### **Input And Output**

The main inputs ,outputs and the major function the details are :

#### **INPUT**

- Councillor can login using op-id and password.
- Admin can login using admin-id and password.
- Admin insert and modify the details councillor, branch, course, installment, and payment details.
- Councillors can make admission of a student by registering student details.
- Councillor can modify student details searching them by id and Name.

#### **OUTPUT**

- Operators can view the details of the faculty, branch, course, installment , batch.
- Admin can view the details of councillor,branch,course, installment,student and payment details.

### **Software Requirements Specifications(SRS)**

Software Requirements Specification provides an overview of the entire project. It is a description of a software system to be developed, laying out functional and nonfunctional requirements. The software requirements specification document enlists enough and necessary requirements that are required for the project development. To derive the requirements we need to have a clear and thorough understanding of the project to be developed. This is prepared after the detailed communication with the project team and the customer.

The developer is responsible for:-

- ✓ Developing the system, which meets the SRS and solving all the requirements of the system?
- ✓ Demonstrating the system and installing the system at client's location after acceptance testing is successful.
- ✓ Submitting the required user manual describing the system interfaces to work on it and also the documents of the system.
- ✓ Conducting any user training that might be needed for using the system.
- ✓ Maintain the system for a period of one year after installation.

### **Hardware Requirements:**

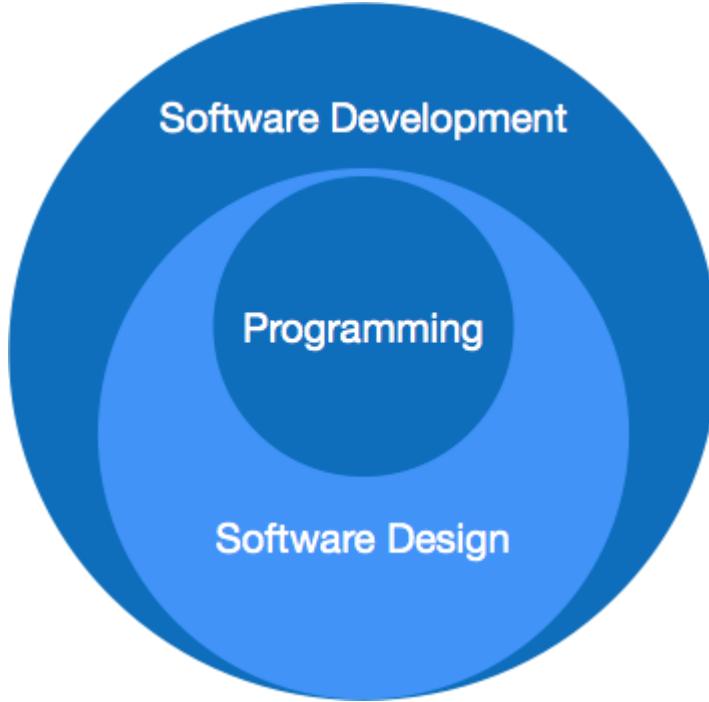
- Computer that has a 1.6GHz or faster processor.
- 1 GB (32 Bit) or 2 GB (64 Bit) RAM (Add 512 MB if running in a virtual machine).
- HDD 20 GB Hard Disk Space and Above Hardware Requirements.
- 5400 RPM hard disk drive.

### **Software Requirements:**

- WINDOWS OS (XP/2000/2003/2010 or 2000 Server/Vista or 7)
- Python 3.8
- Notepad/Notepad++
- Browser :- Google Chrome,Mozilla Firefox

## **SOFTWARE ENGINEERING PARADIGM APPLIED**

Software paradigms refer to the methods and steps, which are taken while designing the software. There are many methods proposed and are in work today, but we need to see where in the software engineering these paradigms stand. These can be combined into various categories, though each of them is contained in one another.



Programming paradigm is a subset of Software design paradigm which is further a subset of Software development paradigm. There are two levels of reliability. The first is meeting the right requirement. A careful and thorough systems study is needed to satisfy this aspect of reliability. The second level of systems reliability involves the actual working delivered to the user. At this level, the systems reliability is interwoven with software engineering and development. There are three approaches to reliability.

- 1. Error avoidance:** Prevents errors from occurring in software.
- 2. Error detection and correction:** In this approach errors are recognized whenever they are encountered and correcting the error by the effect of error on the system does not fail.

**3. Error tolerance:** In this approach errors are recognized whenever they occur, but enable the system to keep running through degraded performance or applying values that instruct the system to continue the process.

**Maintenance:**

The key to reducing need for maintenance, while working, if possible to do essential tasks.

1. More accurately defining user requirements during system development.
2. Assembling better systems documents.
3. Using some effective methods for designing, processing, and login and communicating information with project teammembers.
4. Making better use of existing tools and techniques.

## SYSTEM DESIGN

### Data Flow Diagram(DFD)

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modelling its process aspects. A DFD is often used as a preliminary step to create an overview of the system, which can later be elaborated.

DFDs can also be used for the visualization of data processing (structured design). A DFD shows what kind of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of the process or information about whether processes will operate in sequence or in parallel (which is shown on a flowchart).

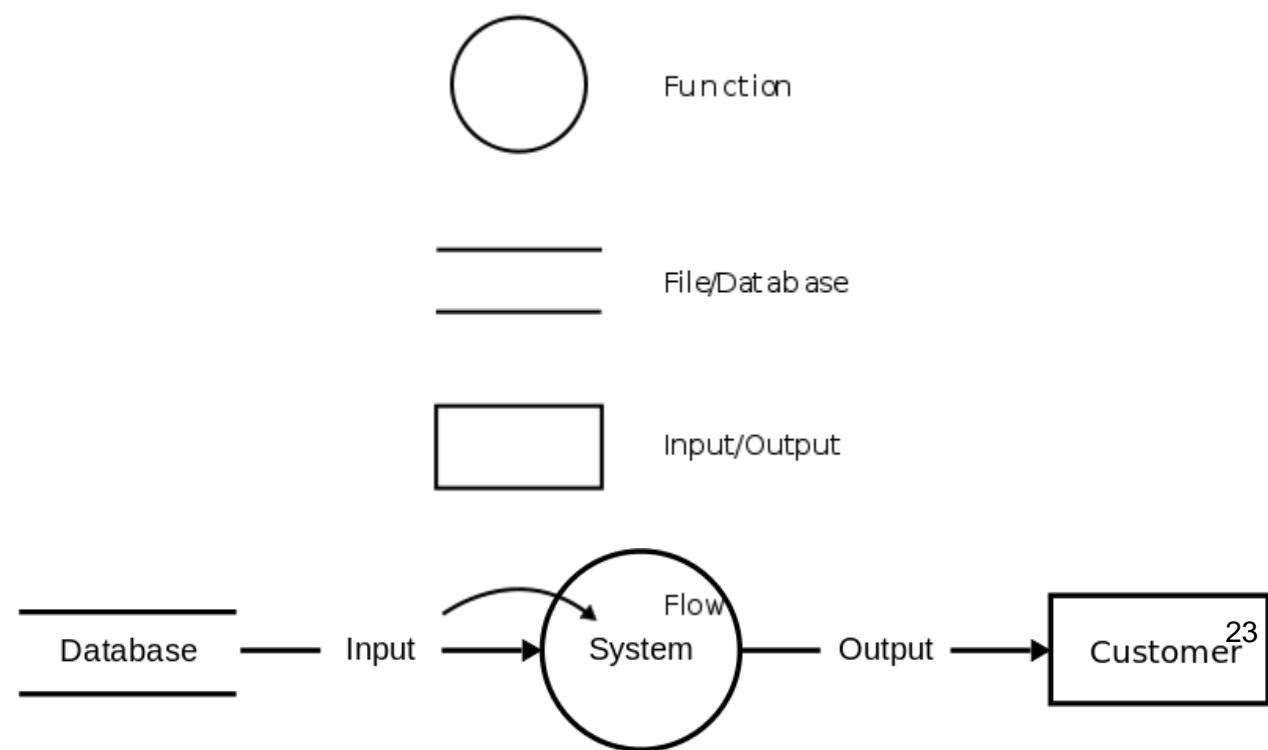
This context-level DFD is next "exploded", to produce a Level 1 DFD that shows some of the details of the system being modeled. The Level 1 DFD shows how the system is divided into subsystems (processes), each of which deals with one or more of the data flows to or from an external agent, and which together provide all of the functionality of the system as a whole. It also identifies internal data stores that must be present in order for the system to do its job, and shows the flow of data between the various parts of the system.

Data flow diagrams are one of the three essential perspectives of the structured-systems analysis and design method SSADM. The sponsor of a project and the end users will need to be briefed and consulted throughout all stages of a system's evolution. With a data flow diagram, users are able to visualize how the system will operate, what the system will accomplish, and how the system will be implemented. The old system's data flow diagrams can be drawn up and compared with the new system's data flow diagrams to draw comparisons to implement a more efficient system. Data flow diagrams can be used to provide the end user with a physical idea of where the data they input ultimately has an effect upon the structure of the whole system from order to dispatch to report. How any system is developed can be determined through a data flow diagram model. In the course of developing a set of leveled dataflow diagrams the analyst/designer is forced to address how the system may be decomposed into component subsystems, and to

identify the transaction data in the data model. Data flow diagrams can be used in both Analysis and Design phases of the SDLC.

There are different notations to draw dataflow diagrams. Defining different visual representations for processes, data stores, data flow, and external entities.

#### DFD Notations

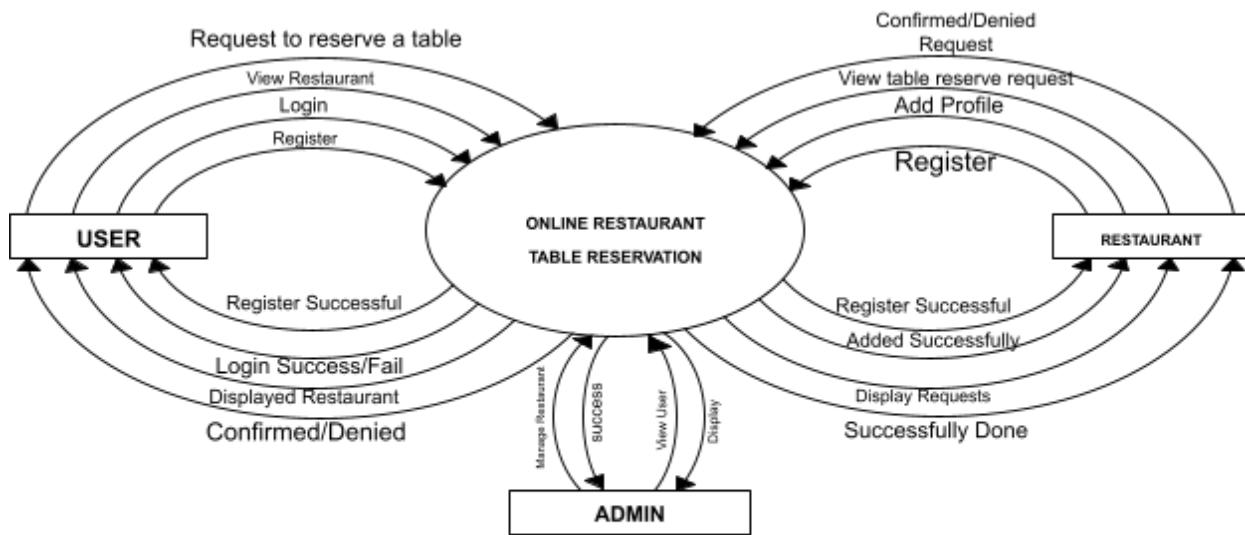


Steps to Construct Data FlowDiagram:-

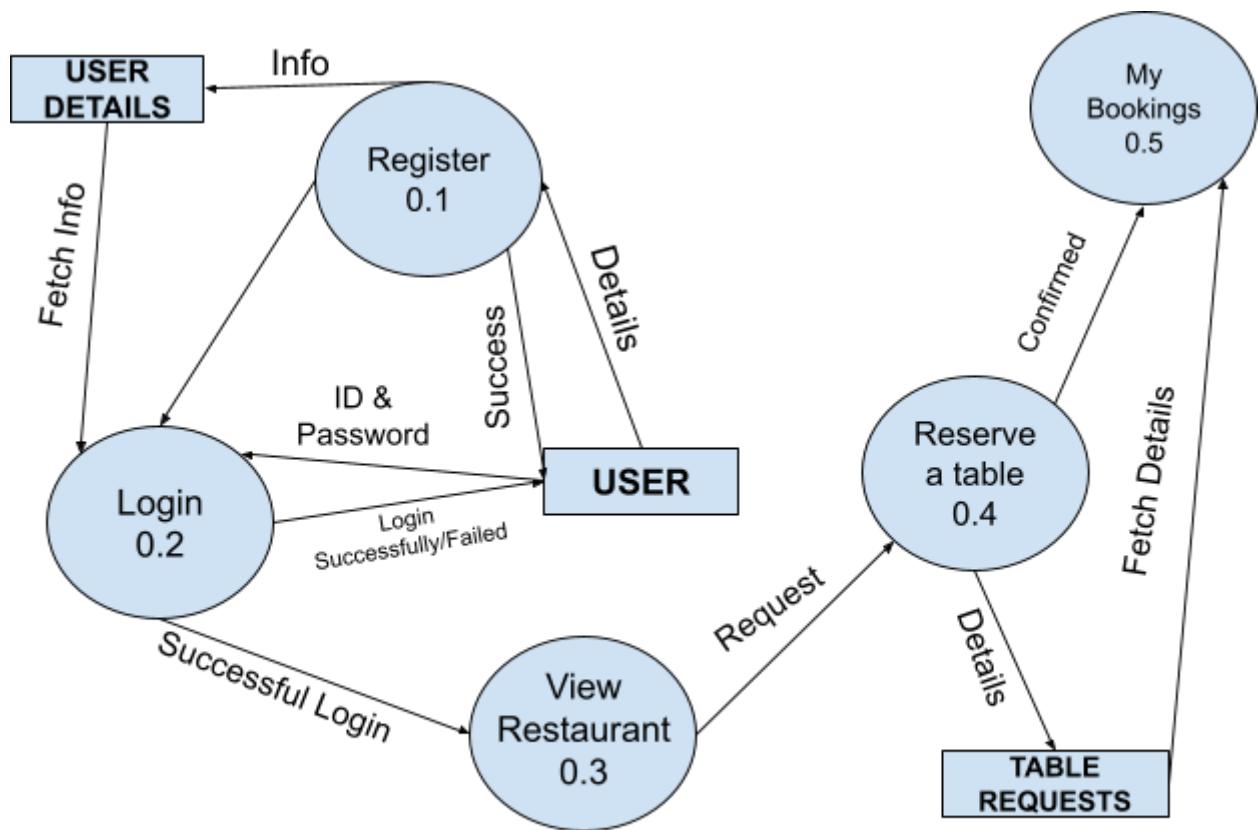
Four Steps are generally used to construct a DFD.

- Process should be named and referred for easy reference. Each name should be representative of the reference.
- The destination of flow is from top to bottom and from left to right.
- When a process is distributed into lower level details they are numbered.
- The names of data stores, sources and destinations are written in capital letters. Rules for constructing a Data Flow Diagram:-

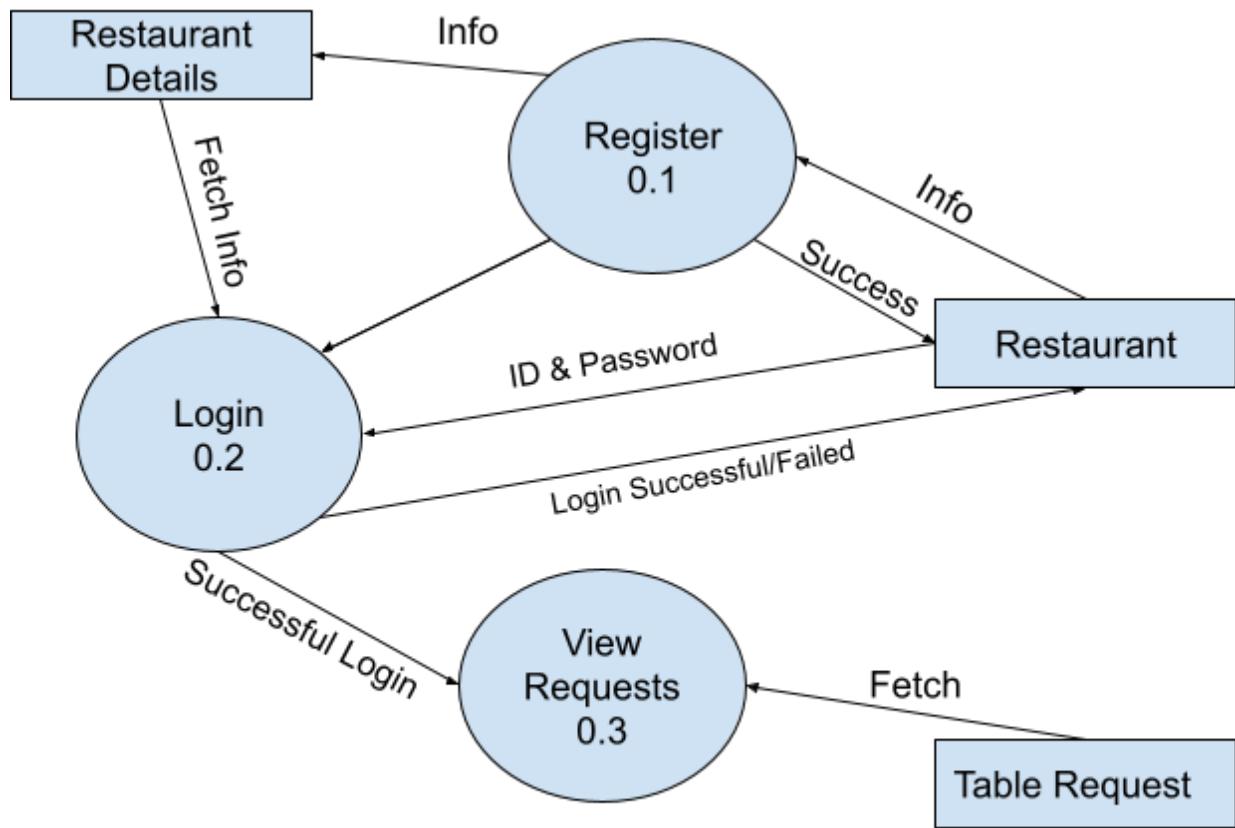
- Arrows should not cross eachother.
- Squares, Circles, Files must bear a name.
- Decomposed data flow squares and circles can have the same names. Draw all data flow around the outside of the diagram.



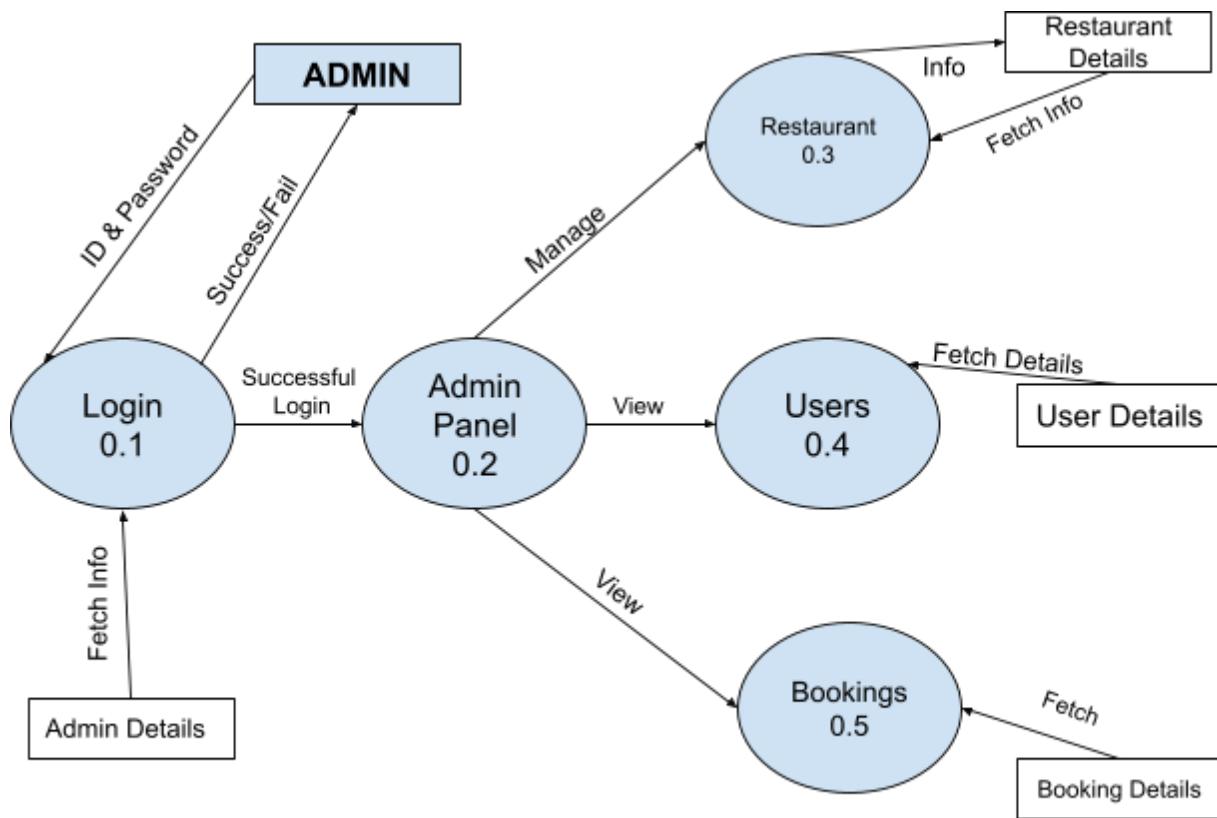
## LEVEL 0 DFD



**LEVEL 1 USER DFD**



**LEVEL 1 RESTAURANT DFD**



**LEVEL 1 ADMIN DFD**

## Sequence Diagram

A Sequence diagram is an interaction diagram that shows how processes operate with one another and what is their order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called **event diagrams** or **event scenarios**.

A sequence diagram shows, as parallel vertical lines (*lifelines*), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical Manner.

**Sequence diagram** is the most common kind of **interaction diagram**, which focuses on the message interchange between a number of **lifelines**.

Sequence diagram describes an interaction by focusing on the sequence of messages that are exchanged, along with their corresponding occurrence specifications on the lifelines.

The following nodes and edges are typically drawn in a UML sequence diagram: **lifeline**, **execution-specification**, **message**, **fragment**, **interaction**, **state invariant**, **continuation**, **destruction occurrence**.

### **Entity Relationship Diagram**

In software engineering, an **entity–relationship model(ER model)** is a data model for describing the data or information aspects of a business domain or its process requirements, in an abstract way that lends itself to ultimately being implemented in a database such as a relational. The main components of ER models are entities (things) and the relationships that can exist among them.

An entity–relationship model is the result of using a systematic process to describe and define a subject area of business data. It does not define business processes; only visualize business data. The data is represented as components (entities) that are linked with each other by relationships that express the dependencies and requirements between them, such as: one building may be divided into zero or more apartments, but one apartment can only be located in one building. Entities may have various properties (attributes) that characterize them. Diagrams created to represent these entities, attributes, and relationships graphically are called entity–relationship diagrams.

An ER model is typically implemented as a database. In the case of a relational database, which stores data in tables, every row of each table represents one instance of an entity. Some data fields in these tables point to indexes in other tables; such pointers are the physical implementation of the relationships.

The three schema approach to software engineering uses three levels of ER models that may be developed.

#### **Conceptual data model**

The conceptual ER model normally defines master reference data entities that are commonly used by the organization. Developing an enterprise-wide conceptual ER model is useful to support documenting the data architecture for an organization. A conceptual ER model may be used as the foundation for one or more logical data models . The purpose of the conceptual ER model is then to establish structural metadata commonality for the master data entities between the set of logical ER models. The conceptual data model may be used to form

commonality relationships between ER models as a basis for data model integration.

### **Logical data model**

The logical ER model contains more detail than the conceptual ER model. In addition to master data entities, operational and transactional data entities are now defined. The details of each data entity are developed and the relationships between these data entities are established. The logical ER model is however developed independent of technology into which it can be implemented.

### **Physical data model**

One or more physical ER models may be developed from each logical ER model. The physical ER model is normally developed to be instantiated as a database. Therefore, each physical ER model must contain enough detail to produce a database and each physical ER model is technology dependent since each database management system is somewhat different.

The physical model is normally instantiated in the structural metadata of a database management system as relational database objects such as database tables, database indexes such as unique key indexes, and database constraints such as a foreign key constraint or a commonality constraint. The ER model is also normally used to design modifications to the relational database objects and to maintain the structural metadata of the database.

The first stage of information system design uses these models during the requirements analysis to describe information needs or the type of information that is to be stored in a database. The data modeling technique can be used to describe any ontology (i.e. an overview and classifications of used terms and their relationships) for a certain area of interest. In the case of the design of an information system that is based on a database, the conceptual data model is, at a later stage (usually called logical design), mapped to a logical data model, such as the relational model; this in turn is mapped to a physical model during physical design. Note that sometimes, both of these phases are referred to as "physical design". It is also used in databases management systems.



Two related entities



An entity with an attribute



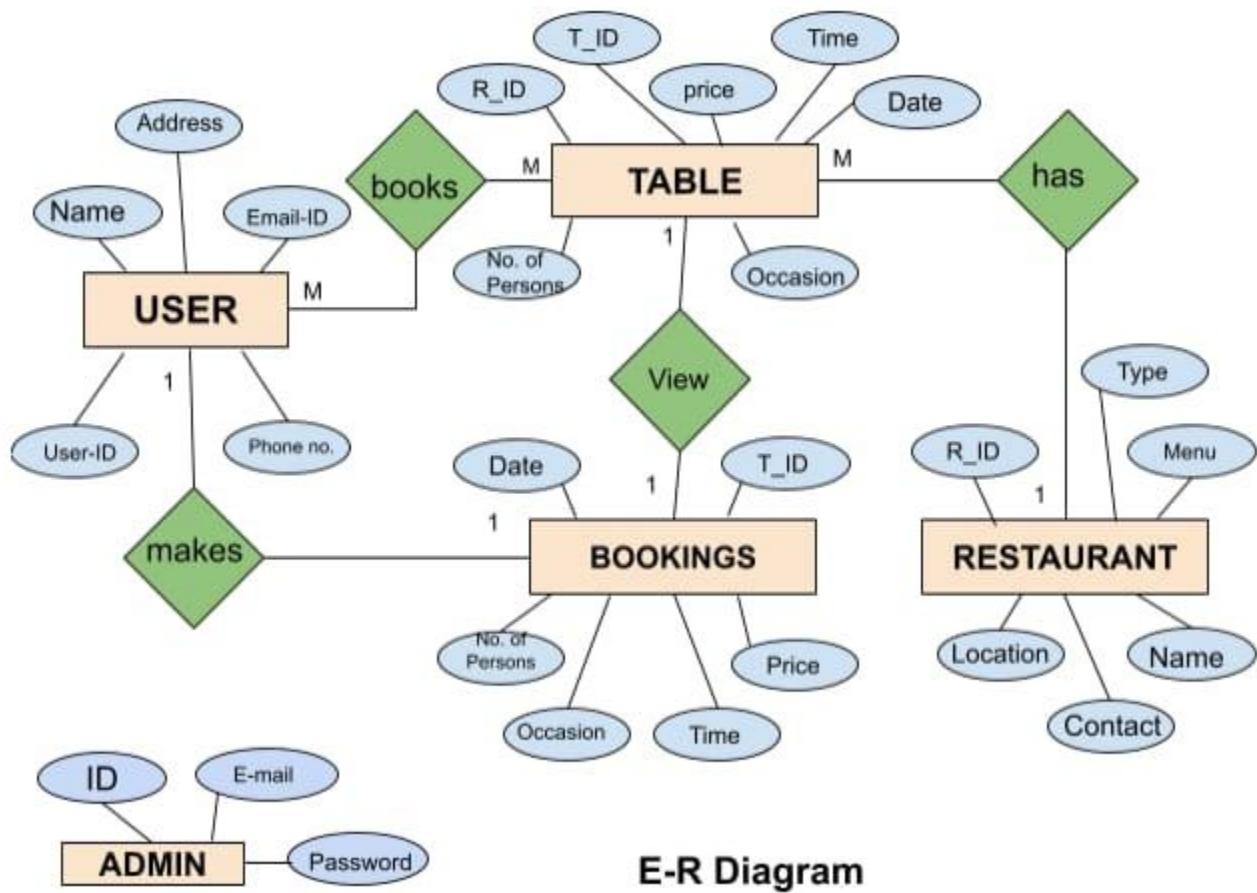
A relationship with an entity



## Entity Relationship Modeling

Cardinality constraints are expressed as follows:

- A double line indicates a participation constraint, totality or subjectivity: all entities in the entity set must participate in at least one relationship in the relationship set.
- An arrow from entity set to relationship set indicates a key constraint, i.e. injectivity: each entity of the entity set can participate in at most one relationship in the relationship set.
- A thick line indicates both, i.e. bijection: each entity in the entity set is involved in exactly one relationship.
- An underlined name of an attribute indicates that it is a key: two different entities or relationships with this attribute always have different values for this attribute.



## **Modularization Details**

As Modularization has gained increasing focus from companies outside its traditional industries of aircraft and automotive, more and more companies turn to it as a strategy and product development tool. I intend to explain the important aspects of modularization and how it should be initiated within a company. After determining the theoretical steps of modularization success described in literature, I intend to conduct a multiple case study of companies who have implemented modularization in order to find how real world modularization was initiated and used to improve the company's competitiveness. By combining theory and practical approach to modularization I will derive a convergence and divergence between theoretical implementation to modularization and real world implementation to modularization. This gives a valuable input for both implantations in companies as well as new aspects to be further.

### **Data Integrity and constraints**

Data integrity is normally enforced in a database system by a series of integrity constraints or rules. Three types of integrity constraints are an inherent part of the relational data model: entity integrity, referential integrity and domain integrity:

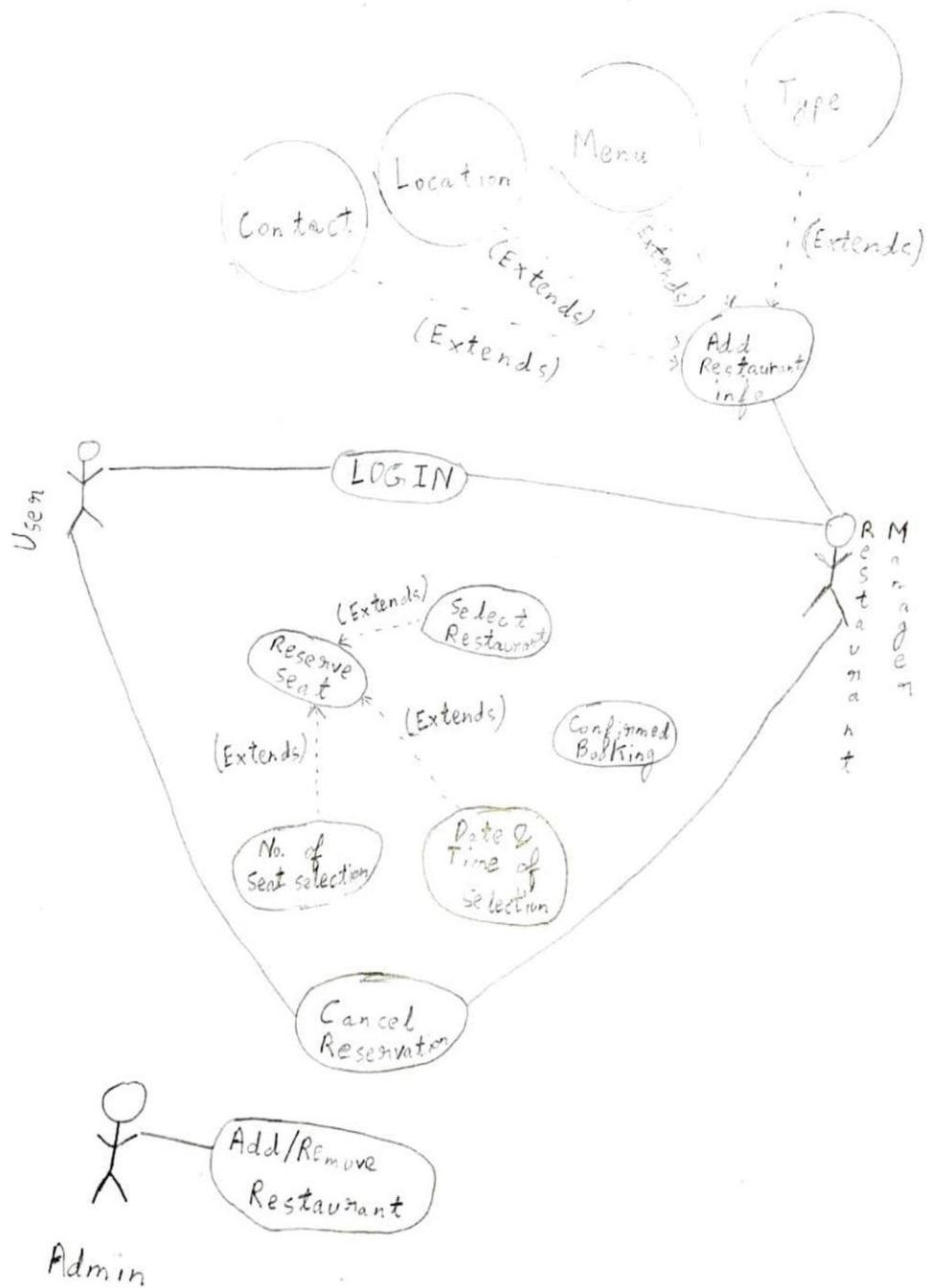
- Entity integrity concerns the concept of a primary key. Entity integrity is an integrity rule which states that every table must have a primary key and that the column or columns chosen to be the primary key should be unique and not null.
- Concerns the concept of a foreign key. The referential integrity rule states that any foreign-key value can only be in one of two states. The usual state of affairs is that the foreign-key value refers to a primary key value of some table in the database.  
Occasionally, and this will depend on the rules of the data owner, a foreign-key value can be null. In this case we are explicitly saying that either there is no relationship between the objects represented in the database or that this relationship is unknown.

- Domain integrity specifies that all columns in a relational database must be declared upon a defined domain. The primary unit of data in the relational data model is the data item. Such data items are said to be non-decomposable or atomic. A domain is a set of values of the same type.

### **Database Design**

A database is an organized mechanism that has capability of storing information through which a user can retrieve stored information in an effective and efficient manner. The data is the purpose of any database and must be protected. The database design is two level processes. In the first step, user requirements are gathered together and a database is designed which will meet these requirements as clearly as possible. This step is called information Level design and it is taken independent of any individual DBMS. In the following snapshots we display the way we have used SQL Server as the back-end RDBMS for our project and the various entities that have been used along with their table definition and table data.

# USE CASE DIAGRAM



## DATA DICTIONARY

### MY BOOKINGS TABLE DATA

SELECT \* FROM `bs`

Profiling (Ed)

	<input type="button" value="1"/>	>	>>	<input type="checkbox"/> Show all	Number of rows:	25	<input type="text"/> Filter rows: Search this table
+ Options							
tid	day	people	time	email	phno	rname	
9	2021-04-15	1	13:58:00.000000	mk12@gmail.com	6788376464		
2	2021-04-16	2	14:00:00.000000	kk21@gmail.com	9831554183		
2	2021-04-17	3	17:05:00.000000	kk21@gmail.com	9831554183		
10	2021-04-19	2	12:30:00.000000	rahul1@gmail.com	9831554184		
3	2021-04-19	24	11:30:00.000000	rahul1@gmail.com	9831554184		
7	2021-04-25	1	14:35:00.000000	rahul1@gmail.com	9831554184		
2	2021-04-21	5	13:25:00.000000	rahul1@gmail.com	9831554184		
11	2021-04-20	3	12:13:00.000000	ykshaw@gmail.com	8013724949	Indian Restaurant	
8	2021-04-20	4	12:34:00.000000	yk333shaw@gmail.com	8013724948	OCTA	
4	2021-04-20	0	12:40:00.000000	y3Shaw@gmail.com	9855555555	Urban Masala	
3	2021-04-20	2	12:40:00.000000	y	9999999999	Friends Of Pho - Asian Fine Dining Club	
9	2021-04-21	5	17:00:00.000000	lon@gmail.com	7891554183	OCTA	
12	2021-04-22	1	20:30:00.000000	mv705@gmail.com	7059965591	India Restaurant	
13	2021-04-22	0000000	12:10:00.000000	mv705@gmail.com	7059965591	India Restaurant	
15	2021-04-25	8888888	12:22:00.000000	mv705@gmail.com	7059965591	India Restaurant	
17	2021-04-30	fafsa	12:32:00.000000	saff	sffafasf	India Restaurant	
9	2021-04-21	5	17:00:00.000000	lon@gmail.com	7891554183	OCTA	
1	2021-04-14	3	13:41:00.000000	table c	nn	Urban Masala	
3	2021-04-20	2	12:40:00.000000	y	9999999999	Friends Of Pho - Asian Fine Dining Club	
9	2021-04-23	2	19:00:00.000000	nab2@gmail.com	8697039865	OCTA	
16	2021-04-22	12	20:22:00.000000	nab2@gmail.com	9876987656	India Restaurant	
15	2021-04-25	8888888	12:22:00.000000	mv705@gmail.com	7059965591	India Restaurant	
13	2021-04-22	0000000	12:10:00.000000	mv705@gmail.com	7059965591	India Restaurant	
10	2021-04-25	13	20:35:00.000000	nab2@gmail.com	9876563377	OCTA	
5	2021-04-26	14	20:40:00.000000	nab2@gmail.com	9855555555	Urban Masala	

■ Console > >>  Show all Number of rows: 25  Filter rows: Search this table

## CONFIRMED BOOKING TABLE DATA

Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

Showing rows 0 - 6 (7 total, Query took 0.0011 seconds.)

```
SELECT * FROM `comb`
```

Show all | Number of rows: 25 | Filter rows: Search this table

+ Options

tid	day	people	time	email	phno
2	2021-04-16	2	14:00:00.000000	kk21@gmail.com	9831554183
7	2021-04-25	1	14:35:00.000000	rahul1@gmail.com	9831554184
8	2021-04-20	4	12:34:00.000000	yk333shaw@gmail.com	8013724948
9	2021-04-21	5	17:00:00.000000	lion@gmail.com	7891554183
9	2021-04-21	5	17:00:00.000000	lion@gmail.com	7891554183
9	2021-04-23	2	19:00:00.000000	nab2@gmail.com	8697039865
10	2021-04-25	13	20:35:00.000000	nab2@gmail.com	9876563377

Show all | Number of rows: 25 | Filter rows: Search this table

Query results operations

[Print](#) [Copy to clipboard](#) [Export](#) [Display chart](#) [Create view](#)

[Bookmark this SQL query](#)

Label:   Let every user access this bookmark

## USER DETAILS TABLE DATA

Browse Structure SQL Search Insert Export Import Privileges Operations

Showing rows 0 - 5 (6 total, Query took 0.0013 seconds.)

```
SELECT * FROM `user`
```

Show all Number of rows: 25 Filter rows: Search this table Sort by key: None

+ Options

	id	name	addr	phno	email	pwd
<input type="checkbox"/>	1	Yogesh Shaw	B-2/4 Brahmapur Northern Park	8013724948	yk333shaw@gmail.com	1234
<input type="checkbox"/>	2	Yogesh Shaw	B-2/4 Brahmapur Northern Park	8013724948	yk333shaw@gmail.com	9876
<input type="checkbox"/>	3	Yogesh Shaw	B-2/4 Brahmapur Northern Park	8013724948	yk333shaw@gmail.com	5678
<input type="checkbox"/>	6	Rahul	2A Padda Pukur Road	9831554184	rahul1@gmail.com	1234
<input type="checkbox"/>	7	Lion	B-2/4 Brahmapur Northern Park	7891554183	lion@gmail.com	1234
<input type="checkbox"/>	9	Naabil	B-2/4 Brahmapur Northern Park	8697039865	nab2@gmail.com	1234

Check all With selected:  Edit  Copy  Delete  Export

Show all Number of rows: 25 Filter rows: Search this table Sort by key: None

Query results operations

Print  Copy to clipboard  Export  Display chart  Create view

Bookmark this SQL query

Label:  Let every user access this bookmark

**TABLE REQUEST TABLE DATA**

Screenshot of a MySQL Workbench interface showing the results of a SELECT query on the 'user' table.

Query results:

```
SELECT * FROM `user`
```

Showing rows 0 - 5 (6 total, Query took 0.0013 seconds.)

	id	name	addr	phno	email	pwd
<input type="checkbox"/>	1	Yogesh Shaw	B-2/4 Brahmapur Northern Park	8013724948	yk333shaw@gmail.com	1234
<input type="checkbox"/>	2	Yogesh Shaw	B-2/4 Brahmapur Northern Park	8013724948	yk333shaw@gmail.com	9876
<input type="checkbox"/>	3	Yogesh Shaw	B-2/4 Brahmapur Northern Park	8013724948	yk333shaw@gmail.com	5678
<input type="checkbox"/>	6	Rahul	2A Padda Pukur Road	9831554184	rahul1@gmail.com	1234
<input type="checkbox"/>	7	Lion	B-2/4 Brahmapur Northern Park	7891554183	lion@gmail.com	1234
<input type="checkbox"/>	9	Naabil	B-2/4 Brahmapur Northern Park	8697039865	nab2@gmail.com	1234

Operations:

- Check all
- With selected:
  - Edit
  - Copy
  - Delete
- Export

Query results operations:

- Print
- Copy to clipboard
- Export
- Display chart
- Create view

Bookmark this SQL query:

Label:  Let every user access this bookmark

## RESTAURANT TABLE DATA

The screenshot shows the PHPMyAdmin interface for a MySQL database named 'reserve'. The current table is 'rest'. The page displays 4 rows of data from the 'rest' table.

rid	rname	raddr	rpno	remail	rpwd
1	India Restaurant	33,Level 5, Quest Mall , Syed Amir Ali Avenue , Ba	8013724948	yk333shaw@gmail.com	336699
2	OCTA	34, Karl Marx Sarani , Kidderpore , Kolkata	9874018767	yog333shaw@gmail.com	1234
3	Urban Masala	33,Level 5, Quest Mall , Syed Amir Ali Avenue , Ba	9831554183	yks12@gmail.com	5678
4	Friends Of Pho - Asian Fine Dining Club	196 ,Chakraborty Para , Rajdanga , Near Water Tank	9748079777	bh3@gmail.com	12345678

Below the table, there are buttons for 'Edit', 'Copy', and 'Delete' for each row. There is also a 'Check all' checkbox and a 'With selected:' dropdown menu containing 'Edit', 'Copy', 'Delete', and 'Export' options.

At the bottom of the page, there are links for 'Query results operations' including 'Print', 'Copy to clipboard', 'Export', 'Display chart', and 'Create view'. There is also a 'Bookmark this SQL query' button and a 'Label' input field with a 'Let every user access this bookmark' checkbox.

## **OUTPUT SCREEN**

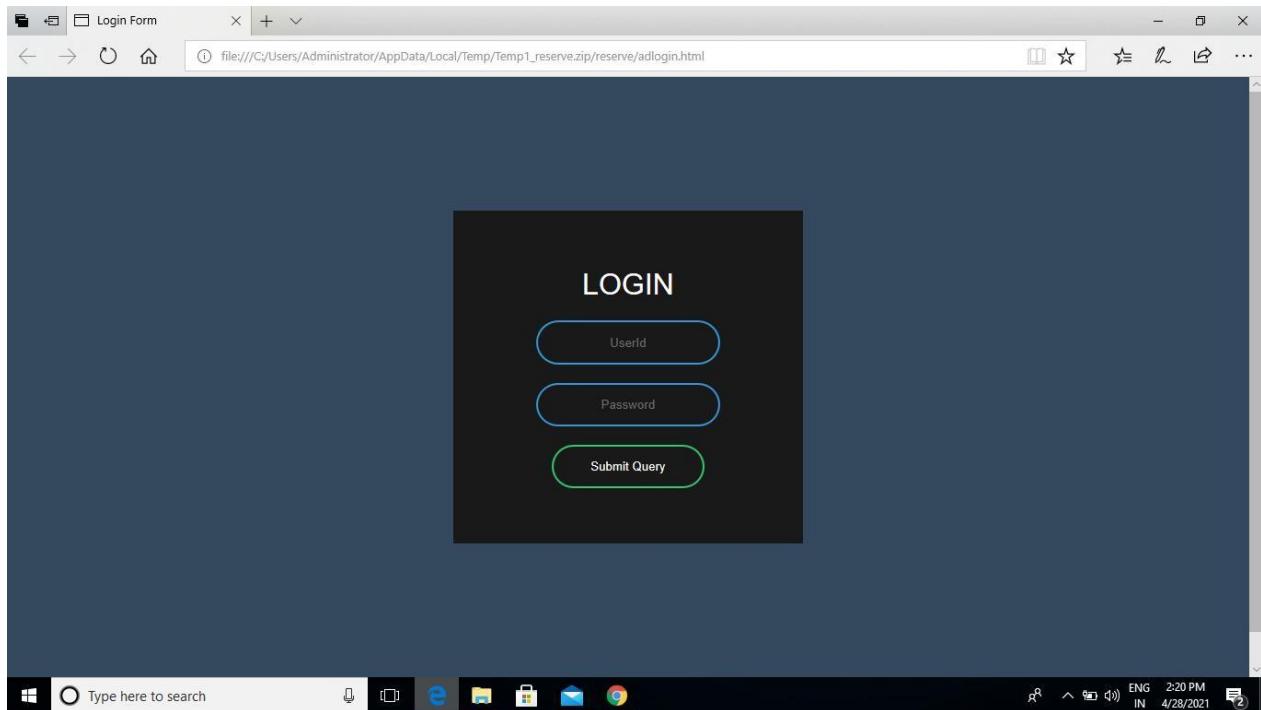
**USER INTERFACE DESIGN** User interface design (UID) or user interface engineering is the design of user interfaces for machines and software, such as computers, home appliances, mobile devices, and other electronic devices, with the focus on maximizing the user experience. The goal of user interface design is

to make the user's interaction as simple and efficient as possible, in terms of accomplishing user goals (usercentered design). Good user interface design facilitates finishing the task at hand without drawing unnecessary attention to it.

Graphic design and typography are utilized to support its usability, influencing how the user performs certain interactions and improving the aesthetic appeal of the design; design aesthetics may enhance or detract from the ability of users to use the functions of the interface. The design process must balance technical functionality and visual elements (e.g., mental model) to create a system that is not only operational but also usable and adaptable to changing user needs. Interface design is involved in a wide range of projects from computer systems, to cars, to commercial planes; all of these projects involve much of the same basic human interactions yet also require some unique skills and knowledge. As a result, designers tend to specialize in certain types of projects and have skills centered on their expertise, whether that be software design, user research, web design, or industrial design

## SNAPSHOTS

### LOGIN PAGE



HTML code:

```
<html>
<head>
<meta charset="utf-8" />
<title>Login Form</title>
<link href="StyleSheet2.css" rel="stylesheet" type="text/css" />
</head>
<body>
<style>
body
{
    margin: 0;
    padding: 0;
    font-family: Sans-Serif; background: #34495e;
}
.box
{
    width: 300px; padding:
    40px; position: absolute;
    top: 50%;
    left: 50%;
    transform: translate(-50%,-50%); background:
    #191919;
    text-align:center;
```

```

}

.box h1
{
color:White;
text-transform:uppercase; font-weight:
500;
}

.box input[type="text"],.box input[type="password"]
{
border:0; background:
none; display:block;
margin: 20px auto; text-
align:center;
border: 2px solid #3498db; padding:
14px 10px; width: 200px;
outline: none; color:
White;
border-radius: 24px;
transition:0.25s;
}

.box input[type="text"]:focus,.box input[type="password"]:focus
{
width: 280px;
border-color: #2ecc71;
}

.box input[type="submit"]
{
border:0; background:
none; display:block;
margin: 20px auto;
text-align:center;
border: 2px solid #2ecc71; padding:
14px 40px; outline: none;
color: White;
border-radius: 24px;
transition:0.25s;
cursor:pointer;
}

.box input[type="submit"]:hover
{
background:#2ecc71;
}

</style>

<form class="box" action="adlogin.py">

```

```

<h1>Login</h1>
<input type="text" placeholder="UserId" name="a1" required>
<input type="password" placeholder="Password" name="a2" required>
<input type="submit" text="Login">
</form>
</body>
</html>

```

**Python code:**

```

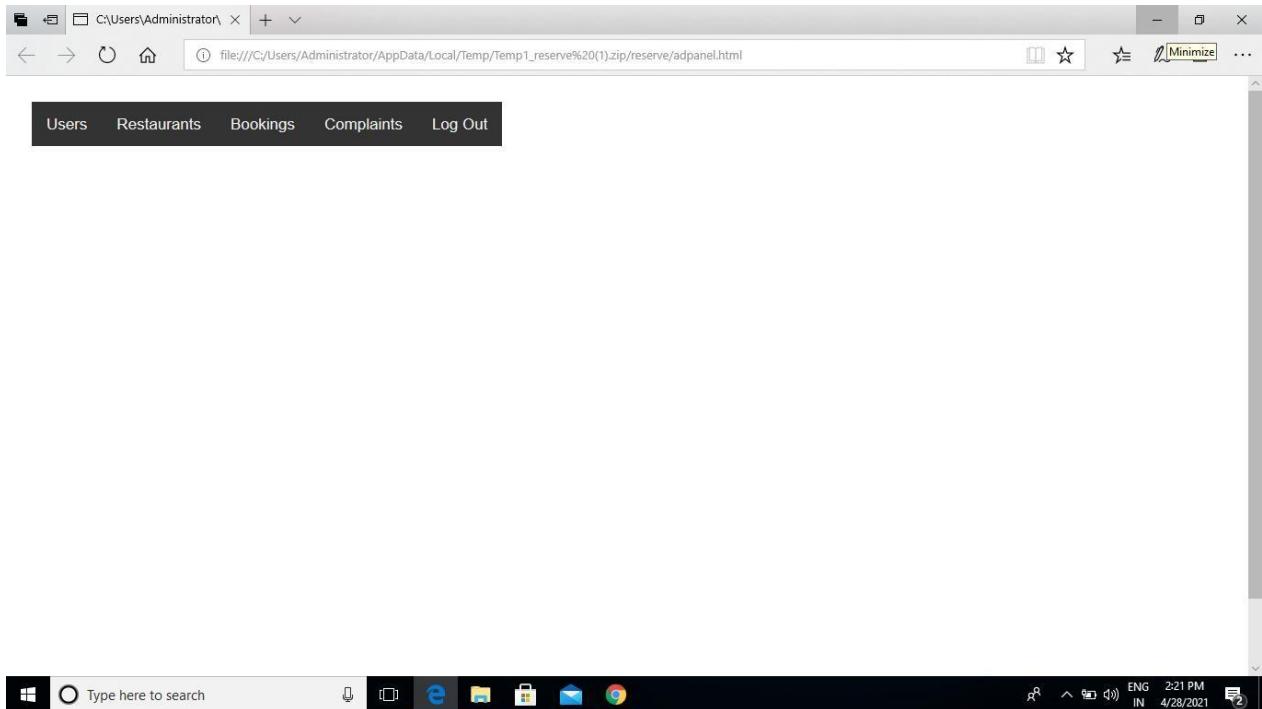
#!C:\Users\user1\AppData\Local\Programs\Python\Python38\pythonw.exe

import cgi
import mysql.connector
db=mysql.connector.connect(host="localhost",user="root",passwd="",db="rese
rve")
cur=db.cursor()
f=cgi.FieldStorage()
a=f.getvalue('a1')
b=f.getvalue('a2')
cur.execute("select * from ad")
q=cur.fetchall()
y=0
for i in q:
    if i[1]==a and i[2]==b:
        y=1
        break
    else:
        y=0

if y==1:
    print("location:adpanel.html?msg=done\r\n\r\n")
else:
    print("location:adlogin.html?msg=done\r\n\r\n")

```

ADMIN PANEL:



HTML code:

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
body {font-family: Arial, Helvetica, sans-serif;}
* {box-sizing: border-box; }

.bg-img {
/* The image used */
background-image: url("am.jpg");
min-height: 715px;

/* Center and scale the image nicely */
background-position: center; background-repeat: no-repeat;
background-size: cover;
/* Needed to position the navbar */
position: relative;
}

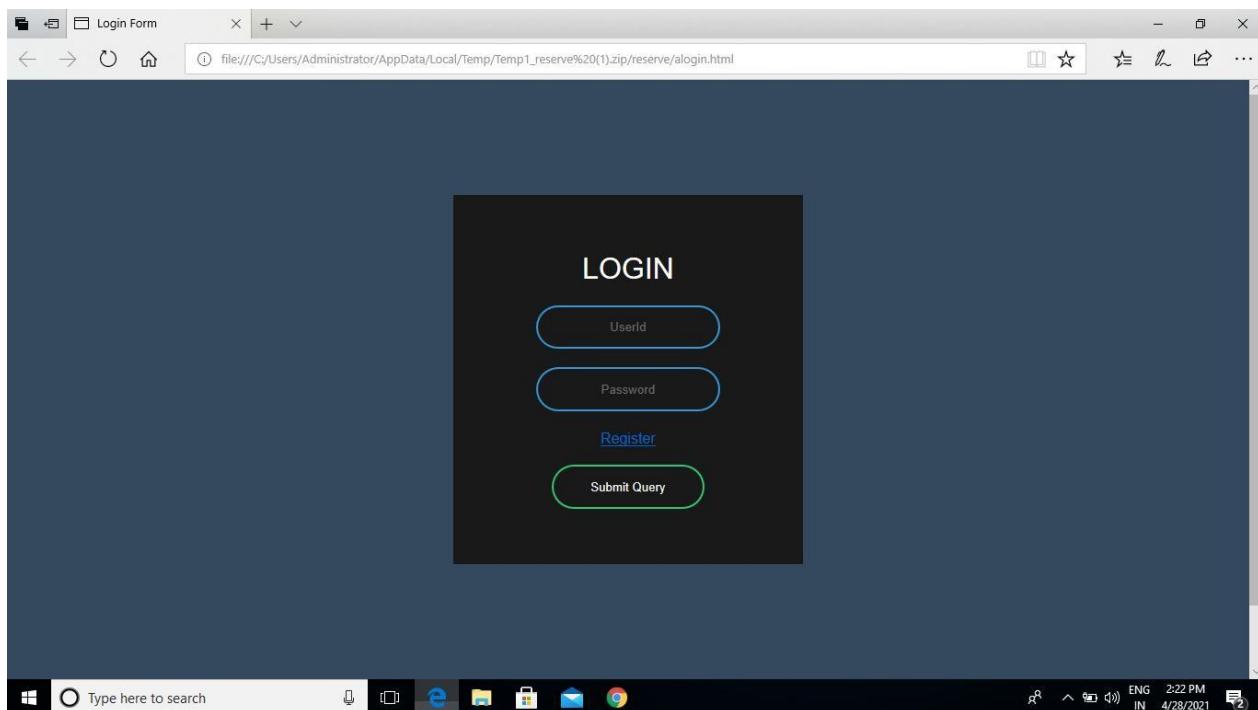
/* Position the navbar container inside the image */
.container {
position: absolute; margin: 20px;
width: auto;
}
```

```
/* The navbar */
.topnav { overflow: hidden;
background-color: #333;
}

/* Navbar links */
.topnav a { float: left;
color: #f2f2f2;
text-align: center; padding: 14px
16px; text-decoration: none;
font-size: 17px;
}
.topnav a:hover { background-color:
#ddd; color: black;
}

</style>
</head>
<body>
<div class="bg-img">
<div class="container">
<div class="topnav">
<a href="adu.py">Users</a>
<a href="adr.py">Restaurants</a>
<a href="adb.py">Bookings</a>
<a href="adc.py">Complaints</a>
<a href="home.html">Log Out</a>
</div>
</div>
</div>
</body>
</html>
```

ADMIN LOGIN:



HTML code:

```
<html>
<head>
<meta charset="utf-8" />
<title>Login Form</title>
<link href="StyleSheet2.css" rel="stylesheet" type="text/css" />
</head>
<body>
<style>
body
{
    margin: 0;
    padding: 0;
    font-family: Sans-Serif; background: #34495e;
}
.box
{
    width: 300px; padding:
    40px; position: absolute;
    top: 50%;
    left: 50%;
    transform: translate(-50%,-50%); background:
    #191919;
```

```
text-align:center;
}

.box h1
{
color:White;
text-transform:uppercase; font-weight:
500;

}

.box input[type="text"],.box input[type="password"]
{
border:0; background:
none; display:block;
margin: 20px auto; text-
align:center;
border: 2px solid #3498db; padding:
14px 10px; width: 200px;
outline: none; color:
White;
border-radius: 24px;
transition:0.25s;
}

.box input[type="text"]:focus,.box input[type="password"]:focus
{
width: 280px;
border-color: #2ecc71;
}

.box input[type="submit"]
{
border:0; background:
none; display:block;
margin: 20px auto; text-
align:center;
border: 2px solid #2ecc71; padding:
14px 40px; outline: none;
color: White;
border-radius: 24px;
transition:0.25s;
cursor:pointer;
}

.box input[type="submit"]:hover
{
background:#2ecc71;
}

</style>

<form class="box" action="adlogin.py">
```

```
<h1>Login</h1>
<input type="text" placeholder="UserId" name="a1" required>
<input type="password" placeholder="Password" name="a2" required>
<input type="submit" text="Login">
</form>
</body>
</html>
```

**PYTHON code:**

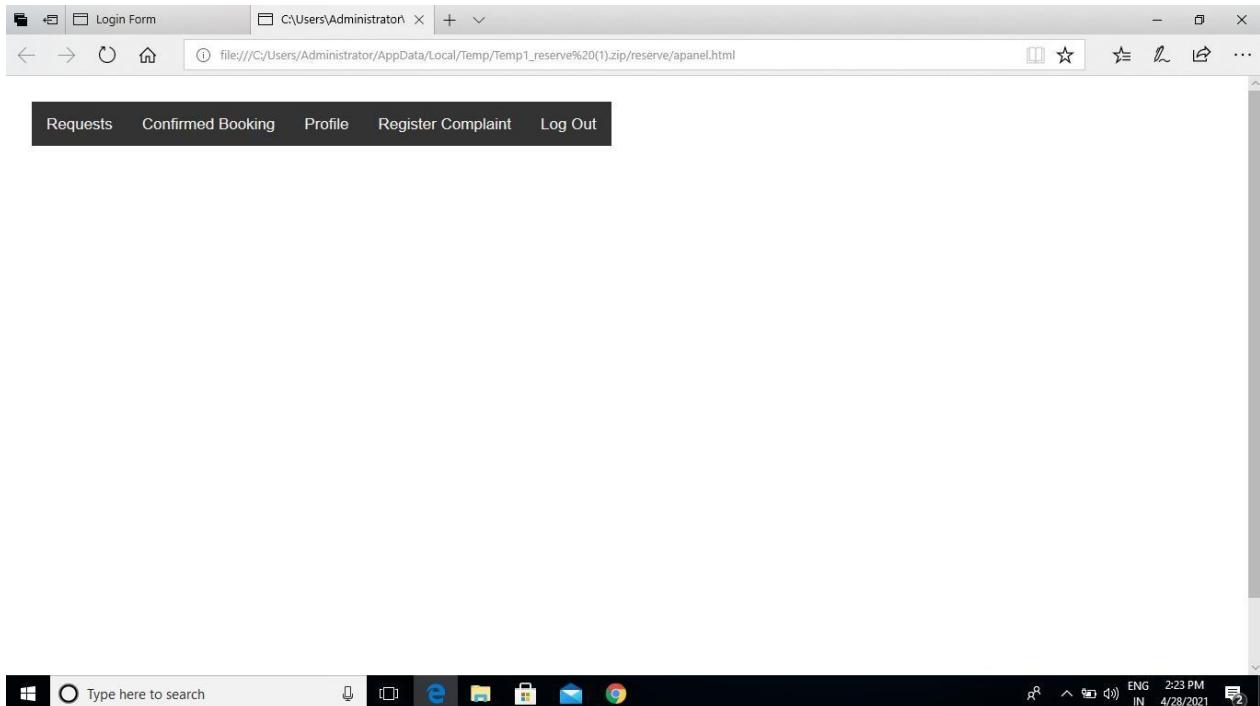
```
#!C:\Users\user1\AppData\Local\Programs\Python\Python38\pythonw.exe

import cgi
import mysql.connector
db=mysql.connector.connect(host="localhost",user="root",passwd="",db="rese
rve")
cur=db.cursor()
f=cgi.FieldStorage()
a=f.getvalue('a1')
b=f.getvalue('a2')
cur.execute("select * from rest")
q=cur.fetchall()
y=0
for i in q:
    if i[4]==a and i[5]==b:
        y=1
        break

else:
    y=0

if y==1: print("location:apanel.html?msg=done\r\n\r\n")
else:   print("location:alogin.html?msg=done\r\n\r\n")
```

ADMIN Panel:



HTML code:

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
body {font-family: Arial, Helvetica, sans-serif;}
* {box-sizing: border-box; }

.bg-img {
/* The image used */
background-image:
url("r1.jpg");
min-height: 715px;

/* Center and scale the image nicely */
background-position: center; background-repeat: no-repeat;
background-size: cover;

/* Needed to position the navbar */
position: relative;
}

/* Position the navbar container inside the image */
.container { position:
absolute; margin: 20px;
```

```
width: auto;

}

/* The navbar */

.topnav { overflow: hidden;
background-color: #333;
}

/* Navbar links */

.topnav a { float: left;
color: #f2f2f2;
text-align: center; padding: 14px
16px; text-decoration: none;
font-size: 17px;
}

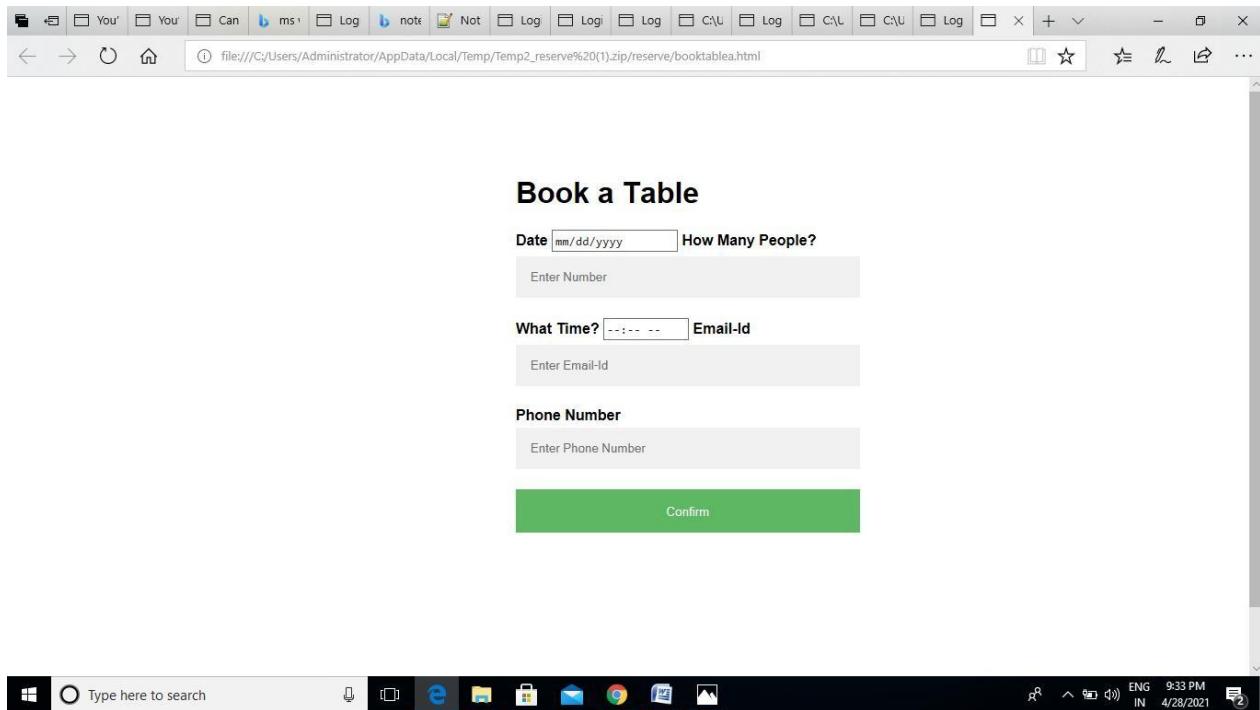
.topnav a:hover { background-color:
#ddd; color: black;
}

</style>
</head>
<body>
<div class="bg-img">
<div class="container">
<div class="topnav">
<a href="areq.py">Requests</a>
<a href="seecon.py"> Confirmed Booking</a>
<a href="portfolio.html">Profile</a>
<a href="comp.html">Register Complaint</a>
<a href="home.html">Log Out</a>
</div>
</div>
</div>
</body>
</html>
```

PYTHON code:

```
#!C:\Users\user1\AppData\Local\Programs\Python\Python38\pythonw.exe 0
import cgi
import mysql.connector
db=mysql.connector.connect(host="localhost",user="root",passwd="",db="rese
rve")
cur=db.cursor()
f=cgi.FieldStorage()
a=int(f.getvalue('id'))
b=f.getvalue('a1')
c=f.getvalue('a2')
d=f.getvalue('a3')
e=f.getvalue('a4')
g=f.getvalue('a5')
r="OCTA"
cur.execute("insert into comb (tid,day,people,time,email,phno)
values('%d','%s','%s','%s','%s','%s')"% (a,b,c,d,e,g))
cur.execute("insert into bs (tid,day,people,time,email,phno,rname)
values('%d','%s','%s','%s','%s','%s','%s')"% (a,b,c,d,e,g,r))
cur.execute("insert into st (tid,day,people,time,email,phno,rname)
values('%d','%s','%s','%s','%s','%s','%s')"% (a,b,c,d,e,g,r))
cur.execute("delete from tableb where tid=%d"% (a))
db.commit()
db.close()
print("location:breq.py?msg=done\r\n\r\n")
```

**BOOKING PAGE:**



HTML code:

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style> body,
html {
height: 100%;
font-family: Arial, Helvetica, sans-serif;
}

* {
box-sizing: border-box;
}

.bg-img {
/* The image used */
background-image:
url("t3.jpg");
min-height: 700px;

/* Center and scale the image nicely */
background-position: center;
background-repeat: no-repeat;
background-size: cover;
position: relative;
}
```

```

/* Add styles to the form container */

.container { position:
    absolute; top: 4%;
    left: 38%;

    max-width: 450px; padding: 38px;
    background-color: white;

    /* Full-width input fields */ input[type=text],
    input[type=password] { width: 100%;

        padding: 15px; margin: 5px 0
        22px 0; border: none;
        background: #f1f1f1;
    }

    input[type=text]:focus, input[type=password]:focus { background-color: #ddd;
        outline: none;
    }
}

/* Set a style for the submit button */

.btn {
    background-color: #4CAF50; color:
    white;
    padding: 16px 20px; border:
    none; cursor: pointer; width:
    100%;

    opacity: 0.9;
    .btn:hover {
        opacity: 1;
    }
}

</style>
</head>
<body>

<h2></h2>
<div class="bg-img">
<form action="tablea.py" class="container">
    <h1>Book a Table</h1>

    <label for="email"><b>Date</b></label>
    <input type="date" placeholder="Enter Day" name="a1" required>
    <label for="email"><b>How Many People? </b></label>
    <input type="text" placeholder="Enter Number" name="a2" required>
    <label for="email"><b>What Time?</b></label>
    <input type="time" placeholder="Enter Time" name="a3" required>

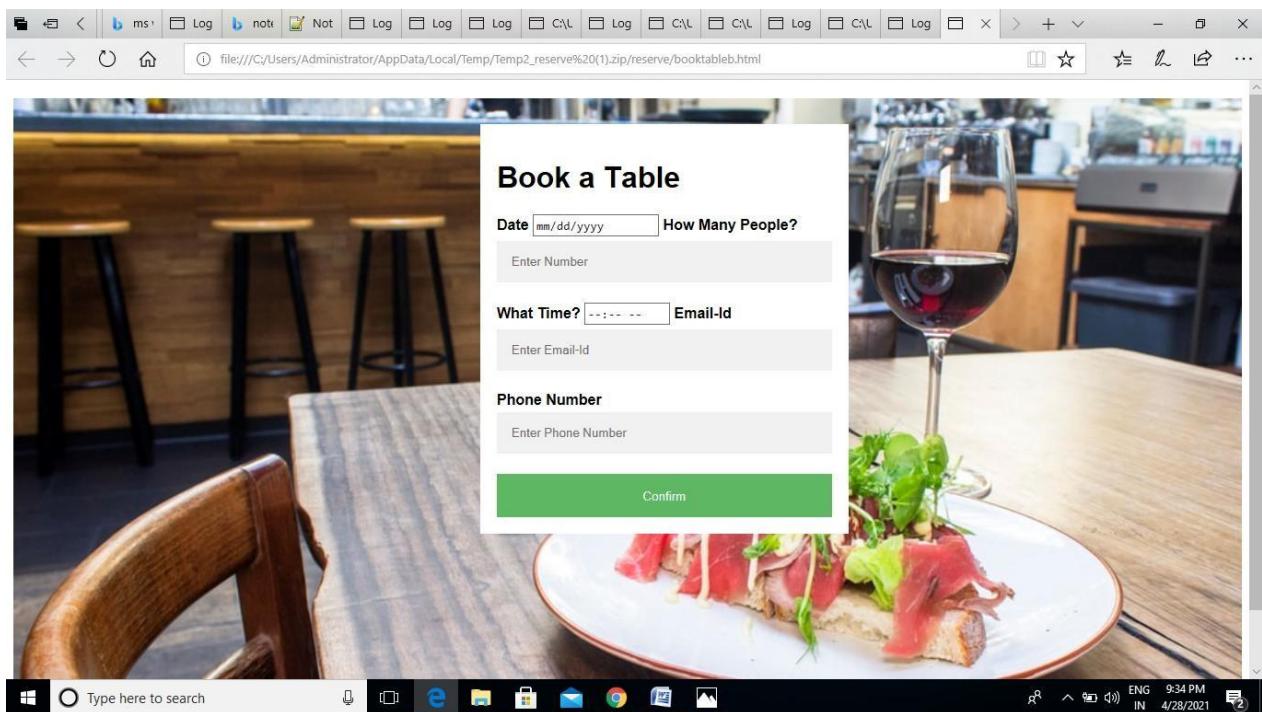
```

```
<label for="email"><b>Email-Id</b></label>
<input type="text" placeholder="Enter Email-Id" name="a4" required>

<label for="psw"><b>Phone Number</b></label>
<input type="text" placeholder="Enter Phone Number" name="a5" required>

<button type="submit" class="btn">Confirm</button>
</form>
</div>
</body>
</html>
```

TABLE BOOKING:



HTML code:

```
HT<!DOCTYPE html>

<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style> body,
html {
    height: 100%;
    font-family: Arial, Helvetica, sans-serif;
}
* {
    box-sizing: border-box;
}
.bg-img {
    /* The image used */
    background-image: url("t3.jpg");
    min-height: 700px;

    /* Center and scale the image nicely */
    position: center; background-repeat: no-repeat;
    background-size: cover;
    position: relative;
}

/* Add styles to the form container */
.container { position:
```

```
absolute; top: 4%;  
left: 38%;  
max-width: 450px; padding: 38px;  
background-color: white;  
  
/* Full-width input fields */ input[type=text],  
input[type=password] { width: 100%;  
padding: 15px; margin: 5px 0  
22px 0; border: none;  
background: #f1f1f1;  
  
}  
  
input[type=text]:focus, input[type=password]:focus { background-color: #ddd;  
outline: none;  
}  
  
/* Set a style for the submit button */  
.btn {  
background-color: #4CAF50; color:  
white;  
padding: 16px 20px; border:  
none; cursor: pointer; width:  
100%;  
opacity: 0.9;  
  
.btn:hover {  
opacity: 1;  
}  
</style>  
</head>  
<body>  
  
<h2></h2>  
<div class="bg-img">  
<form action="tablea.py" class="container">  
<h1>Book a Table</h1>  
  
<label for="email"><b>Date</b></label>  
<input type="date" placeholder="Enter Day" name="a1" required>  
<label for="email"><b>How Many People? </b></label>  
<input type="text" placeholder="Enter Number" name="a2" required>  
<label for="email"><b>What Time?</b></label>  
<input type="time" placeholder="Enter Time" name="a3" required>  
<label for="email"><b>Email-Id</b></label>
```

```
<input type="text" placeholder="Enter Email-Id" name="a4" required>

<label for="psw"><b>Phone Number</b></label>
<input type="text" placeholder="Enter Phone Number" name="a5" required>
<button type="submit" class="btn">Confirm</button>

</form>
</div>
</body>
</html>
```

## TABLE BOOKING

The screenshot shows a web browser window with a "Book a Table" form. The form fields are:

- Date:  mm/dd/yyyy
- How Many People?:  Enter Number
- What Time?:  hh:mm:ss
- Email-Id:  Enter Email-Id
- Phone Number:  Enter Phone Number

A large green "Confirm" button is at the bottom.



HTML code:

```
<!DOCTYPE html>

<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
body {font-family: Arial, Helvetica, sans-serif;}
* {box-sizing: border-box;}

.bg-img {
/* The image used */
background-image:
url("r4.jpg");
min-height: 715px;

/* Center and scale the image nicely */
background-position: center;
background-repeat: no-repeat;
background-size: cover;

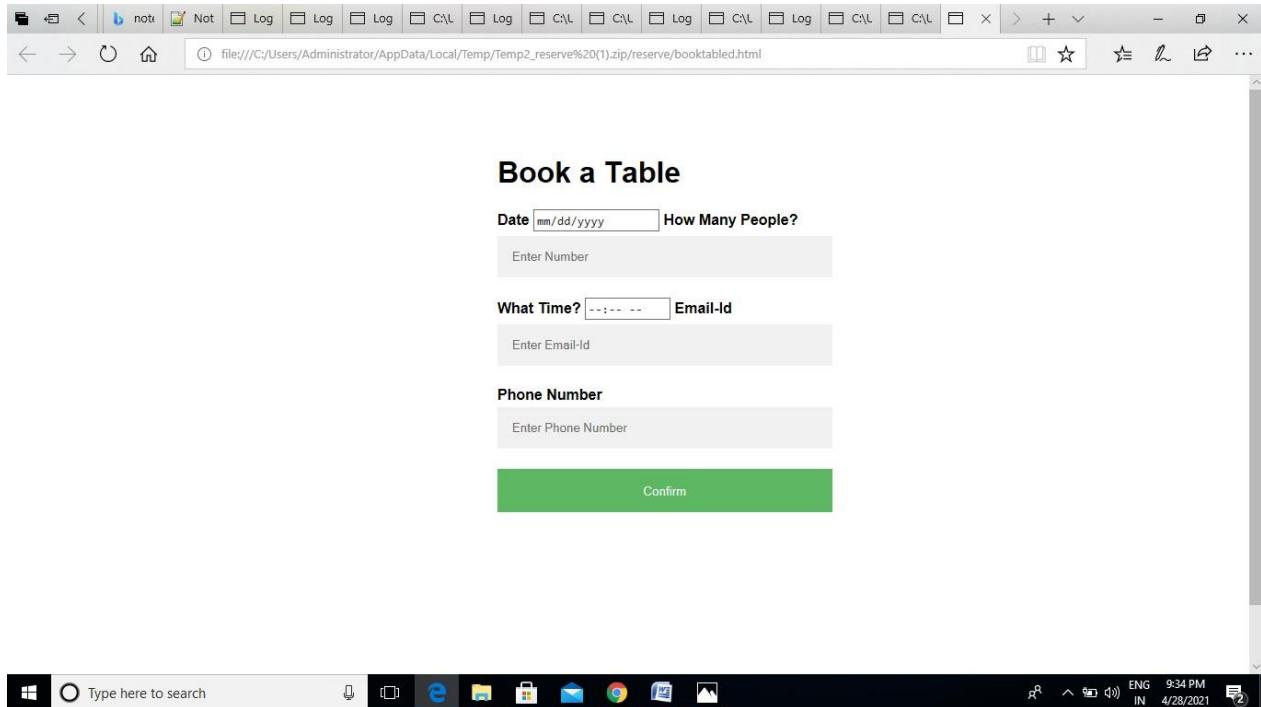
/* Needed to position the navbar */
position: relative;
}

/* Position the navbar container inside the image */
.container {
position:
absolute;
margin: 20px;
width: auto;
}
```

```
/* The navbar */
.topnav { overflow: hidden;
background-color: #333;
}
/* Navbar links */
.topnav a { float: left;
color: #f2f2f2;
text-align: center; padding: 14px
16px; text-decoration: none;
font-size: 17px;
}
.topnav a:hover { background-color:
#ddd; color: black;
}
</style>
</head>
<body>
<div class="bg-img">
<div class="container">
<div class="topnav">
<a href="breq.py">Requests</a>
<a href="seeconb.py">Confirmed Booking</a>
<a href="portfolioob.html">Profile</a>
<a href="comp.html">Register Complaint</a>
<a href="home.html">Log Out</a>
</div>
</div>
</div>
</body>
</html>:
```

## TABLE BOOKING

HTML code:



The screenshot shows a 'Book a Table' form in a web browser. The form consists of several input fields and labels. At the top, there are fields for 'Date' (mm/dd/yyyy) and 'How Many People?' (with a placeholder 'Enter Number'). Below that is a field for 'What Time?' (with a placeholder 'Enter Email-Id') and another for 'Email-Id'. Further down is a field for 'Phone Number' (with a placeholder 'Enter Phone Number'). At the bottom right of the form is a large green 'Confirm' button. The browser's address bar shows the URL 'file:///C:/Users/Administrator/AppData/Local/Temp/Tmp2\_reserve%20(1).zip/reserve/booktabled.html'. The taskbar at the bottom shows various pinned icons and the system clock.

```
html>
<head>
<meta charset="utf-8" />
<title>Register Form</title>
<link href="StyleSheet2.css" rel="stylesheet" type="text/css" />
</head>
<body>
<style>
body
{
    margin: 0;
    padding: 0;
    font-family: Sans-Serif; background: #34495e;
}
.box
{
    width: 400px; padding:
    40px; position: absolute;
    top: 50%;
    left: 50%;
    transform: translate(-50%,-50%); background:
    #191919;
    text-align:center;
}
```

```

.box h1
{
    color:White;
    text-transform:uppercase; font-weight:
    500;
}

.box input[type="text"],.box input[type="password"],.box input[type="email"]
{
    border:0; background: none;
    display:block;
    margin: 20px auto; text-
    align:center;
    border: 2px solid #3498db; padding:
    14px 10px; width: 200px;
    outline: none; color:
    White;
    border-radius: 24px;
    transition:0.25s;
}

.box input[type="text"]:focus,.box input[type="password"]:focus,.box input[type="email"]:focus
{
    width: 280px;
    border-color: #2ecc71;
}

.box input[type="submit"]
{
    border:0; background:
    none; display:block;
    margin: 20px auto; text-
    align:center;
    border: 2px solid #2ecc71; padding:
    14px 40px; outline: none;
    color: White;
    border-radius: 24px;
    transition:0.25s;
    cursor:pointer;
}

.box input[type="submit"]:hover
{
    background:#2ecc71;
}

</style>

<form class="box" action="register.py">
<h1>Register</h1>

```

```

<input type="text" placeholder="Name" name="a1" required>
<input type="text" placeholder="Address" name="a2" required>
<input type="text" placeholder="Phone Number" name="a3" required>
<input type="email" placeholder="E-mail Id" name="a4" required>
<input type="password" placeholder="Password" name="a5" required>
<input type="password" placeholder="Confirm Password" name="a6" required>
<input type="submit" text="Submit">
</form>
</body>
</html>

```

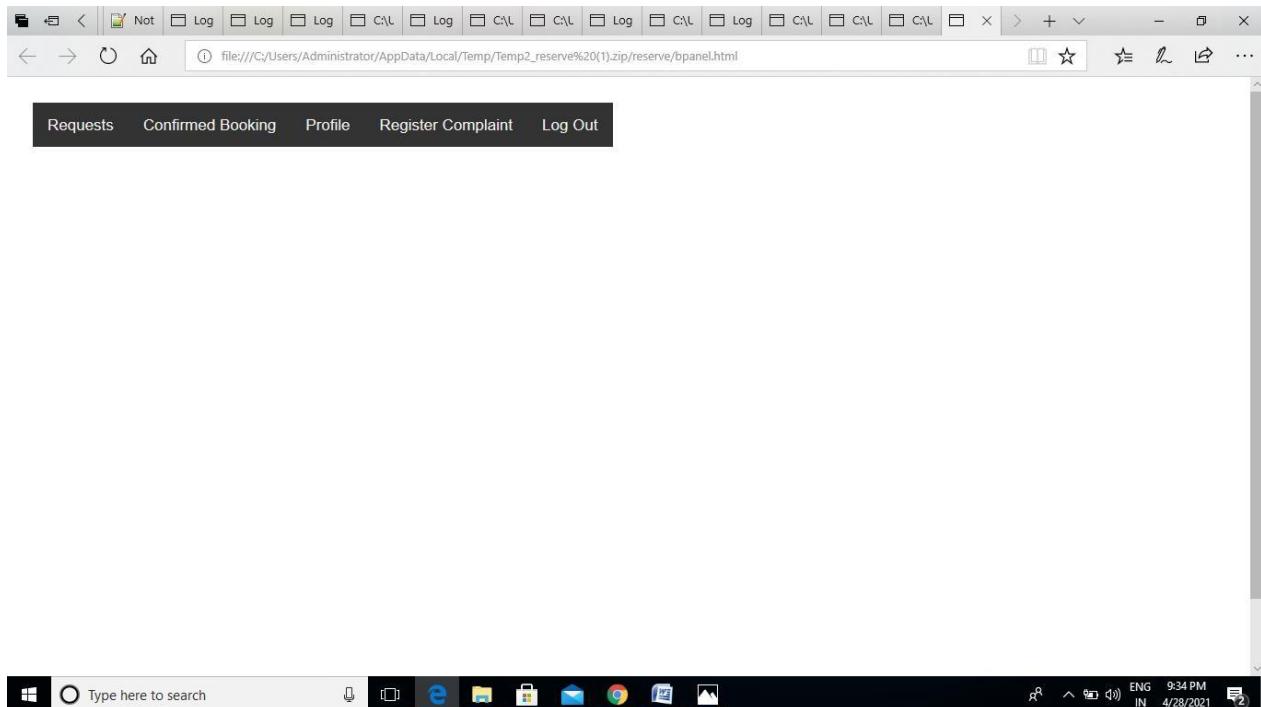
Python code:

```

#!/usr/bin/python3
#C:\Users\user1\AppData\Local\Programs\Python\Python38\pythonw.exe import cgi
import mysql.connector
db=mysql.connector.connect(host="localhost",user="root",passwd="",db="rese rve")
cur=db.cursor()
t=cgi.FieldStorage()
a=t.getvalue('a1')
b=t.getvalue('a2')
c=t.getvalue('a3')
d=t.getvalue('a4')
e=t.getvalue('a5')
cur.execute("insert into user (name,addr,phno,email,pwd)
values ('%s','%s','%s','%s','%s')%(a,b,c,d,e)) cur.execute("insert into log (email,pwd)
values ('%s','%s')%(d,e)) db.commit()
print("location:sr.html?msg=done\r\n\r\n")

```

## RESTAURANT PANEL



HTML code:

```
!DOCTYPE html>

<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
body {font-family: Arial, Helvetica, sans-serif;}
* {box-sizing: border-box;}
```

```
.bg-img {
/* The image used */
background-image:
url("r3.jpg");
```

```
min-height: 715px;
```

```
/* Center and scale the image nicely */
background-position: center; background-repeat: no-repeat;
background-size: cover;
```

```
/* Needed to position the navbar */
position: relative;
}
```

```
/* Position the navbar container inside the image */
.container {
position: absolute; margin: 20px;
```

```

width: auto;

}

/* The navbar */
.topnav { overflow: hidden;
background-color: #333;
}

/* Navbar links */
.topnav a { float: left;
color: #f2f2f2;
text-align: center; padding: 14px
16px; text-decoration: none;
font-size: 17px;
}

}

.topnav a:hover { background-color:
#ddd; color: black;
}

</style>
</head>
<body>
<div class="bg-img">
<div class="container">
<div class="topnav">
<a href="creq.py">Requests</a>
<a href="seeconc.py">Confirmed Booking</a>
<a href="portfolioc.html">Profile</a>
<a href="comp.html">Register Complaint</a>
<a href="home.html">Log Out</a>
</div>
</div>
</div>
</body>
</html>

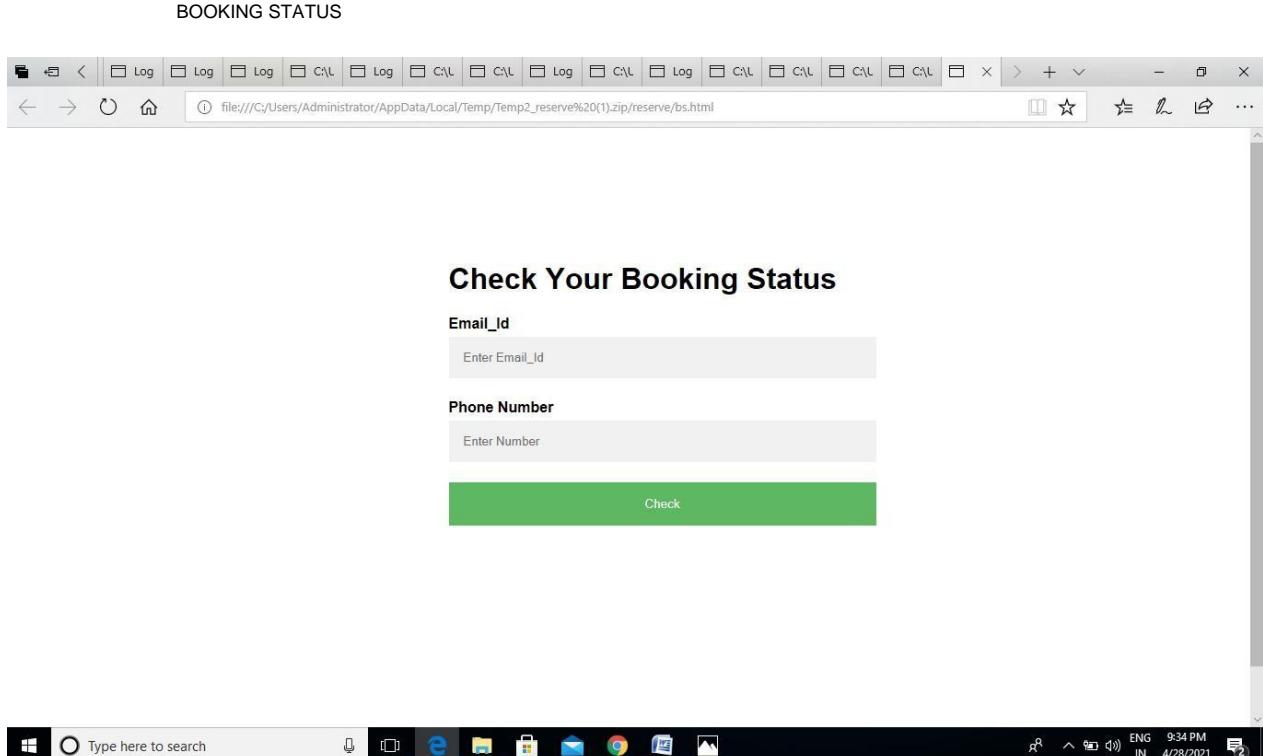
```

#### Python code:

```

#!/usr/bin/pythonw.exe
import cgi
import mysql.connector
db=mysql.connector.connect(host="localhost",user="root",passwd="",db="rese rve")
cur=db.cursor()
f=cgi.FieldStorage()
a=int(f.getvalue('id'))
b=f.getvalue('a1')
c=f.getvalue('a2')
d=f.getvalue('a3')
e=f.getvalue('a4')
g=f.getvalue('a5') r="India
Restaurant"
cur.execute("insert into con (tid,day,people,time,email,phno)
values('%d','%s','%s','%s','%s')%(a,b,c,d,e,g)") cur.execute("insert into bs
(tid,day,people,time,email,phno,rname) values('%d','%s','%s','%s','%s','%s')%(a,b,c,d,e,g,r)")
cur.execute("insert into st (tid,day,people,time,email,phno,rname)
values('%d','%s','%s','%s','%s')%(a,b,c,d,e,g,r)") cur.execute("delete from tablea where
tid=%d%(a)")
db.commit()
db.close()
print("location:areq.py?msg=done\r\n\r\n")

```



HTML code:

```
html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
body {
    font-family: Arial;
}
.coupon {
    border: 5px dotted #bbb; width:
    80%; border-radius: 15px; float:
    left; margin: 100 auto; max-
    width: 600px;
}
.bg-img {
/* The image used */ background-image:
url("sh.jpg");
    min-height: 700px;
/* Center and scale the image nicely */ background-
position: center; background-repeat: no-repeat;
background-size: cover; position: relative;
}
```

```
.container { padding: 2px  
16px;  
background-color: #f1f1f1;  
}  
.promo { background: #ccc;  
padding: 3px;  
}  
  
.expire { color:  
red;  
}  
</style>  
<style>  
body {  
font-family: Arial;  
}  
  
.coupons {  
border: 5px dotted #bbb; width:  
80%;  
border-radius: 15px; float:  
right;  
margin: 0 auto; max-width:  
600px;  
}  
.container { padding: 2px  
16px;  
background-color: #f1f1f1;  
}  
.bg-img {  
/* The image used */ background-image:  
url("sh.jpg");  
  
min-height: 700px;  
  
/* Center and scale the image nicely */ background-  
position: center; background-repeat: no-repeat;  
background-size: cover;  
position: relative;  
}  
  
.promo { background: #ccc;  
padding: 3px;
```

```
}

.expire { color:
  red;
}

</style>

<style>
* {box-sizing: border-box;}
body { margin:
  0;
font-family: Arial, Helvetica, sans-serif;
}

.header { overflow: hidden;
background-color: #f1f1f1; padding:
  20px 10px;
}

.header a { float:
  left; color: black;
text-align: center;
padding: 12px;
text-decoration: none; font-size:
  18px;
line-height: 25px; border-
radius: 4px;
}

.header a.logo { font-size:
  25px; font-weight: bold;
}

.header a:hover { background-color:
  #ddd; color: black;
}

.header a.active {
background-color: dodgerblue; color: white;
}

.header-right { float:
  right;
}

@media screen and (max-width: 500px) {
.header a { float: none;
display: block; text-
align: left;
```

```

}

.header-right { float:
    none;
}

}

</style>

</head>

<body>

<div class="header">

<a href="userpanel.html" class="logo">RESTAURANTS</a>

<div class="header-right">

<a class="active" href="home.html">HOME</a>

<a href="comp.html">Contact</a>

</div>

</div>

<div class="bg-img">

<div class="coupon">

<div class="container">

<h2>India Restaurant</h2>

</div>

<a href="booktable.html" ></a>

<div class="container" style="background-color:white">

<h3><b>Location</b></h3>

<p>33,Level 5, Quest Mall , Syed Amir Ali Avenue , Ballygunge , Kolkata</p>

<h3><b>Timings</b></h3>

<p>Mon-Fri              12noon - 11pm</p>

</div>

<div class="container">

<a href="booktablea.html" ><p> <span class="promo">Book a Table</span></p></a>

<a href="portfolio.html" ><p class="expire">View Details</p></a>

</div>

</div>

<div class="coupons">

<div class="container">

<h3> OCTA </h3>

</div>

<a href="booktable.html" ></a>

<div class="container" style="background-color:white">

<h3><b>Location</b></h3>

<p>34, Karl Marx Sarani , Kidderpore , Kolkata</p>

<h3><b>Timings</b></h3>

```

```

<p>Mon-Sat           12noon - 11pm</p>
</div>

<div class="container">
  <a href="booktableb.html"> <p> <span class="promo">Book a Table</span></p></a>
  <a href="portfoliob.html"><p class="expire">View Details</p></a>
</div>
</div>
</div>

<div class="bg-img">
<div class="coupons">
  <div class="container">
    <h3>Friends Of Pho - Asian Fine Dining Club</h3>
  </div>
  <a href="booktable.html"> </a>
  <div class="container" style="background-color:white">
    <h3><b>Location</b></h3>
    <p>196 ,Chakraborty Para , Rajdanga , Near Water Tank, Kasba, Kolkata</p>
    <h3><b>Timings --: Mon-Fri           12noon - 11pm</b></h3>
  </div>
<div class="container">
  <a href="booktableb.html"> <p> <span class="promo">Book a Table</span></p></a>
  <a href="portfoliob.html"><p class="expire">View Details</p></a>
</div>
</div>

<div class="coupon">
  <div class="container">
    <h3> Urban Masala</h3>
  </div>
  <a href="booktable.html"> </a>
  <div class="container" style="background-color:white">
    <h3><b>Location</b></h3>
    <p>33,Level 5, Quest Mall , Syed Amir Ali Avenue , Ballygunge , Kolkata</p>
    <h3><b>Timings</b></h3>
    <p>Mon-Fri           12noon - 11pm</p>
  </div>
<div class="container">
  <a href="booktablec.html"> <p> <span class="promo">Book a Table</span></p></a>
  <a href="portfolioc.html"><p class="expire">View Details</p></a>
</div>
</div>
</body>

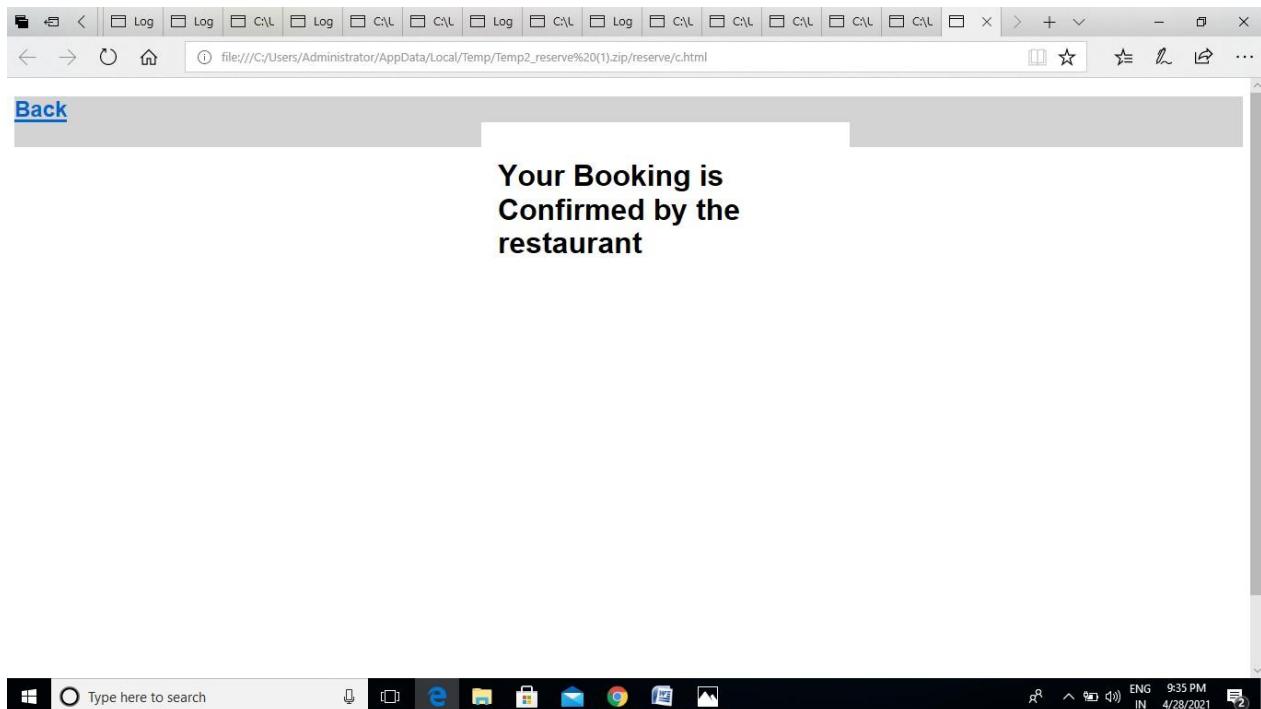
```

```
</html>
```

Python code:

```
#!/C:/Users/User/AppData/Local/Programs/Python/Python38/python.exe import cgi
import mysql.connector db=mysql.connector.connect(host="localhost",user="root",passwd="",db="reserve") cur=db.cursor()
f=cgi.FieldStorage() a=f.getvalue('a1')
b=f.getvalue('a2') cur.execute("select * from
st ") q=cur.fetchall()
y=0
for i in q:
if i[4]==a and i[5]==b: y=1
    break
else:
    y=0
if y==1:
    print("location:c.html?msg=done\r\n\r\n")
else:
    print("location:d.html?msg=done\r\n\r\n")
```

CONFIRMED BOOKING TABLE



HTML code:

```
<!DOCTYPE html>

<html>
<head>

<meta name="viewport" content="width=device-width, initial-scale=1">

<style> body,
html {
    height: 100%;
    font-family: Arial, Helvetica, sans-serif;
}

* {
    box-sizing: border-box;
}

.bg-img {
    /* The image used */
    background-image:
        url("sh.jpg");

    min-height: 700px;

    /* Center and scale the image nicely */
    background-position: center;
    background-repeat: no-repeat;
    background-size: cover;
    position: relative;
}

/* Add styles to the form container */
```

```

.container { position:
    absolute; top: 4%;
    left: 38%;

    max-width: 400px; padding: 18px;
    background-color: white;

/* Full-width input fields */ input[type=text],
input[type=password] { width: 100%;

    padding: 15px; margin: 5px 0
    22px 0; border: none;
    background: #f1f1f1;

}

input[type=text]:focus, input[type=password]:focus { background-color: #ddd;
outline: none;
}

/* Set a style for the submit button */

.btn {
background-color: #4CAF50; color:
white;
padding: 16px 20px; border:
none; cursor: pointer; width:
100%;
opacity: 0.9;

.opacity: 1;

}

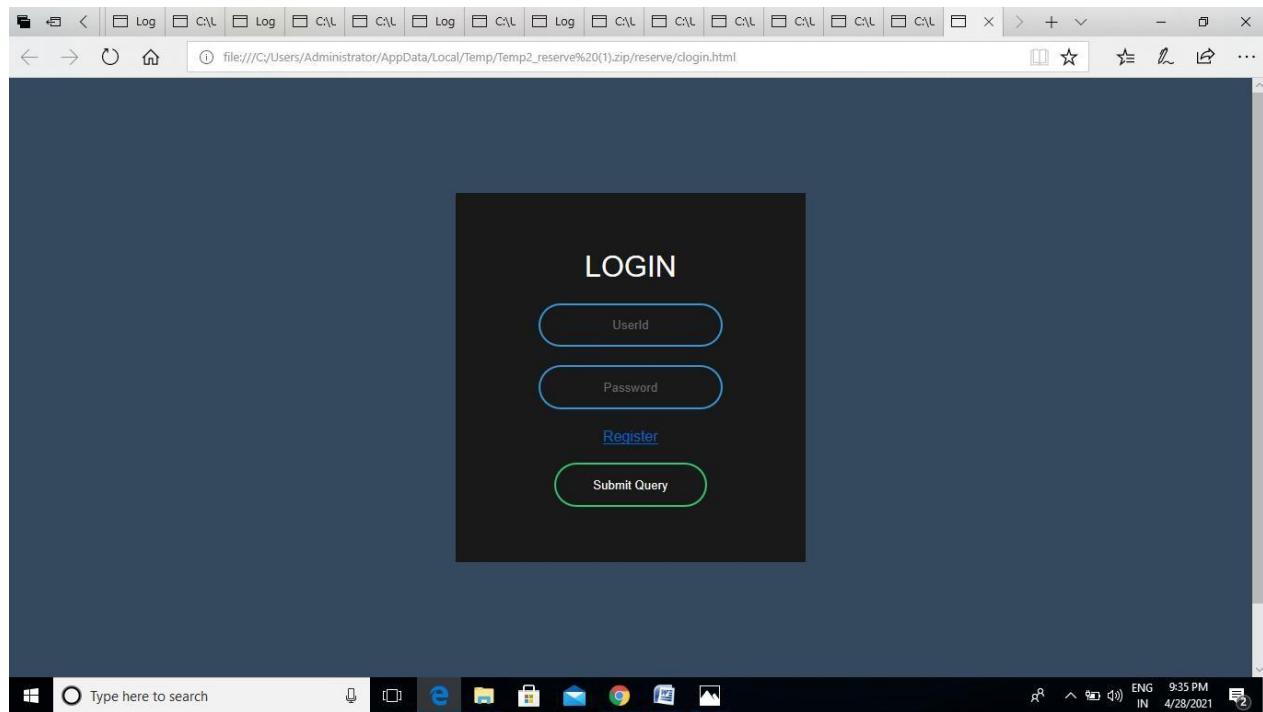
</style>
</head>
<body>

<div class="bg-img">
<h2 style="background-color:LightGray;"><a href="userpanel.html">Back</a><center>Booking Status</center></h2>
<form action="bs.py" class="container">
<h1> Your Booking is Confirmed by the restaurant</h1>
</form>
</div>

</body>
</html>

```

## RESTAURANT LOGIN



HTML code:

```
<html>
<head>
<meta charset="utf-8" />
<title>Login Form</title>
<link href="StyleSheet2.css" rel="stylesheet" type="text/css" />
</head>
<body>
<style>
body
{
    margin: 0;
    padding: 0;
    font-family: Sans-Serif; background: #34495e;
}
.box
{
    width: 300px; padding:
    40px; position: absolute;
    top: 50%;
    left: 50%;
    transform: translate(-50%,-50%); background:
    #191919;
    text-align:center;
}
```

```
.box h1
{
    color:White;
    text-transform:uppercase; font-weight:
    500;
}

.box input[type="text"],.box input[type="password"],.box input[type="email"]
{
    border:0; background: none;
    display:block;
    margin: 20px auto; text-
    align:center;
    border: 2px solid #3498db; padding:
    14px 10px; width: 200px;
    outline: none; color:
    White;
    border-radius: 24px;
    transition:0.25s;
}

.box input[type="text"]:focus,.box input[type="password"]:focus,.box input[type="email"]:focus
{
    width: 280px;
    border-color: #2ecc71;
}

.box input[type="submit"]
{
    border:0; background:
    none; display:block;
    margin: 20px auto; text-
    align:center;
    border: 2px solid #2ecc71; padding:
    14px 40px; outline: none;
    color: White;
    border-radius: 24px;
    transition:0.25s;
    cursor:pointer;
}

.box input[type="submit"]:hover
{
    background:#2ecc71;
}

</style>

<form class="box" action="clogin.py">
<h1>Login</h1>
```

```
<input type="email" placeholder="UserId" name="a1" required>
<input type="password" placeholder="Password" name="a2" required>
<a href="restaurantregister.html" >Register</a>
<input type="submit" text="Login">
</form>
</body>
</html>
```

## COMPLAINT PAGE

The screenshot shows a web browser window with a title bar containing multiple tabs, all labeled 'Complaint'. The active tab's URL is 'file:///C:/Users/Administrator/AppData/Local/Temp/Tmp2\_reserve%20(1).zip/reserve/comp.html'. The main content area is titled 'Complaint'. It contains several input fields: 'Date mm/dd/yyyy' (with a placeholder 'Issue?'), 'Name' (placeholder 'Enter Name'), 'Email-Id' (placeholder 'Enter Email-Id'), and 'Phone Number' (placeholder 'Enter Phone Number'). A large green 'Submit' button is at the bottom.



HTML code:

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style> body,
html {
height: 100%;
font-family: Arial, Helvetica, sans-serif;
}&lt;
*&lt;
```

```
box-sizing: border-box;
}
```

```
.bg-img {
/* The image used */ background-image:
url("sh.jpg");
```

```
min-height: 700px;
```

```
/* Center and scale the image nicely */ background-
position: center; background-repeat: no-repeat;
background-size: cover;
```

```

position: relative;
}

/* Add styles to the form container */

.container { position:
absolute; top: 4%;

left: 38%;

max-width: 400px; padding: 18px;

background-color: white;

}

/* Full-width input fields */ input[type=text],
input[type=password] { width: 100%;

padding: 15px; margin: 5px 0

22px 0; border: none;

background: #f1f1f1;

}

input[type=text]:focus, input[type=password]:focus { background-color: #ddd;

outline: none;

}

/* Set a style for the submit button */

.btn {

background-color: #4CAF50; color:

white;

padding: 16px 20px; border:

none; cursor: pointer; width:

100%;

opacity: 0.9;

}

.btn:hover {

opacity: 1;

}

</style>

</head>

<body>

<div class="bg-img">

<form action="comp.py" class="container">

<h1>Complaint</h1>

<label for="email"><b>Date</b></label>

<input type="date" placeholder="Enter Day" name="a1" required>

<label for="email"><b>Issue? </b></label>

<input type="text" placeholder="Enter Details" name="a2" required>

<label for="email"><b> Name</b></label>

```

```

<input type="text" placeholder="Enter Name" name="a3" required>
<label for="email"><b>Email-Id</b></label>
<input type="text" placeholder="Enter Email-Id" name="a4" required>

<label for="psw"><b>Phone Number</b></label>
<input type="text" placeholder="Enter Phone Number" name="a5" required><button type="submit" class="btn">Submit</button>
</form>
</div>

</body>
</html>

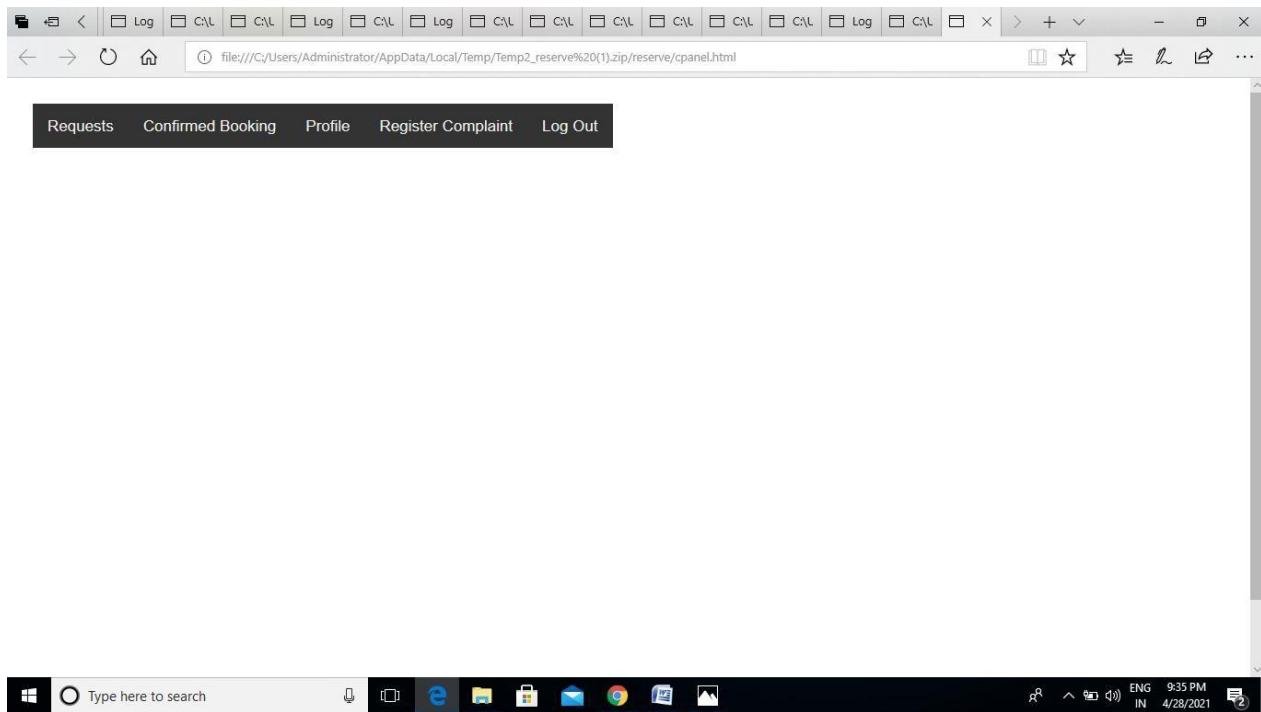
```

```

Python code: #!C:\Users\User\AppData\Local\Programs\Python\Python38\python.exe import cgi
import mysql.connector db=mysql.connector.connect(host="localhost",user="root",passwd="",db="reserve")
cur=db.cursor()
t=cgi.FieldStorage()
a=t.getvalue('a1')
b=t.getvalue('a2')
c=t.getvalue('a3')
d=t.getvalue('a4')
e=t.getvalue('a5')
cur.execute("insert into comp (day,issue,name,email,phno) values('%s','%s','%s','%s','%s')%(a,b,c,d,e)) db.commit()
print("location:sr.html?msg=Success\r\n\r\n")

```

## RESTAURANT PANEL



HTML code:

```
<!DOCTYPE html>

<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
body {font-family: Arial, Helvetica, sans-serif;}
* {box-sizing: border-box;}
.bg-img {
/* The image used */
background-image:
url("r3.jpg");
min-height: 715px;

/* Center and scale the image nicely */
background-position: center; background-repeat: no-repeat;
background-size: cover;

/* Needed to position the navbar */
position: relative;
}

/* Position the navbar container inside the image */
.container {
position:
absolute; margin: 20px;
width: auto;
}
```

```
/* The navbar */
.topnav { overflow: hidden;
background-color: #333;
}

/* Navbar links */
.topnav a { float: left;
color: #f2f2f2;
text-align: center; padding: 14px
16px; text-decoration: none;
font-size: 17px;
}

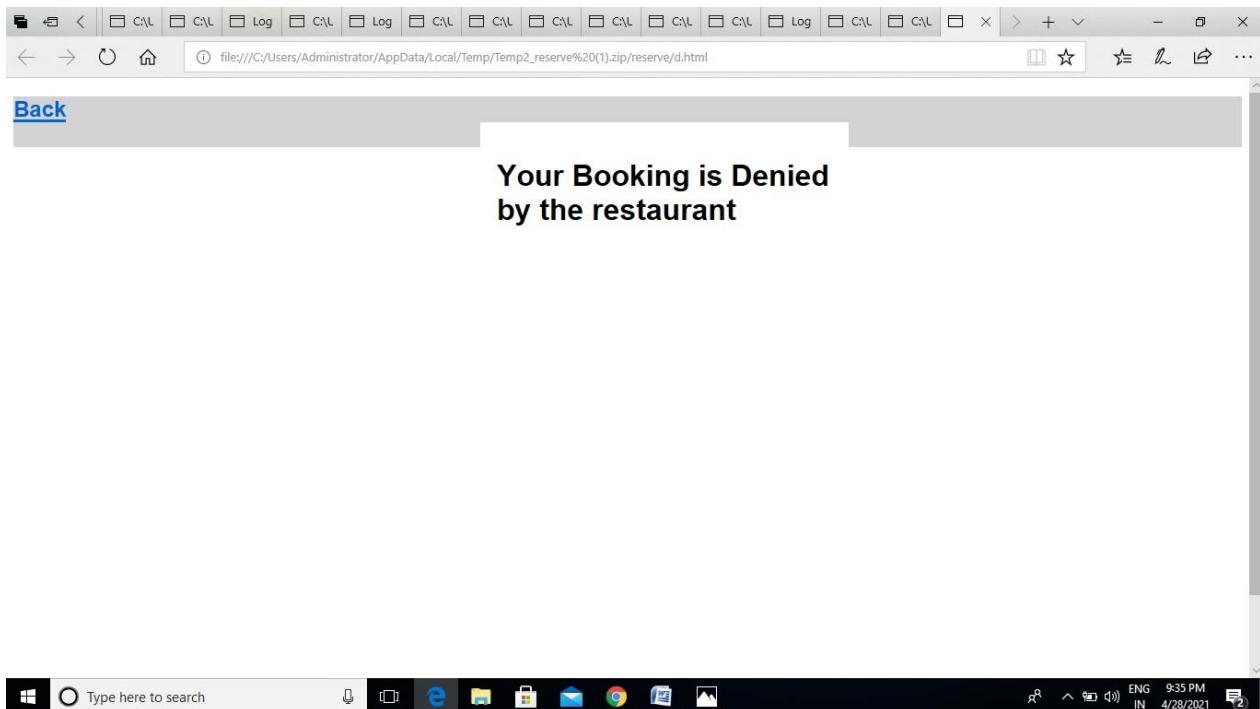
.topnav a:hover { background-color:
#ddd; color: black;
}

</style>
</head>
<body>

<div class="bg-img">
<div class="container">
<div class="topnav">
<a href="creq.py">Requests</a>
<a href="seeconc.py">Confirmed Booking</a>
<a href="portifolioc.html">Profile</a>
<a href="comp.html">Register Complaint</a>
<a href="home.html">Log Out</a>
</div>
</div>
</div>

</body>
</html>
```

BOOKING DENIED



HTML code:

```
<!DOCTYPE html>

<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style> body,
html {
    height: 100%;
    font-family: Arial, Helvetica, sans-serif;
}
* {
    box-sizing: border-box;
}

.bg-img {
    /* The image used */
    background-image:
        url("sh.jpg");
    min-height: 700px;
    /* Center and scale the image nicely */
    background-position: center;
    background-repeat: no-repeat;
    background-size: cover;
    position: relative;
}
/* Add styles to the form container */
```

```

.container { position:
absolute; top: 4%;
left: 38%;  

max-width: 400px; padding: 18px;
background-color: white;  

/* Full-width input fields */ input[type=text],
input[type=password] { width: 100%;  

padding: 15px; margin: 5px 0  

22px 0; border: none;  

background: #f1f1f1;  

}  

input[type=text]:focus, input[type=password]:focus { background-color: #ddd;  

outline: none;  

}  

/* Set a style for the submit button */  

.btn {  

background-color: #4CAF50; color:  

white;  

padding: 16px 20px; border:  

none; cursor: pointer; width:  

100%;  

opacity: 0.9;  

.btn:hover {  

opacity: 1;  

}  

</style>  

</head>  

<body>  

<div class="bg-img">  

<h2 style="background-color:LightGray;"><a href="userpanel.html">Back</a><center>Booking Status</center></h2>  

<form action="bs.py" class="container">  

<h1>You have no bookings yet ,or,if you requested for a table then your booking might have been pending approval/rejection from restaurant</h1>  

<h1>Please check again later </h1>  

</form>  

</div>  

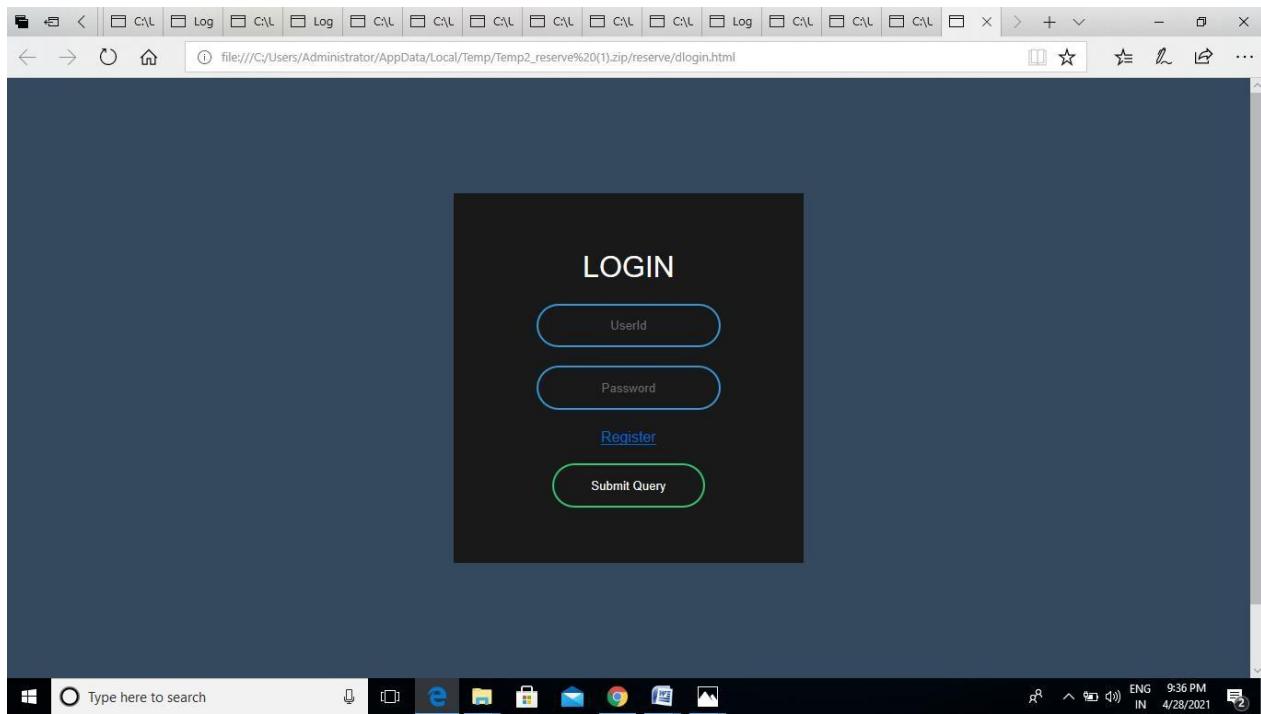
  

</body>  

</html>

```

## RESTAURANT LOGIN



HTML code:

```
<html>
<head>
<meta charset="utf-8" />
<title>Login Form</title>
<link href="StyleSheet2.css" rel="stylesheet" type="text/css" />
</head>
<body>
<style>
body
{
    margin: 0;
    padding: 0;
    font-family: Sans-Serif; background: #34495e;
}
.box
{
    width: 300px; padding:
    40px; position: absolute;
    top: 50%;
    left: 50%;
    transform: translate(-50%,-50%); background:
    #191919;
    text-align:center;
}
```

```
.box h1
{
    color:White;
    text-transform:uppercase; font-weight:
    500;
}

.box input[type="text"],.box input[type="password"],.box input[type="email"]
{
    border:0; background: none;
    display:block;
    margin: 20px auto; text-
    align:center;
    border: 2px solid #3498db; padding:
    14px 10px; width: 200px;
    outline: none; color:
    White;
    border-radius: 24px;
    transition:0.25s;
}

.box input[type="text"]:focus,.box input[type="password"]:focus,.box input[type="email"]:focus
{
    width: 280px;
    border-color: #2ecc71;
}

.box input[type="submit"]
{
    border:0; background:
    none; display:block;
    margin: 20px auto; text-
    align:center;
    border: 2px solid #2ecc71; padding:
    14px 40px; outline: none;
    color: White;
    border-radius: 24px;
    transition:0.25s;
    cursor:pointer;
}

.box input[type="submit"]:hover
{
    background:#2ecc71;
}

</style>

<form class="box" action="dlogin.py">
<h1>Login</h1>
```

```
<input type="email" placeholder="UserId" name="a1" required>
<input type="password" placeholder="Password" name="a2" required>
<a href="restaurantregister.html" >Register</a>
<input type="submit" text="Login">
</form>
</body>
</html>
```

Python code:

```
#!C:\Users\User\AppData\Local\Programs\Python\Python38\python.exe
```

```
import cgi

import mysql.connector
db=mysql.connector.connect(host="localhost",user="root",passwd="",db="reserve")
cur=db.cursor()

f=cgi.FieldStorage()
a=f.getvalue('a1')
b=f.getvalue('a2')
cur.execute("select * from rest")
q=cur.fetchall()

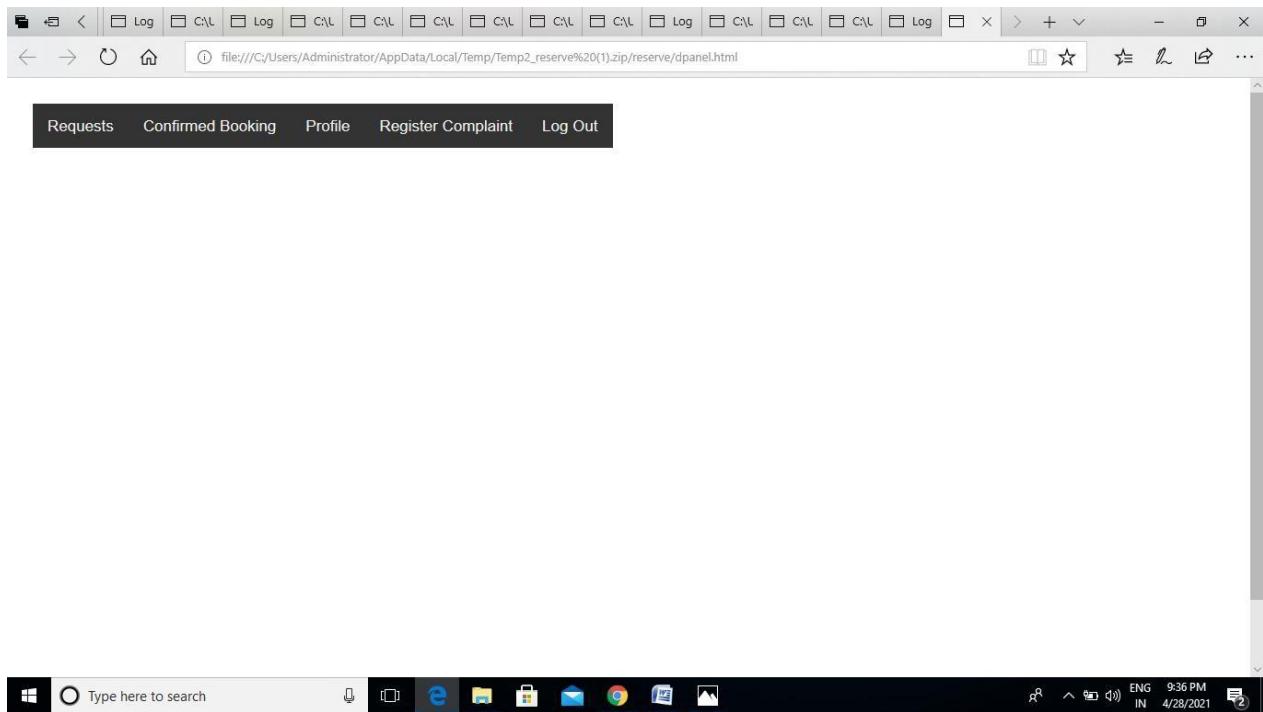
y=0

for i in q:
    if i[4]==a and i[5]==b: y=1
    break

else:
    y=0

if y==1:
    print("location:dpanel.html?msg=done\r\n\r\n")
else:
    print("location:dlogin.html?msg=done\r\n\r\n")
```

## RESTAURANT PANEL



HTML code:

```
<!DOCTYPE html>

<html>

<head>

<meta name="viewport" content="width=device-width, initial-scale=1">

<style>

body {font-family: Arial, Helvetica, sans-serif;}

* {box-sizing: border-box;}
```

```
.bg-img {

/* The image used */ background-image:
url("r2.jpg");

min-height: 715px;

/* Center and scale the image nicely */ background-
position: center; background-repeat: no-repeat;
background-size: cover;

/* Needed to position the navbar */ position: relative;
```

```
}
```

```
/* Position the navbar container inside the image */

.container { position:
absolute; margin: 20px;
width: auto;
```

```
}

/* The navbar */
.topnav { overflow: hidden;
background-color: #333;
}

/* Navbar links */
.topnav a { float: left;
color: #f2f2f2;
text-align: center; padding: 14px
16px; text-decoration: none;
font-size: 17px;
}

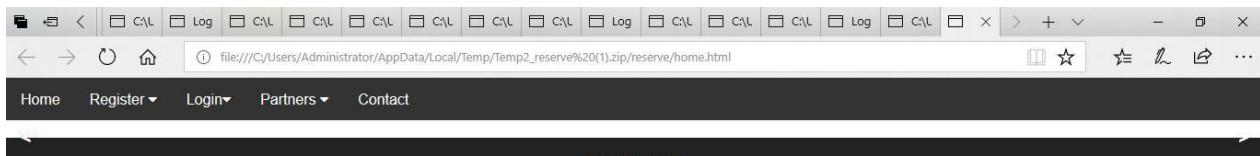
.topnav a:hover { background-color:
#ddd; color: black;
}

</style>
</head>
<body>

<div class="bg-img">
<div class="container">
<div class="topnav">
<a href="dreq.py">Requests</a>
<a href="seecond.py">Confirmed Booking</a>
<a href="portfoliod.html">Profile</a>
<a href="comp.html">Register Complaint</a>
<a href="home.html">Log Out</a>
</div>
</div>
</div>

</body>
</html>
```

## RESTAURANT HOMEPAGE



HTML code:

```
<!DOCTYPE html>

<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
<style>
body {
    font-family: Arial, Helvetica, sans-serif; margin: 0;
}
.navbar {
    overflow: hidden; background-
    color: #333;
}
<div
<div class="numbertext">6 /
6 /  
    
<style>
body {
font-family: Arial; margin:
0;
}

* {
box-sizing: border-box;
}

img {
vertical-align: middle;
}

<div
<div class="numbertext">6 /
6 /
  6 /  


</head>

<body>

<div class="navbar">

<a href="#home">Home</a>

<div class="subnav">

<button class="subnavbtn">Register <i class="fa fa-caret-down"></i></button>

<div class="subnav-content">

<a href="register.html">User</a>

<a href="restaurantregister.html">Restaurant</a>

</div>

</div>

<div class="subnav">

<button class="subnavbtn">Login<i class="fa fa-caret-down"></i></button>

<div class="subnav-content">

<a href="login.html">User</a>

<a href="restaurantlogin.html">Restaurant</a>

</div>

</div>

<div class="subnav">

<button class="subnavbtn">Partners <i class="fa fa-caret-down"></i></button>

<div class="subnav-content">

<a href="portfolio.html">India Restaurant</a>

<div

<div class="numbertext">6 /



```
<a href="portfolioob.html">OCTA</a>
<a href="portfolioc.html">Urban Masala</a>
<a href="portfolioid.html">Friends Of Pho - Asian Fine Dining Club</a>
</div>
</div>
<a href="comp.html">Contact</a>
</div>
<div style="padding:0 16px">
</div>
<div class="container">
<div class="mySlides">
<div class="numbertext">2 / 6</div>

</div>

<div class="mySlides">
<div class="numbertext">3 / 6</div>

</div>

<div class="mySlides">
<div class="numbertext">4 / 6</div>

</div>

<div class="mySlides">
<div class="numbertext">5 / 6</div>

</div>

<div>
<div class="numbertext">6 / </div>

```

```

<a class="prev" onclick="plusSlides(-1)"><</a>
<a class="next" onclick="plusSlides(1)">></a>

<div class="caption-container">
  <p id="caption"></p>
</div>

<div class="row">
  <div class="column">
    
  </div>
  <div class="column">
    
  </div>
</div>
</div>

<script>
var slideIndex = 1;
showSlides(slideIndex);

function plusSlides(n) {
  showSlides(slideIndex += n);
}

function currentSlide(n) {
  showSlides(slideIndex = n);
}

function showSlides(n) { var i;
  var slides = document.getElementsByClassName("mySlides");
  var dots =

```

```
document.getElementsByName("demo");

var captionText = document.getElementById("caption"); if (n > slides.length)
{slideIndex = 1}

if (n < 1) {slideIndex = slides.length}
for (i = 0; i < slides.length; i++) {
    slides[i].style.display = "none";
}

for (i = 0; i < dots.length; i++) {
    dots[i].className = dots[i].className.replace(" active", "");
}

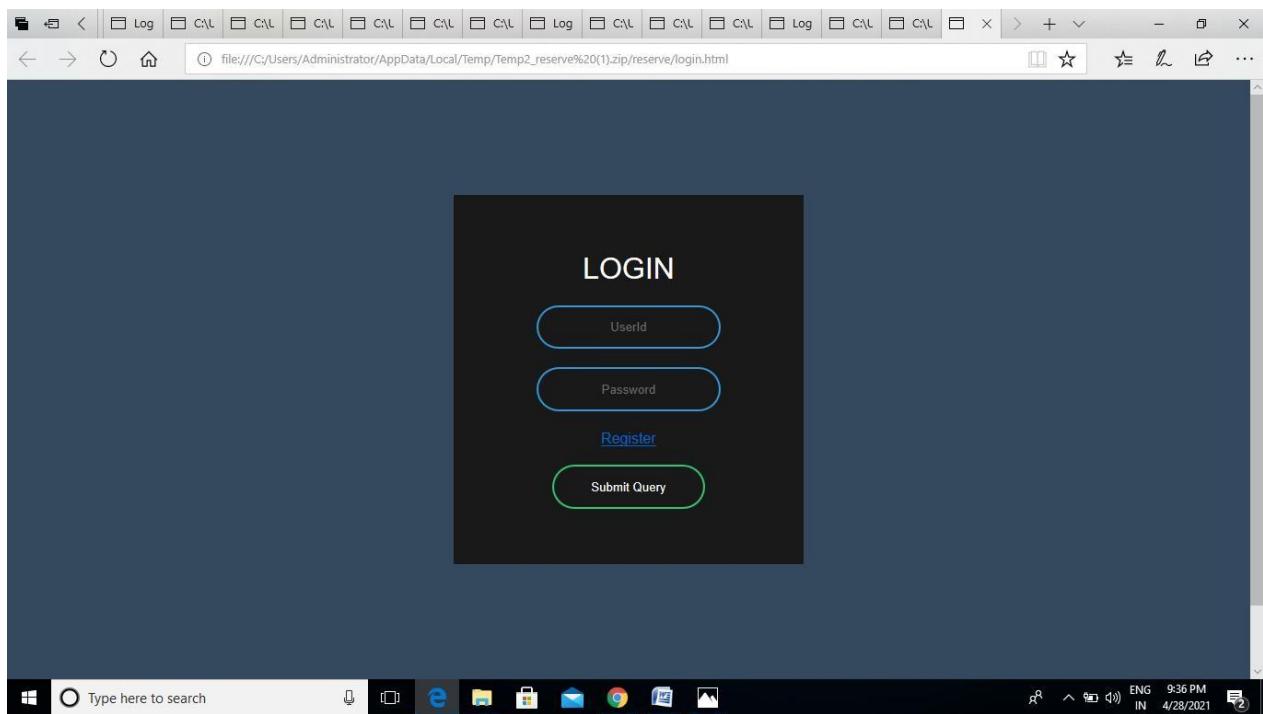
slides[slideIndex-1].style.display = "block"; dots[slideIndex-1].className += " active"; captionText.innerHTML = dots[slideIndex-1].alt;

}

</script>

</body>
</html>
```

## RESTAURANT LOGIN



HTML code:

```
<html>
<head>
<meta charset="utf-8" />
<title>Login Form</title>
<link href="StyleSheet2.css" rel="stylesheet" type="text/css" />
</head>
<body>
<style>
body
{
    margin: 0;
    padding: 0;
    font-family: Sans-Serif; background: #34495e;
}
.box
{
    width: 300px; padding:
    40px; position: absolute;
    top: 50%;
    left: 50%;
    transform: translate(-50%,-50%); background:
    #191919;
    text-align:center;
}
```

```
.box h1
{
    color:White;
    text-transform:uppercase; font-weight:
    500;
}

.box input[type="text"],.box input[type="password"],.box input[type="email"]
{
    border:0; background: none;
    display:block;
    margin: 20px auto; text-
    align:center;
    border: 2px solid #3498db; padding:
    14px 10px; width: 200px;
    outline: none; color:
    White;
    border-radius: 24px;
    transition:0.25s;
}

.box input[type="text"]:focus,.box input[type="password"]:focus,.box input[type="email"]:focus
{
    width: 280px;
    border-color: #2ecc71;
}

.box input[type="submit"]
{
    border:0; background:
    none; display:block;
    margin: 20px auto; text-
    align:center;
    border: 2px solid #2ecc71; padding:
    14px 40px; outline: none;
    color: White;
    border-radius: 24px;
    transition:0.25s;
    cursor:pointer;
}

.box input[type="submit"]:hover
{
    background:#2ecc71;
}

</style>

<form class="box" action="login.py" method="get" >
<h1>Login</h1>
```

```
<input type="email" placeholder="UserId" name="a1" required>
<input type="password" placeholder="Password" name="a2" required>
<a href="register.html" >Register</a>
<input type="submit" text="Login">
</form>
</body>
</html>
```

Python code: #!C:\Users\User\AppData\Local\Programs\Python\Python38\python.exe

```
import cgi

import mysql.connector
db=mysql.connector.connect(host="localhost",user="root",passwd="",db="reserve")
cur=db.cursor()

f=cgi.FieldStorage()
a=f.getvalue('a1')
b=f.getvalue('a2')
cur.execute("select * from log")
q=cur.fetchall()

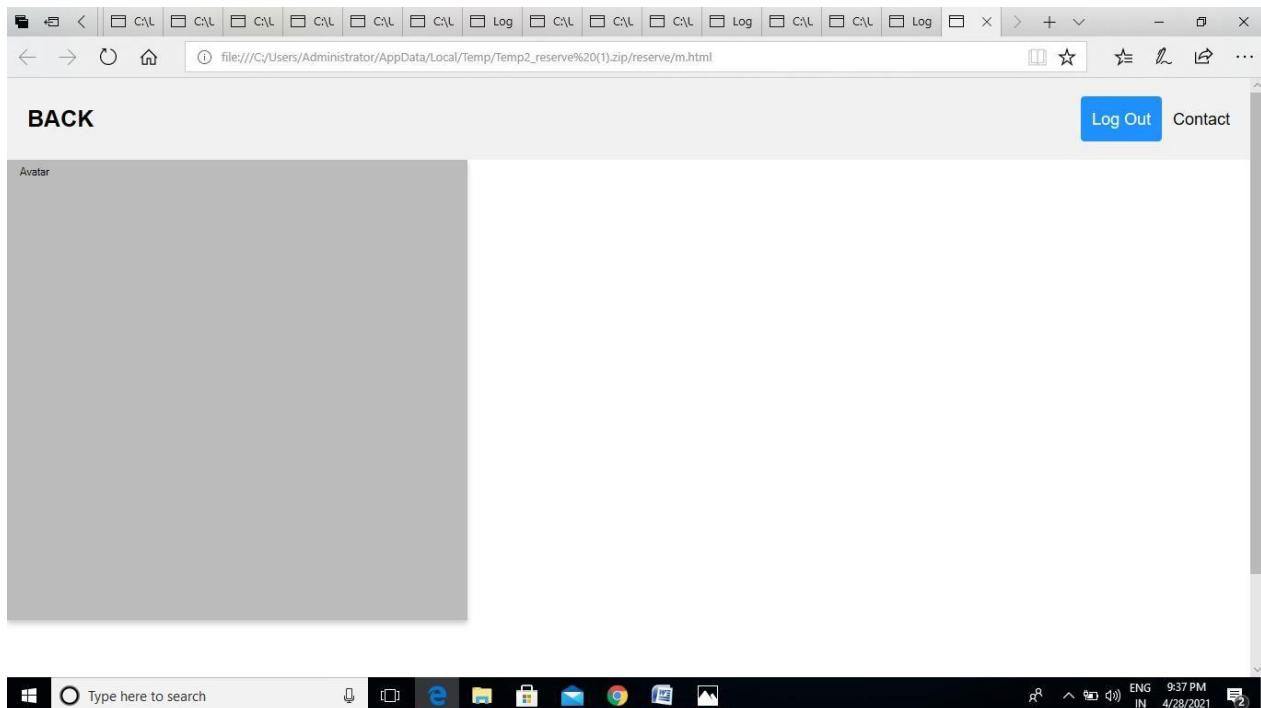
y=0

for i in q:
    if i[0]==a and i[1]==b:
        y=1
        break

else:
    y=0

if y==1:
    print("location:userpanel.html?msg=done?id=a\r\n\r\n")
else:
    print("location:login.html?msg=done\r\n\r\n")
```

## MESSAGE PAGE



HTML code:

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
* {box-sizing: border-box;}
```

```
body { margin:
0;
font-family: Arial, Helvetica, sans-serif;
}
```

```
.bg-img {
/* The image used */
background-image:
url("sh.jpg");
```

```
min-height: 770px;
```

```
/* Center and scale the image nicely */
background-
position: center; background-repeat: no-repeat;
background-size: cover;
position: relative;
```

```
}
```

```
.header { overflow: hidden;
background-color: #f1f1f1; padding:
20px 10px;
```

```
}

.header a { float:
left; color: black;
text-align: center;
padding: 12px;
text-decoration: none; font-size:
18px;
line-height: 25px;
border-radius: 4px;

}

.header a.logo { font-size:
25px; font-weight: bold;
}

.header a:hover { background-color:
#ddd; color: black;
}

.header a.active {
background-color: dodgerblue; color: white;
}

.header-right { float:
right;
}

@media screen and (max-width: 500px) {
.header a { float:
none;
display: block; text-
align: left;
}

.header-right { float:
none;
}

}

</style>
<style>
body {
font-family: Arial, Helvetica, sans-serif;
}
```

```

.flip-card {
  background-color: transparent; width: 500px;
  height: 500px; perspective:
  1000px;
}

.flip-card-inner { position:
  relative; width: 100%;
  height: 100%; text-align: center; transition: transform
  0.6s;
  transform-style: preserve-3d;
  box-shadow: 0 4px 8px 0 rgba(0,0,0,0.2);
}

.flip-card:hover .flip-card-inner { transform:
  rotateY(180deg);
}

.flip-card-front, .flip-card-back { position:
  absolute;
  width: 100%;
  height: 100%; -webkit-backface-visibility: hidden; backface-
  visibility: hidden;
}

.flip-card-front { background-color:
  #bbb; color: black;
}

.flip-card-back {
  background-color: #2980b9;
  color: white;
  transform: rotateY(180deg);
}

</style>

```

```

</head>
<body>
<div class="bg-img">
<div class="header">
<a href="userpanel.html" class="logo">BACK</a>
<div class="header-right">

```

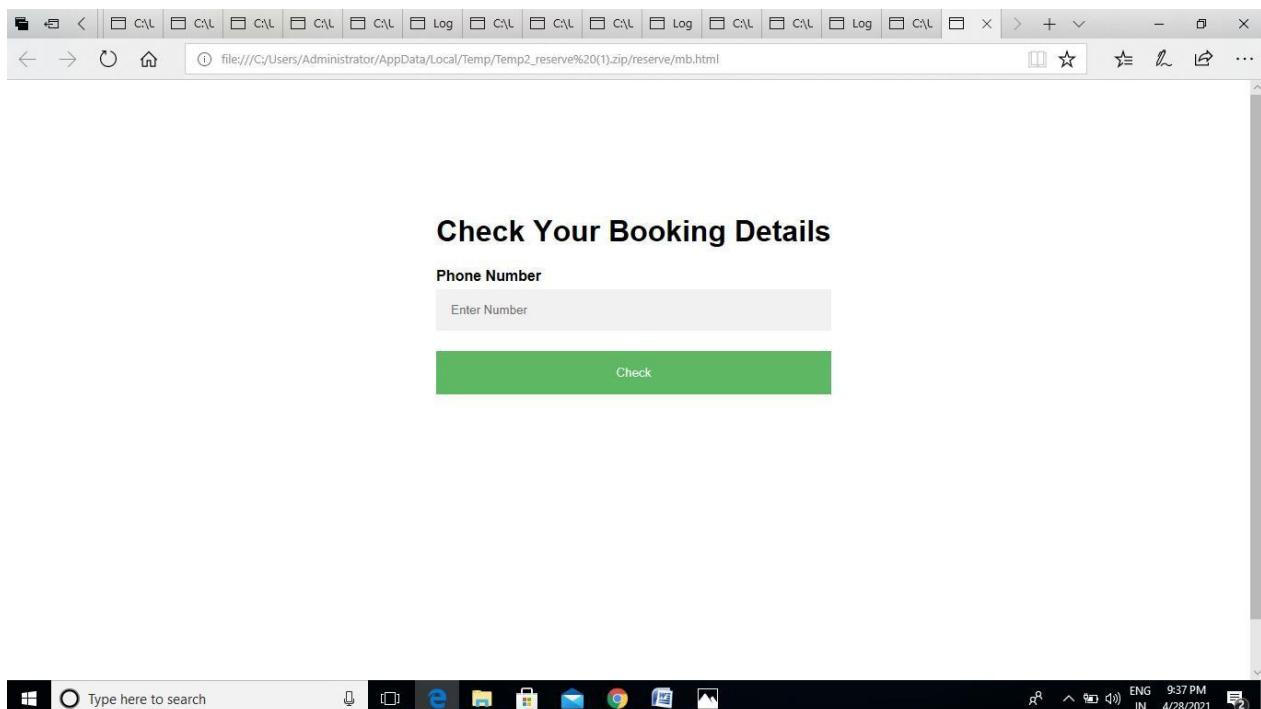
```
<a class="active" href="home.html">Log Out</a>
<a href="comp.html">Contact</a>

</div>
</div>

<div class="flip-card">
<div class="flip-card-inner">
<div class="flip-card-front">

</div>
<div class="flip-card-back">
<h1>Message</h1>
<p>Your Table Request Has Been Sent To The Restaurant</p>
<p>Kindly Wait For The Confirmation</p>
</div>
</div>
</div>
</div>
</body>
</html>
```

## BOOKING DETAILS



HTML code:

```
<!DOCTYPE html>

<html>
<head>

<meta name="viewport" content="width=device-width, initial-scale=1">

<style> body,
html {
    height: 100%;
    font-family: Arial, Helvetica, sans-serif;
}

* {
    box-sizing: border-box;
}

.bg-img {
    /* The image used */
    background-image:
        url("wa.jpg");
    min-height: 700px;
    /* Center and scale the image nicely */
    position: center;
    background-repeat: no-repeat;
    background-size: cover;
    position: relative;
}
```

```

/* Add styles to the form container */

.container { position:
    absolute; top: 14%;
    left: 33%;

    max-width: 500px; padding: 18px;
    background-color: white;

    /* Full-width input fields */ input[type=text],
    input[type=password] { width: 100%;

        padding: 15px; margin: 5px 0
        22px 0; border: none;
        background: #f1f1f1;

    }

    input[type=text]:focus, input[type=password]:focus { background-color: #ddd;
        outline: none;
    }
}

/* Set a style for the submit button */

.btn {
    background-color: #4CAF50; color:
    white;
    padding: 16px 20px; border:
    none; cursor: pointer; width:
    100%;

    opacity: 0.9;

    .btn:hover {
        opacity: 1;
    }
}

</style>
</head>
<body>

<div class="bg-img">
<form action="mb.py" class="container">
    <h1>Check Your Booking Details</h1>

    <label for="email"><b>Phone Number</b></label>
    <input type="text" placeholder="Enter Number" name="a1" required>
    <button type="submit" class="btn">Check</button>
</form>
</div>

</body>
</html>

```

```
Python code: #!C:\Users\User\AppData\Local\Programs\Python\Python38\python.exe print("content-
type:text/html\r\n\r\n")

import cgi

import mysql.connector db=mysql.connector.connect(host="localhost",user="root",passwd="",db="reserve")
cur=db.cursor()
f=cgi.FieldStorage()
p=f.getvalue('a1')
cur.execute("select * from bs where phno=%s%(p)) t=cur.fetchall()

print("""
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
table {
    border-collapse: collapse; border-
    spacing: 0;
    width: 100%;
    border: 1px solid #ddd;
}
.bg-img {
/* The image used */ background-image:
url("sh.jpg");

min-height: 770px;

/* Center and scale the image nicely */ background-
position: center; background-repeat: no-repeat;
background-size: cover;
position: relative;
}

th, td {
    text-align: left;
    padding: 8px;
}
tr:nth-child(even){background-color: #f2f2f2} tr:nth-
child(odd){background-color: #ffffcc} th:nth-
child(even){background-color: #ffffff} th:nth-
child(odd){background-color: #ffffff}

</style>
</head>
<body>

<div class="bg-img">
```

```
<h2 style="background-color:LightGray;"><a href="userpanel.html">Back</a><center>Booking Details</center></h2>

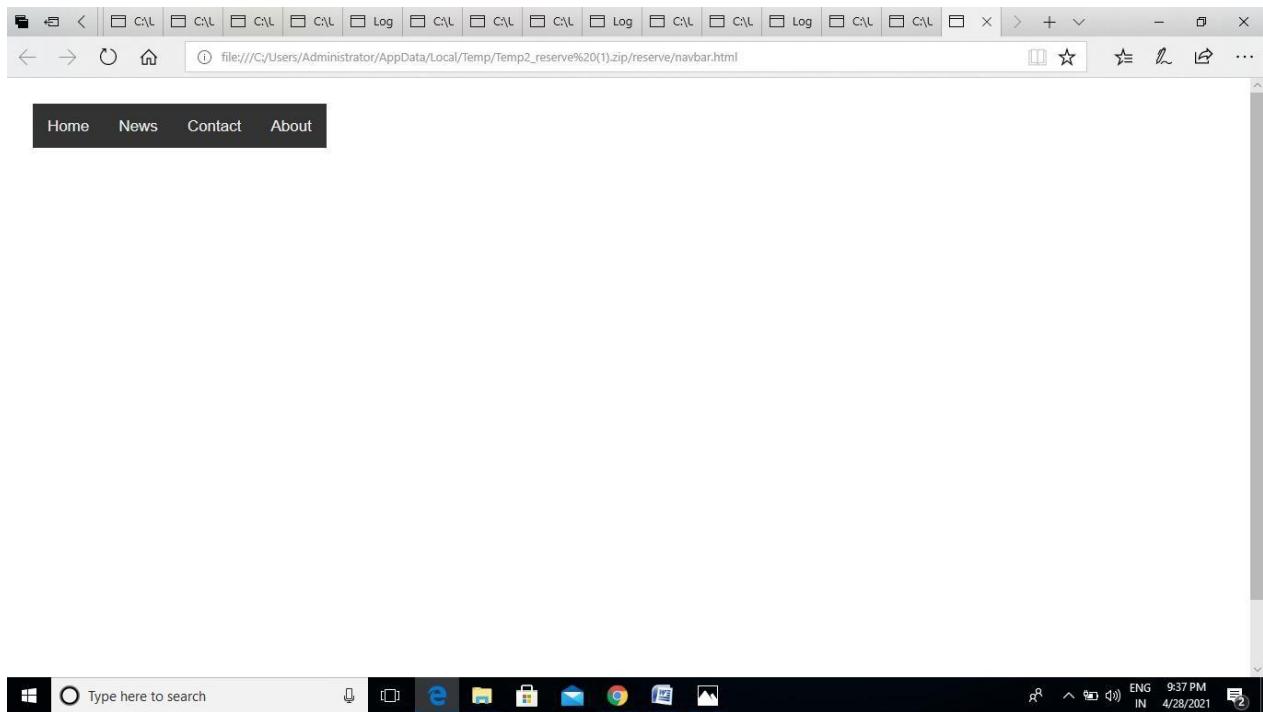
<div style="overflow-x:auto;">
<table>
<tr>
<th>Table Number</th>
<th>Restaurant Name</th>
<th>Date</th>
<th>Guests</th>
<th>Time</th>
<th>Email-Id</th>
<th>Phone Number</th>

</tr>
</div>
""")
for i in t:
    print(
"""

<tr><td>%d</td><td>%s</td><td>%s</td><td>%s</td><td>%s</td><td>%s</td><td>%s</td></tr>"" "%(i[0],i[6],i[1],i[2],i[3],i[4],i[5]))

print("""
</table>
</body>
</html> """)
```

## NAVIGATION BAR



HTML code:

```
<!DOCTYPE html>

<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
body {font-family: Arial, Helvetica, sans-serif;}
* {box-sizing: border-box;}
```

```
.bg-img {
/* The image used */
background-image:
url("D1.jpg");
min-height: 715px;

/* Center and scale the image nicely */
background-position: center; background-repeat: no-repeat;
background-size: cover;

/* Needed to position the navbar */
position: relative;
}
```

```
/* Position the navbar container inside the image */
.container {
position:
absolute; margin: 20px;
width: auto;
```

```
}

/* The navbar */
.topnav { overflow: hidden;
background-color: #333;
}

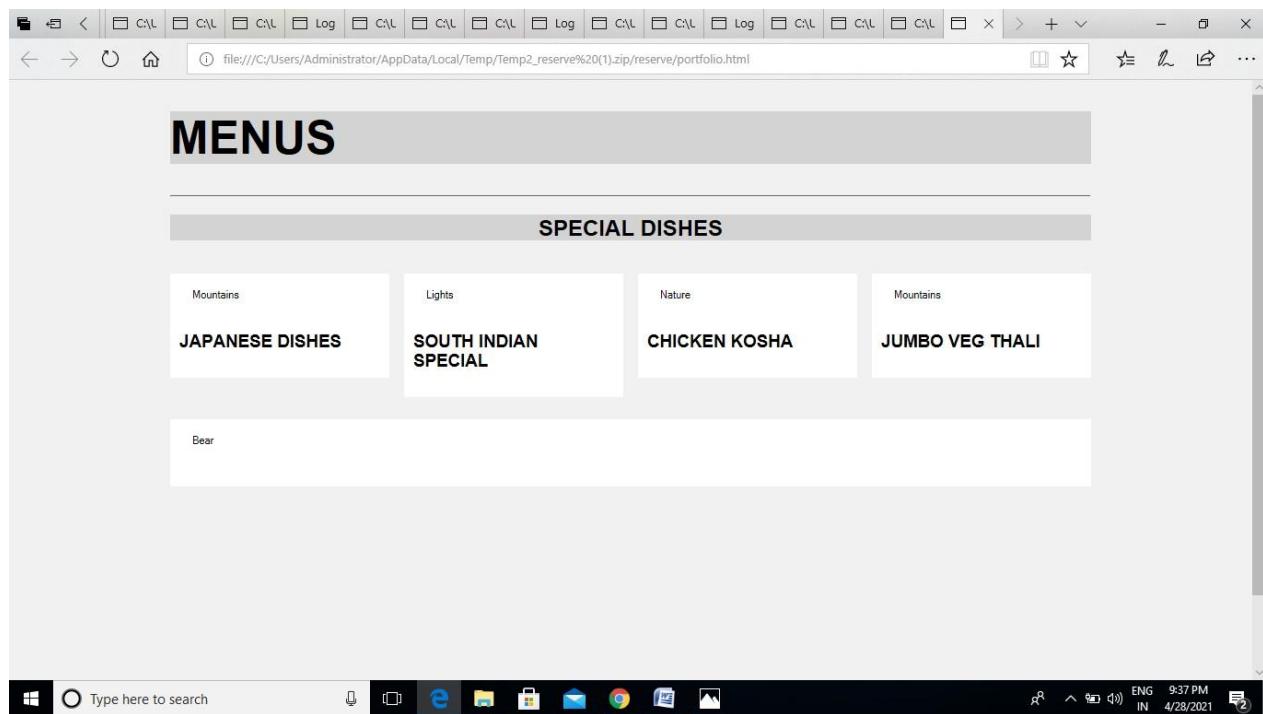
/* Navbar links */
.topnav a { float: left;
color: #f2f2f2;
text-align: center; padding: 14px
16px; text-decoration: none;
font-size: 17px;
}

.topnav a:hover { background-color:
#ddd; color: black;
}

</style>
</head>
<body>
<div class="bg-img">
<div class="container">
<div class="topnav">
<a href="#home">Home</a>
<a href="#news">News</a>
<a href="#contact">Contact</a>
<a href="#about">About</a>
</div>
</div>
</div>

</body>
</html>
```

RESTAURANT MENU



HTML code:

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
* {
    box-sizing: border-box;
}

body {
    background-color: #f1f1f1;
    padding: 0px; font-family:
    Arial;
}

.bg-img {
    /* The image used */
    background-image:
    url("sh.jpg");

    min-height: 700px;
}
```

```
/* Center and scale the image nicely */
background-position: center; background-repeat: no-repeat;
background-size: cover;
position: relative;
```

```
}

/* Center website */
.main {
    max-width: 1000px; margin:
    auto;
}

h1 {
    font-size: 50px;
    word-break: break-all;
}

.row {
    margin: 8px -16px;
}

/* Add padding BETWEEN each column */
.row,
.row > .column {
    padding: 8px;
}

/* Create four equal columns that floats next to each other */
.column { float:
    left; width: 25%; }

/* Clear floats after rows */
.row:after { content:
    ""; display: table;
    clear: both;
}

/* Content */
.content {
    background-color: white; padding:
    10px;
}

/* Responsive layout - makes a two column-layout instead of four columns */
@media screen and (max-width:
900px) {
    .column { width:
    50%; }
```

```

}

/* Responsive layout - makes the two columns stack on top of each other instead of next to each other
*/
@media screen and (max-width: 600px) {

.column { width:
100%;

}

}

</style>
</head>
<body>

<div class="bg-img">
<!-- MAIN (Center website) --&gt;
&lt;div class="main"&gt;

&lt;h1 style="background-color:LightGray;">MENUS</h1>
<hr>

<Center><h2 style="background-color:LightGray;">SPECIAL DISHES</h2></Center>
<p></p>

<!-- Portfolio Gallery Grid --&gt;
&lt;div class="row"&gt;
&lt;div class="column"&gt;
&lt;div class="content"&gt;
&lt;img src="a1.jpg" alt="Mountains" style="width:100%"&gt;
&lt;h3&gt;JAPANESE DISHES&lt;/h3&gt;
&lt;p&gt;&lt;/p&gt;
&lt;/div&gt;
&lt;/div&gt;
&lt;div class="column"&gt;
&lt;div class="content"&gt;
&lt;img src="a2.jpg" alt="Lights" style="width:100%"&gt;
&lt;h3&gt;SOUTH INDIAN SPECIAL&lt;/h3&gt;
&lt;p&gt;&lt;/p&gt;
&lt;/div&gt;
&lt;/div&gt;
&lt;div class="column"&gt;
&lt;div class="content"&gt;
</pre>

```

```

<h3>CHICKEN KOSHA</h3>
<p></p>
</div>
</div>
<div class="column">
<div class="content">

<h3>JUMBO VEG THALI</h3>
<p></p>
</div>
</div>
<!-- END GRID -->
</div>

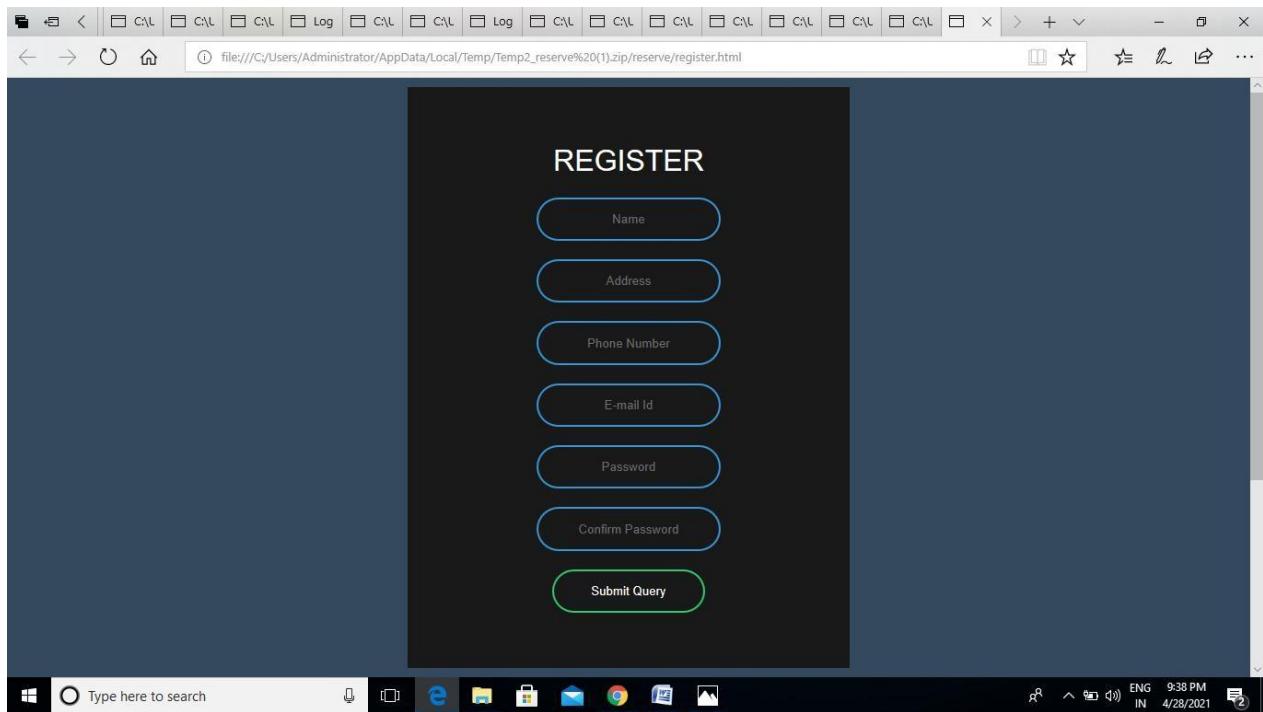
<div class="content">

<h3></h3>
<p></p>
<p></p>
</div>

<!-- END MAIN -->
</div>
</div>

</body>
</html>
```

## RESTAURANT REGISTER



HTML code:

```
<html>
<head>
<meta charset="utf-8" />
<title>Register Form</title>
<link href="StyleSheet2.css" rel="stylesheet" type="text/css" />
</head>
<body>
<style>
body
{
    margin: 0;
    padding: 0;
    font-family: Sans-Serif; background: #34495e;
}
.box
{
    width: 400px; padding:
    40px; position: absolute;
    top: 50%;
    left: 50%;
    transform: translate(-50%,-50%); background:
    #191919;
    text-align:center;
}
```

```
.box h1
{
    color:White;
    text-transform:uppercase; font-weight:
    500;
}

.box input[type="text"],.box input[type="password"],.box input[type="email"]
{
    border:0; background: none;
    display:block;
    margin: 20px auto; text-
    align:center;
    border: 2px solid #3498db; padding:
    14px 10px; width: 200px;
    outline: none; color:
    White;
    border-radius: 24px;
    transition:0.25s;
}

.box input[type="text"]:focus,.box input[type="password"]:focus,.box input[type="email"]:focus
{
    width: 280px;
    border-color: #2ecc71;
}

.box input[type="submit"]
{
    border:0; background:
    none; display:block;
    margin: 20px auto; text-
    align:center;
    border: 2px solid #2ecc71; padding:
    14px 40px; outline: none;
    color: White;
    border-radius: 24px;
    transition:0.25s;
    cursor:pointer;
}

.box input[type="submit"]:hover
{
    background:#2ecc71;
}

</style>

<form class="box" action="register.py">
<h1>Register</h1>
```

```
<input type="text" placeholder="Name" name="a1" required>
<input type="text" placeholder="Address" name="a2" required>
<input type="text" placeholder="Phone Number" name="a3" required>
<input type="email" placeholder="E-mail Id" name="a4" required >
<input type="password" placeholder="Password" name="a5" required>
<input type="password" placeholder="Confirm Password" name="a6" required >
<input type="submit" text="Submit">
</form>
</body>
</html>
```

```
Python code: #!C:\Users\User\AppData\Local\Programs\Python\Python38\python.exe import cgi
import mysql.connector
db=mysql.connector.connect(host="localhost",user="root",passwd="",db="reserve")
cur=db.cursor()
t=cgi.FieldStorage()
a=t.getvalue('a1')
b=t.getvalue('a2')
c=t.getvalue('a3')
d=t.getvalue('a4')
e=t.getvalue('a5')
cur.execute("insert into user (name,addr,phno,email,pwd) values('%s','%s','%s','%s','%s')%(a,b,c,d,e)) cur.execute("insert into log (email,pwd)
values('%s','%s')%(d,e))
db.commit() print("location:sr.html?msg=done\r\n\r\n")
```

## RESTAURANTS

The screenshot shows a web browser window with two side-by-side restaurant profiles. The left profile is for "India Restaurant" and the right is for "OCTA". Both profiles include sections for "Avatar", "Location", "Timings", and buttons for "Book a Table" and "View Details". The browser toolbar at the top includes icons for file operations like Open, Save, Print, and a search bar with the URL "file:///C:/Users/Administrator/AppData/Local/Temp/Temp2\_reserve%20(1).zip/reserve/restaurant.html". The taskbar at the bottom shows various pinned app icons.

HTML code:

```
html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
body {
```

```
    font-family: Arial;
```

```
}
```

```
.coupon {
    border: 5px dotted #bbb; width:
    80%; border-radius: 15px; float:
    left; margin: 100 auto; max-
    width: 600px;
}
```

```
.bg-img {
/* The image used */ background-image:
url("sh.jpg");
min-height: 700px;
```

```
/* Center and scale the image nicely */ background-
position: center; background-repeat: no-repeat;
background-size: cover;
```

```
position: relative;  
}  
  
.container { padding: 2px  
16px;  
background-color: #f1f1f1;  
}  
  
.promo { background: #ccc;  
padding: 3px;  
}  
  
.expire { color:  
red;  
}  
</style>  
<style>  
body {  
font-family: Arial;  
}  
  
.coupons {  
border: 5px dotted #bbb; width:  
80%;  
border-radius: 15px; float:  
right;  
margin: 0 auto; max-width:  
600px;  
}  
  
.container { padding: 2px  
16px;  
background-color: #f1f1f1;  
}  
.bg-img {  
/* The image used */ background-image:  
url("sh.jpg");  
  
min-height: 700px;  
  
/* Center and scale the image nicely */ background-  
position: center; background-repeat: no-repeat;  
background-size: cover;  
position: relative;  
}
```

```
.promo { background: #ccc;
padding: 3px;
}

.expire { color:
red;
}

</style>
<style>
* {box-sizing: border-box;}
body { margin:
0;
font-family: Arial, Helvetica, sans-serif;
}

.header { overflow: hidden;
background-color: #f1f1f1; padding:
20px 10px;
}

.header a { float:
left; color: black;
text-align: center;
padding: 12px;
text-decoration: none; font-size:
18px;
line-height: 25px; border-
radius: 4px;
}

.header a.logo { font-size:
25px; font-weight: bold;
}

.header a:hover { background-color:
#ddd; color: black;
}

.header a.active {
background-color: dodgerblue; color: white;
}

.header-right { float:
right;
}
```

```

@media screen and (max-width: 500px) {

.header a { float: none;
    display: block; text-
    align: left;
}

.header-right { float:
    none;
}
}

</style>
</head>
<body>

<div class="header">
<a href="userpanel.html" class="logo">RESTAURANTS</a>
<div class="header-right">
<a class="active" href="home.html">HOME</a>
<a href="comp.html">Contact</a>

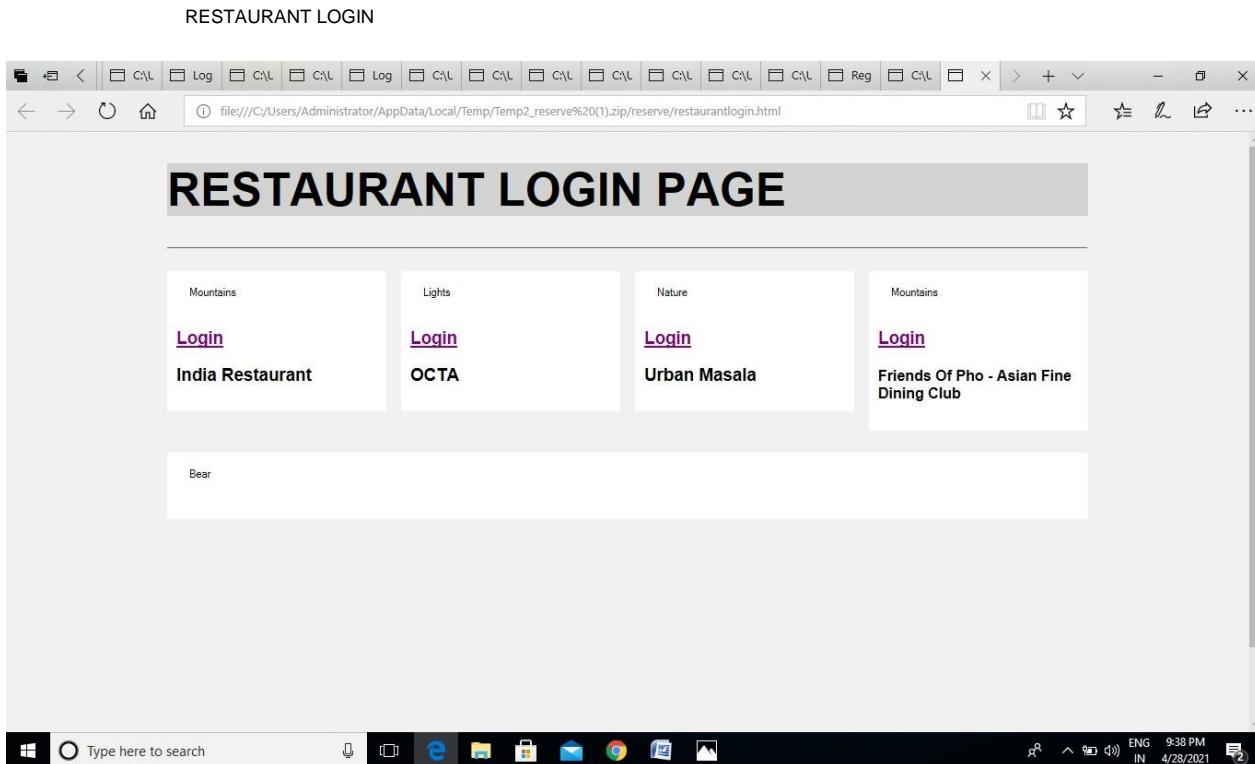
</div>
</div>

<div class="bg-img">
<div class="coupon">
<div class="container">
<h2>India Restaurant</h2>
</div>
<a href="booktable.html" ></a>
<div class="container" style="background-color:white">
<h3><b>Location</b></h3>
<p>33,Level 5, Quest Mall , Syed Amir Ali Avenue , Ballygunge , Kolkata</p>
<h3><b>Timings</b></h3>
<p>Mon-Fri 12noon - 11pm</p>
</div>
<div class="container">
<a href="booktablea.html" ><p> <span class="promo">Book a Table</span></p></a>
<a href="portfolio.html"><p class="expire">View Details</p></a>
</div>
</div>
<div class="coupons">
<div class="container">
<h3> OCTA </h3>
</div>
<a href="booktable.html" ></a>
<div class="container" style="background-color:white">

```

```
<h3><b>Location</b></h3>
<p>34, Karl Marx Sarani , Kidderpore , Kolkata</p>
<h3><b>Timings</b></h3>
<p>Mon-Sat 12noon - 11pm</p>
</div>
<div class="container">
<a href="booktableb.html"><p> <span class="promo">Book a Table</span></p></a>
<a href="portfolioib.html"><p class="expire">View Details</p></a>
</div>
</div>
<div>
<div class="bg-img">
<div class="coupons">
<div class="container">
<h3>Friends Of Pho - Asian Fine Dining Club</h3>
</div>
<a href="booktable.html"></a>
<div class="container" style="background-color:white">
<h3><b>Location</b></h3>
<p>196 ,Chakraborty Para , Rajdanga , Near Water Tank, Kasba, Kolkata</p>
<h3><b>Timings --: Mon-Fri 12noon - 11pm</b></h3>
</div>
<div class="container">
<a href="booktabled.html"><p> <span class="promo">Book a Table</span></p></a>
<a href="portfoliod.html"><p class="expire">View Details</p></a>
</div>
</div>
<div class="coupon">
<div class="container">
<h3> Urban Masala</h3>
</div>
<a href="booktable.html"></a>
<div class="container" style="background-color:white">
<h3><b>Location</b></h3>
<p>33,Level 5, Quest Mall , Syed Amir Ali Avenue , Ballygunge , Kolkata</p>
<h3><b>Timings</b></h3>
<p>Mon-Fri 12noon - 11pm</p>
</div>
<div class="container">
<a href="booktablec.html"><p> <span class="promo">Book a Table</span></p></a>
<a href="portfolioc.html"><p class="expire">View Details</p></a>
</div>
```

```
</div>
</div>
</body>
</html>
```



HTML code:

```
DOC<!TYPE html>

<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
* {
  box-sizing: border-box;
}

body {
  background-color: #f1f1f1;
  padding: 0px; font-family:
  Arial;
}

.bg-img {
/* The image used */
background-image:
url("sh.jpg");

min-height: 700px;

/* Center and scale the image nicely */
background-
position: center; background-repeat: no-repeat;
background-size: cover;
position: relative;
```

```
}

.promo { background: #ccc;
padding: 3px;
}

/* Center website */

.main {
max-width: 1000px; margin:
auto;
}

h1 {
font-size: 50px;
word-break: break-all;
}

.row {
margin: 8px -16px;
}

/* Add padding BETWEEN each column */

.row,
.row > .column {
padding: 8px;
}

/* Create four equal columns that floats next to each other */

.column { float:
left; width: 25%; }

/* Clear floats after rows */

.row:after {
content: ""; display:
table; clear: both;
}

/* Content */

.content {
background-color: white; padding:
10px;
}

/* Responsive layout - makes a two column-layout instead of four columns */ @media screen and (max-width:
900px) {
.column { width:
```

```

50%;

}

}

/* Responsive layout - makes the two columns stack on top of each other instead of next to each other
*/
@media screen and (max-width: 600px) {

.column { width:

100%;

}

}

</style>
</head>

<body>

<div class="bg-img">

<!-- MAIN (Center website) -->

<div class="main">

<h1 style="background-color:LightGray;">RESTAURANT LOGIN PAGE</h1>

<hr>

<!-- Portfolio Gallery Grid -->

<div class="row">

<div class="column">

<div class="content">



<a href="alogin.html"><h3>Login</h3></a>

<h3>India Restaurant</h3>

</div>

</div>

<div class="column">

<div class="content">



<a href="blogin.html"><h3>Login</h3></a>

<h3>OCTA</h3>

</div>

<div class="column">

<div class="content">



<a href="clogin.html"><h3>Login</h3></a>

<h3>Urban Masala</h3>

</div>

</div>

<div class="column">

```

```
<div class="content">

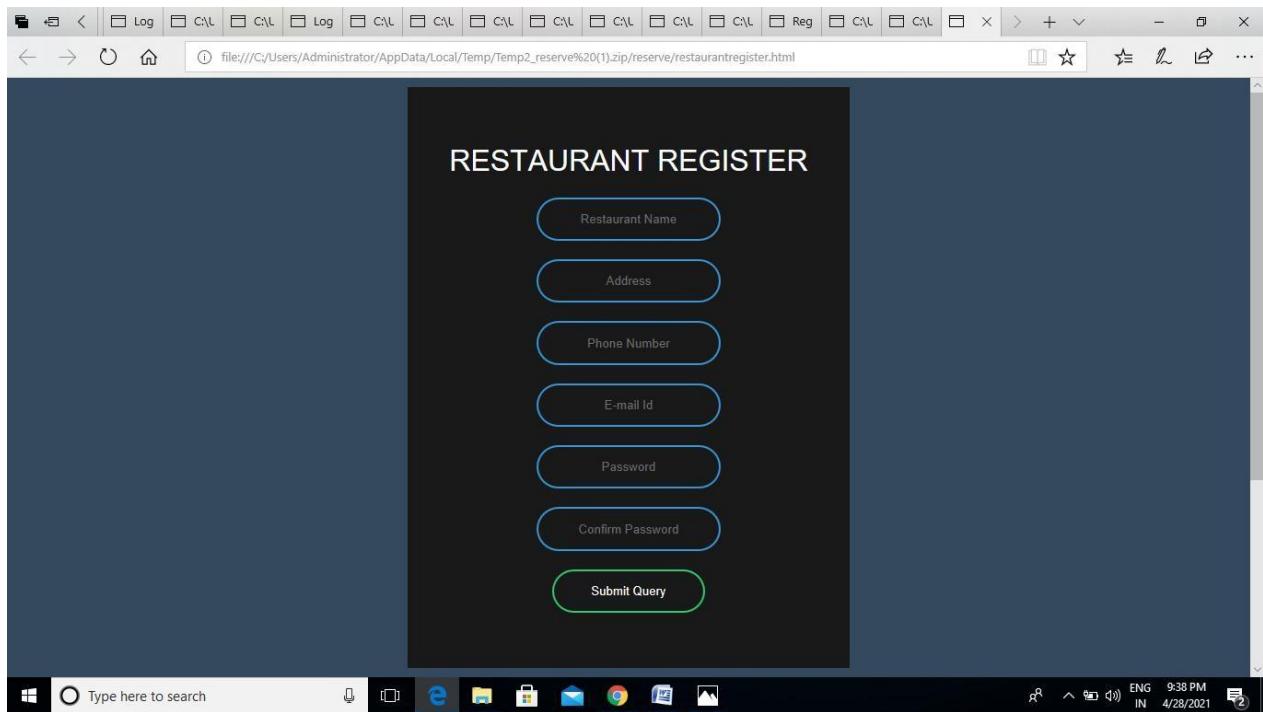
<a href="dlogin.html"><h3>Login</h3></a>
<h4>Friends Of Pho - Asian Fine Dining Club</h4>
</div>
</div>

<!-- END GRID -->
</div>

<div class="content">

<h3></h3>
<p></p>
<p></p>
</div>
</div>
</div>
</body>
</html>
```

## RESTAURANT REGISTER



HTML code:

```
<html>
<head>
<meta charset="utf-8" />
<title>Register Form</title>
<link href="StyleSheet2.css" rel="stylesheet" type="text/css" />
</head>
<body>
<style>
body
{
    margin: 0;
    padding: 0;
    font-family: Sans-Serif; background: #34495e;
}
.box
{
    width: 400px; padding:
    40px; position: absolute;
    top: 50%;
    left: 50%;
    transform: translate(-50%,-50%); background:
    #191919;
    text-align:center;
}
```

```

.box h1
{
    color:White;
    text-transform:uppercase; font-weight:
    500;
}

.box input[type="text"],.box input[type="password"],.box input[type="email"]
{
    border:0; background: none;
    display:block;
    margin: 20px auto; text-
    align:center;
    border: 2px solid #3498db; padding:
    14px 10px; width: 200px;
    outline: none; color:
    White;
    border-radius: 24px;
    transition:0.25s;
}

.box input[type="text"]:focus,.box input[type="password"]:focus,.box input[type="email"]:focus
{
    width: 280px;
    border-color: #2ecc71;
}

.box input[type="submit"]
{
    border:0; background:
    none; display:block;
    margin: 20px auto; text-
    align:center;
    border: 2px solid #2ecc71; padding:
    14px 40px; outline: none;
    color: White;
    border-radius: 24px;
    transition:0.25s;
    cursor:pointer;
}

.box input[type="submit"]:hover
{
    background:#2ecc71;
}

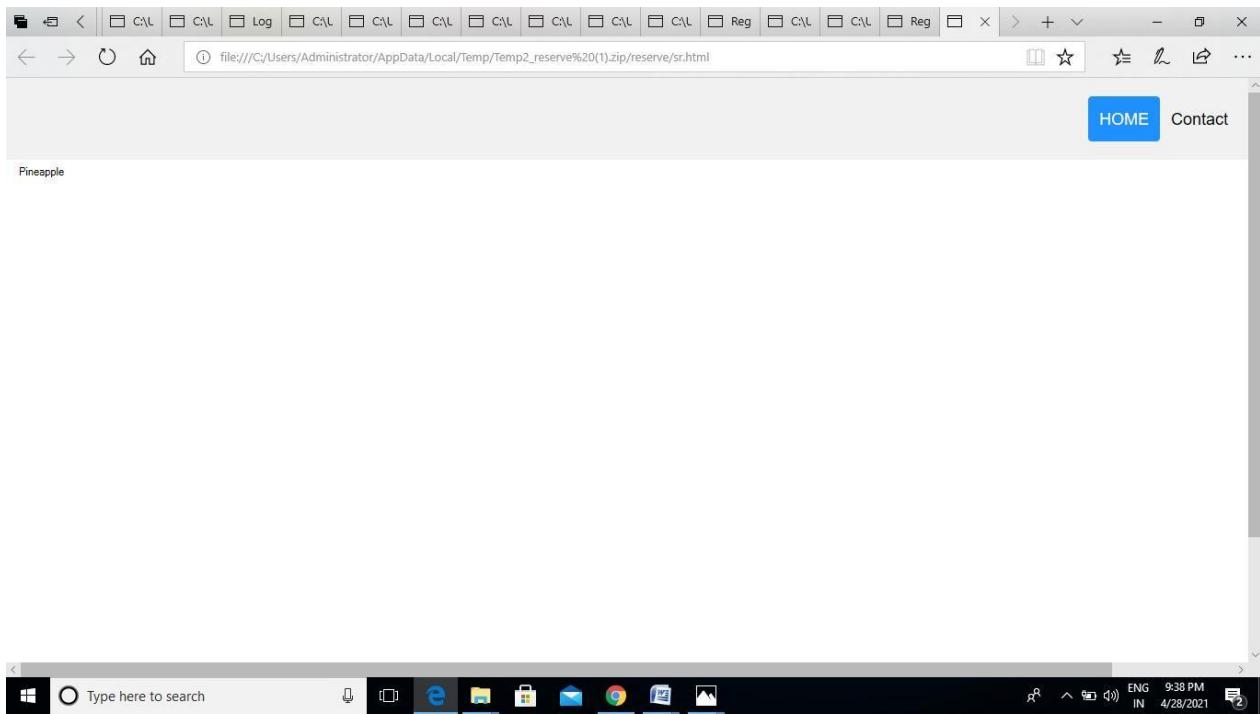
</style>

<form class="box" action="rest.py">
<h1> Restaurant Register</h1>

```

```
<input type="text" placeholder="Restaurant Name" name="a1" required>
<input type="text" placeholder="Address" name="a2" required>
<input type="text" placeholder="Phone Number" name="a3" required>
<input type="email" placeholder="E-mail Id" name="a4" required >
<input type="password" placeholder="Password" name="a5" required>
<input type="password" placeholder="Confirm Password" name="a6" required >
<input type="submit" text="Submit">
</form>
</body>
</html>
```

## SUCCESSFULLY REGISTERED



HTML code:

```
<!DOCTYPE html>

<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
img:hover {
    animation: shake 0.5s;
    animation-iteration-count: infinite;
}

@keyframes shake {
    0% { transform: translate(1px, 1px) rotate(0deg); } 10% { transform:
    translate(-1px, -2px) rotate(-1deg); }
    20% { transform: translate(-3px, 0px) rotate(1deg); } 30% { transform:
    translate(3px, 2px) rotate(0deg); } 40% { transform: translate(1px, -1px)
    rotate(1deg); } 50% { transform: translate(-1px, 2px) rotate(-1deg); } 60% {
    transform: translate(-3px, 1px) rotate(0deg); } 70% { transform: translate(3px,
    1px) rotate(-1deg); } 80% { transform: translate(-1px, -1px) rotate(1deg); }
    90% { transform: translate(1px, 2px) rotate(0deg); } 100% { transform:
    translate(1px, -2px) rotate(-1deg); }

}
</style>
<style>
* {box-sizing: border-box;}
```

```
body { margin:  
0;  
font-family: Arial, Helvetica, sans-serif;  
}
```

```
.header { overflow: hidden;  
background-color: #f1f1f1; padding:  
20px 10px;  
}  
.header a { float:  
left; color: black;  
text-align: center;  
padding: 12px;  
text-decoration: none; font-size:  
18px;  
line-height: 25px; border-  
radius: 4px;  
}
```

```
.header a.logo { font-size:  
25px; font-weight: bold;  
}
```

```
.header a:hover { background-color:  
#ddd; color: black;  
}
```

```
.header a.active {  
background-color: dodgerblue; color: white;  
}
```

```
.header-right { float:  
right;  
}
```

```
@media screen and (max-width: 500px) {  
.header a { float: none;  
display: block; text-  
align: left;  
}
```

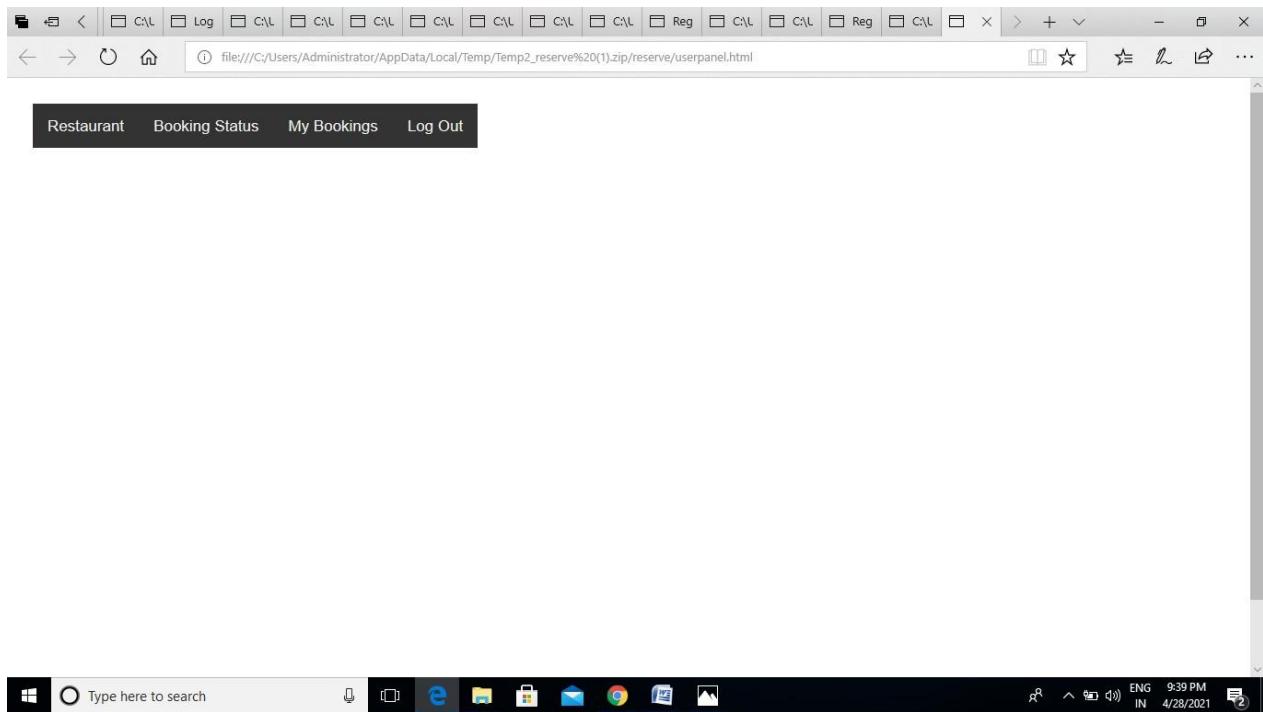
```
.header-right { float:  
none;  
}  
}
```

```
</style>
</head>
<body>
<div class="header">
<a href="login.html" class="logo"></a>
<div class="header-right">
<a class="active" href="home.html">HOME</a>
<a href="comp.html">Contact</a>
</div>
</div>



</body>
</html>
```

## USER PANEL



HTML code:

```
<!DOCTYPE html>

<html>

<head>

<meta name="viewport" content="width=device-width, initial-scale=1">

<style>

body {font-family: Arial, Helvetica, sans-serif;}

* {box-sizing: border-box;}
```

```
.bg-img {

/* The image used */ background-image:
url("ad.jpg");

min-height: 715px;
```

```
/* Center and scale the image nicely */ background-
position: center; background-repeat: no-repeat;
background-size: cover;
```

```
/* Needed to position the navbar */ position: relative;
```

```
}
```

```
/* Position the navbar container inside the image */

.container { position:
absolute; margin: 20px;
width: auto;}
```

```
/* The navbar */
.topnav { overflow: hidden;
background-color: #333;
}

/* Navbar links */
.topnav a { float: left;
color: #f2f2f2;
text-align: center; padding: 14px
16px; text-decoration: none;
font-size: 17px;
}

.topnav a:hover { background-color:
#ddd; color: black;
}

</style>
</head>
<body>
<div class="bg-img">
<div class="container">
<div class="topnav">
<a href="restaurant.html">Restaurant</a>
<a href="bs.html">Booking Status</a>
<a href="mb.html">My Bookings</a>
<a href="home.html">Log Out</a>
</div>
</div>
</div>

</body>
</html>
```

## VIEW REQUEST



### Responsive Table

If you have a table that is too wide, you can add a container element with overflow-x:auto around the table, and it will display a horizontal scroll bar when needed.

Resize the browser window to see the effect. Try to remove the div element and see what happens to the table.

YE VALA NORMAL PRINT VALA SECTION ME HOGA FETCH VALA SECTION ME LIKHNA HOGA YE VALA SECTION %S DE DEKE %S

First Name	Last Name	Points									
Jill	Smith	50	50	50	50	50	50	50	50	50	50
Eve	Jackson	94	94	94	94	94	94	94	94	94	94
Adam	Johnson	67	67	67	67	67	67	67	67	67	67



HTML code:

```
!DOCTYPE html>

<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
table {
    border-collapse: collapse; border-
    spacing: 0;
    width: 100%;

    border: 1px solid #ddd;
}

th, td {
    text-align: left;
    padding: 8px;
}

tr:nth-child(even){background-color: #f2f2f2}

</style>
</head>
<body>

<h2>Responsive Table</h2>
<p>If you have a table that is too wide, you can add a container element with overflow-x:auto around the table, and it will display a horizontal scroll bar when needed.</p>
```

<p>Resize the browser window to see the effect. Try to remove the div element and see what happens to the table.</p>

First Name	Last Name	Points	Points	Points	Points	Points	Points
Jill	Smith	50	50	50	50	50	50
Eve	Jackson	94	94	94	94	94	94

```

<td>94</td>
</tr>
<tr>
<td>Adam</td>
<td>Johnson</td>
<td>67</td>
<td>67</td>
<td>67</td>
<td>67</td>
<td>67</td>
<td>67</td>
<td>67</td>
<td>67</td>
<td><a href="restaurantregister.html" >67</a></td>
</tr>
</table>
</div>
</body>
</html>

```

Python code:

```

#!/C:/Users/user1/AppData/Local/Programs/Python/Python38/pythonw.exe import cgi
import mysql.connector
db=mysql.connector.connect(host="localhost",user="root",passwd="",db="rese rve")
cur=db.cursor()
t=cgi.FieldStorage()
a=t.getvalue('a1')
b=t.getvalue('a2')
c=t.getvalue('a3')
d=t.getvalue('a4')
e=t.getvalue('a5')
cur.execute("insert into table1 (day,people,time,email,phno)
values(%s,%s,%s,%s,%s)%(a,b,c,d,e)")
db.commit() print("location:m.html?msg=done\r\n\r\n")

```

# CONCLUSION

J  
F  
G  
J  
H  
C  
F  
G  
B  
F

This dissertation dealt with the design and development process of context-adaptive Web applications. It reviewed the state of the art in the fields of adaptive hypermedia and Web engineering, identified main shortcomings of existing solutions, and proposed a componentbased approach for engineering adaptive Web sites. For the systematic realization of adaptive Web presentations from reusable components, a model-based authoring process was designed and constructively validated based on a prototypically realized visual component authoring tool, an automatic transformation facility for the model-driven generation of adaptive component structures, as well as a number of example applications. Finally, it was shown how the lessons learned from engineering component-based adaptive Web sites can be generalized in order to add adaptation to existing (not component-based) Web applications.

# **FUTURE SCOPE**

In future we would like to keep working on this project and make new additions to provide users with more advanced features and more detailed information. \\ We have set our sights on the following additions in future:-

1. Online payment process through debit and credit cards.
2. Option to chat between user and restaurant