Tour & Stay

A windows application made by

Basudeb Paul Chowdhury

2024.07.31

TABLE OF CONTENT

ABSTRACT

Tour and Stay is a windows-based application. This application provides users an easy way to search and find any hotel for stay by only using our application. Also, Tour and Stay Provides a feature to book hotels where they want to stay by just downloading our application on your device.

The main objective of our application is facilitating users to provide an online searching and booking of hotels.

INTRODUCTION

Project Name: Tour and Stay

Description: Tour and Stay is a Windows-based application designed to revolutionize the way users search for and book hotels. This application allows users to conveniently book accommodations from the comfort of their homes. It features a comprehensive listing of franchisee hotels that have partnered with our platform. These hotels can update their profiles and showcase the facilities they offer, such as food, TV, WiFi, travel services, and more.

The core functionality of Tour and Stay revolves around facilitating a seamless interaction between users and hotels. Users can browse through various hotel options, view detailed information about the facilities offered, and make specific requests for their stay. Once a user finds a suitable hotel, they can send a booking request along with their desired facilities.

Hotels receive notifications of these booking requests and have the ability to approve or decline them based on the user's requirements and the hotel's availability. Users are then notified of the hotel's decision, ensuring a transparent and efficient booking process.

.

Project Objectives

User-Friendly Interface:

Develop a clean, intuitive, and user-friendly interface that simplifies the hotel booking process for users. Ensure easy navigation and accessibility of features for both users and hotel partners.

Comprehensive Hotel Listings:

Maintain an extensive database of franchisee hotels, each with detailed profiles including available facilities, room types, prices, and images. Allow hotels to update their profiles and facilities to provide accurate and up-to-date information.

Seamless Booking Process:

Enable users to search for hotels based on various criteria such as location, facilities, and price range. Allow users to make specific facility requests when booking a hotel.

Efficient Notification System:

Implement a robust notification system to inform hotels of new booking requests in real-time. Notify users of the status of their booking requests, including confirmations and declines.

Hotel Management Features:

Provide hotels with tools to manage booking requests, including the ability to approve or decline requests based on feasibility. Allow hotels to view user requirements and communicate decisions efficiently.

Feedback and Improvement:

Gather feedback from users and hotel partners to continually improve the application. Implement updates and new features based on user needs and technological advancements.

FEATURES

User Account Management:

- **Registration and Login:** Users can create accounts and log in securely to access personalized features.
- **Profile Management:** Users can update their personal information, contact details, and preferences.

Hotel Listings and Search:

- **Comprehensive Hotel Database:** Access an extensive list of franchisee hotels with detailed profiles.

Hotel Profiles:

- **Detailed Hotel Information:** View detailed descriptions of hotels, including available facilities, room types, prices, and images.
- **Facility Overview:** See a list of facilities offered by the hotel such as food services, TV, WiFi, travel services, and more.

Booking and Facility Requests:

- **Booking Requests:** Users can send booking requests to hotels specifying their desired dates and facilities.
- **Custom Facility Requests:** Users can request specific facilities they wish to have during their stay.

Notification System:

- **Real-Time Notifications:** Receive instant notifications for booking confirmations, declines, and other important updates.
- **Status Updates:** Stay informed about the status of booking requests and any changes made by the hotel.

USED TECHNOLOGIES

Frontend: C#.NET windows frontend features

Backend: C#.NET

Database: MySQL along with ADO.NET

Server: SQL Server Management Studio

Version Control: Git and Git Hub

IDE: Visual Studio

SYSTEM ARCHITECTURE AND REQUIREMENTS

Client-Server Architecture:

Tour and Stay is designed with a robust and scalable system architecture to ensure a seamless and efficient hotel booking experience. Below is a detailed overview of the system architecture and the requirements needed to support it.

Requirements to use this application:

i. **Hardware Requirement:**
   - A windows device.
   - Internet Connection.

ii. **Software Requirement:**
   - Tour and Stay software

PROJECT IMPLEMENTETION USING SPIRAL MODEL

SPIRAL MODEL Spiral model is one of the most important Software Development Life Cycle models, which provides support for Risk Handling. In its diagrammatic representation, it looks like a spiral with many loops. The exact number of loops of the spiral is unknown and can vary from project to project. Each loop of the spiral is called a Phase of the software development process. The exact number of phases needed to develop the product can be varied by the project manager depending upon the project risks. As the project manager dynamically determines the number of phases, so the project manager has an important role to develop a product using the spiral model. The Spiral Model is a software development life cycle (SDLC) model that provides a systematic and iterative approach to software development. It is based on the idea of a spiral, with each iteration of the spiral representing a complete software development cycle, from requirements gathering and analysis to design, implementation, testing, and maintenance.

SYSTEM PLANNING

System Planning Project Planning:

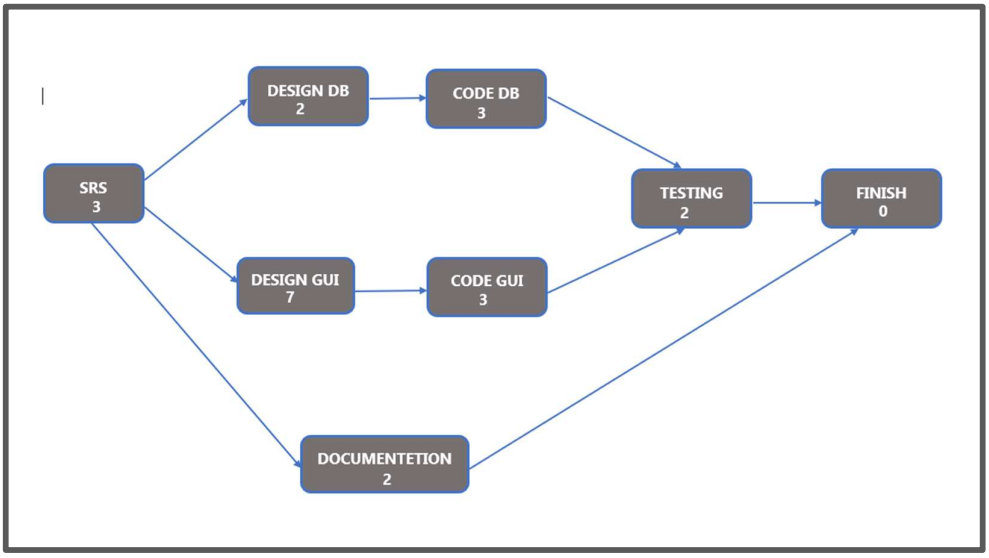
The objective of software project planning is to provide a framework that enables the manager to make reasonable estimates of resources, cost, and schedule, These estimates are made within a limited time frame at the beginning of a software project and should be updated regularly as the project progresses, In addition, estimates should attempt to define best case worst case scenarios that project outcomes can be bounded, The planning objective is achieved through a process of information discovery that leads to reasonable estimates.

Project Scheduling:

Project scheduling is an activity that distributes estimated effort across the planned project duration by allocating the effort to specific software engineering tasks. The number of basic principles guide the project scheduling is as follows:

- **Compartmentalization:** The project must be compartmentalized into a number of manageable activities and tasks. To accomplish compartmentalization, both the product and the process are decomposed.
- **Interdependence:** The interdependence of each compartmentalized activity or task must be determined. Some tasks must occur in-sequence while others can occur in parallel. Some activities cannot commence until the work product produced by another is available. Other activities can occur independently.
- **Time allocation:** Each task is scheduled must be allocated some number of work units. In addition, each task must be assigned a start date and a completion date.

- **Defined outcomes:** Every task that is scheduled should have a defined outcome. For software projects the outcome is normally a work product or a part of a work product.
- **Defined milestones:** Every tasks or group of tasks should be associated with a project milestone. A milestone is accomplished when one or more work products has been reviewed for quality and has been approved.



PERT CHART

Methodology

Development Process

1. **Research and Planning**: Identifying requirements and defining the project scope.
2. **Design**: Creating the system architecture and database schema.
3. **Development**: Implementing the front-end and back-end functionalities.
4. **Testing**: Conducting unit testing, integration testing, and user acceptance testing.
5. **Deployment**: Setting up the server environment and deploying the application.

USER MANUAL

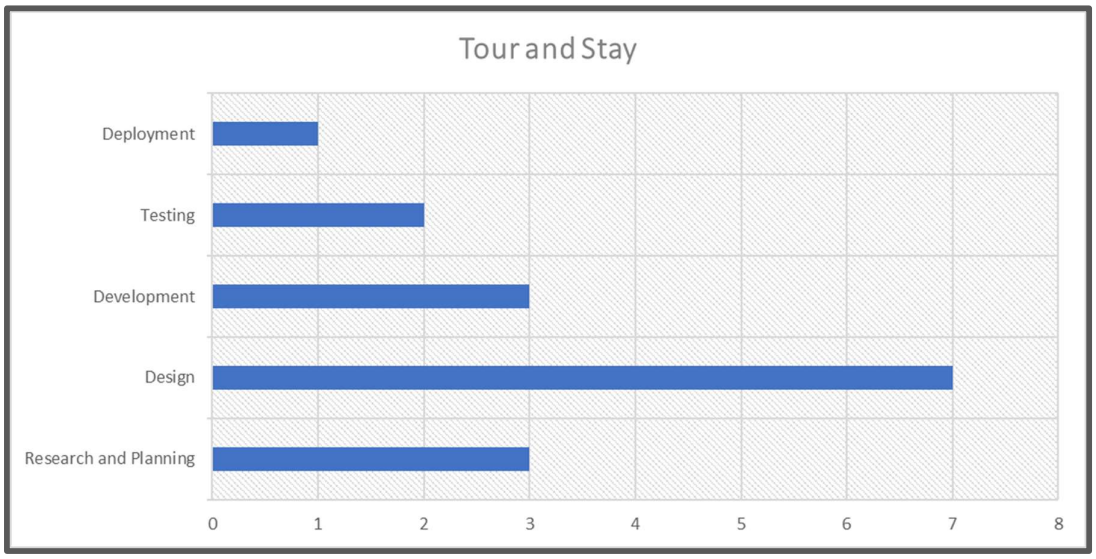
Installation

1. Download and install SSMS-2019 or higher.
2. Download Visual Studio
3. Clone the project repository.
4. Import the database schema.
5. Configure the connection settings in ConnectionDB.CS.
6. Start the SSMS.
7. Access the application via run it.

PROJECT MANAGEMENT

Timeline

- **Research and Planning**: 3days
- **Design**: 7days
- **Development**: 3 days
- **Testing**: 2 days
- **Deployment**: 1 day



GNATT CHART

DATABASE DESIGN

There are mainly 4 tables are used in our application:

i. **hotel_components:** To store faicilities data about the associated hotels.
ii. **hotel_details_tbl:** To save all login data about hotel.
iii. **user_details_tbl:** To save all the data about users.
iv. **notification_tbl:** All notification between hotels and users.

hotel_components:

Column Name	Data Type	NULL	Default	Constraints
hotel_ac	nvarchar(50)	YES	NULL	
hotel_tv	nvarchar(50)	YES	NULL	
hotel_fridge	nvarchar(50)	YES	NULL	
hotel_wifi	nvarchar(50)	YES	NULL	
hotel_view	nvarchar(50)	YES	NULL	
hotel_kitchen	nvarchar(50)	YES	NULL	
hotel_diniin	nvarchar(50)	YES	NULL	
hotel_travel	nvarchar(50)	YES	NULL	
hotel_parking	nvarchar(50)	YES	NULL	
hotel_id	int	NO	AUTO_INCREMENT	FORAIGN
hotel_food	nvarchar(50)	YES	NULL	

hotel_details_tbl

Column Name	Data Type	NULL	Default	Constraints
hotel_id	int	NO	AUTO_INCREMENT	Primary
hotel_phone	nvarchar(50)	YES		
hotel_owner	nvarchar(50)	YES		
hotel_country	nvarchar(50)	YES		
hotel_state	nvarchar(50)	YES		
hotel_spot	nvarchar(50)	YES		
hotel_pin	int	YES		
hotel_password	nvarchar(50)	NO	USER_INPUT	
hotel_pic	image	NO	USER_INPUT	
hotel_fair	int	NO	USER_INPUT	

user_details_tbl:

Column Name	Data Type	NULL	Default	Constraints
user_name	nvarchar(100)	YES		
user_mobile	nvarchar(12)	NO	USER_INPUT	Primary
user_email	nvarchar(100)	YES		
user_password	nvarchar(20)	YES		
user_img	image	YES		

notification_tbl:

Column Name	Data Type	NULL	Default	Constraints
user_mobile	nvarchar(12)	YES		FORAIGN
hotel_id	int	YES		FORAIGN
message	nvarchar(MAX)	YES		
status	nvarchar(50)	NO		
facilities	nvarchar(80)	NO		

CODE SNIPPETS

Connection File code:

```
using System;

using System.Collections.Generic;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;


namespace TourStay.DBUtilities

{

    internal class ConnectionDB

    {

        private string key = @"Data Source=BLACKHEART21\SQLEXPRESS;Initial
Catalog=tour_and_stay_db;Integrated Security=True;";

        public SqlConnection getKey()

        {

            SqlConnection con = new SqlConnection(key);

            return con;

        }


    }

}
```

Login Frontend Code:

```
namespace TourStay.LoginUC
{
    partial class UserLoginUC
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should be disposed; otherwise,
false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Component Designer generated code

        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
```

```
        System.ComponentModel.ComponentResourceManager resources = new
System.ComponentModel.ComponentResourceManager(typeof(UserLoginUC));
        this.login_img = new System.Windows.Forms.PictureBox();
        this.body_pnl = new System.Windows.Forms.Panel();
        this.minimize_btn = new System.Windows.Forms.PictureBox();
        this.exit_btn = new System.Windows.Forms.PictureBox();
        this.pictureBox1 = new System.Windows.Forms.PictureBox();
        this.signup_btn = new TourStay.CustomResources.RoundButton();
        this.login_btn = new TourStay.CustomResources.RoundButton();
        this.pass_inp = new System.Windows.Forms.TextBox();
        this.pass_lbl = new System.Windows.Forms.Label();
        this.mob_inp = new System.Windows.Forms.TextBox();
        this.mob_no = new System.Windows.Forms.Label();
        ((System.ComponentModel.ISupportInitialize)(this.login_img)).BeginInit();
        this.body_pnl.SuspendLayout();
        ((System.ComponentModel.ISupportInitialize)(this.minimize_btn)).BeginInit();
        ((System.ComponentModel.ISupportInitialize)(this.exit_btn)).BeginInit();
        ((System.ComponentModel.ISupportInitialize)(this.pictureBox1)).BeginInit();
        this.SuspendLayout();
        //
        // login_img
        //
        this.login_img.Dock = System.Windows.Forms.DockStyle.Fill;
        this.login_img.Image = ((System.Drawing.Image)(resources.GetObject("login_img.Image")));
        this.login_img.Location = new System.Drawing.Point(0, 0);
        this.login_img.Name = "login_img";
        this.login_img.Size = new System.Drawing.Size(900, 600);
        this.login_img.SizeMode = System.Windows.Forms.PictureBoxSizeMode.StretchImage;
        this.login_img.TabIndex = 2;
        this.login_img.TabStop = false;
        //
```

```
            // body_pnl
            //
            this.body_pnl.Controls.Add(this.minimize_btn);
            this.body_pnl.Controls.Add(this.exit_btn);
            this.body_pnl.Controls.Add(this.pictureBox1);
            this.body_pnl.Controls.Add(this.signup_btn);
            this.body_pnl.Controls.Add(this.login_btn);
            this.body_pnl.Controls.Add(this.pass_inp);
            this.body_pnl.Controls.Add(this.pass_lbl);
            this.body_pnl.Controls.Add(this.mob_inp);
            this.body_pnl.Controls.Add(this.mob_no);
            this.body_pnl.Controls.Add(this.login_img);
            this.body_pnl.Dock = System.Windows.Forms.DockStyle.Fill;
            this.body_pnl.Location = new System.Drawing.Point(0, 0);
            this.body_pnl.Name = "body_pnl";
            this.body_pnl.Size = new System.Drawing.Size(900, 600);
            this.body_pnl.TabIndex = 3;
            //
            // minimize_btn
            //
            this.minimize_btn.BackColor = System.Drawing.Color.White;
            this.minimize_btn.Cursor = System.Windows.Forms.Cursors.Hand;
            this.minimize_btn.Image =
((System.Drawing.Image)(resources.GetObject("minimize_btn.Image")));
            this.minimize_btn.Location = new System.Drawing.Point(823, 12);
            this.minimize_btn.Name = "minimize_btn";
            this.minimize_btn.Size = new System.Drawing.Size(20, 20);
            this.minimize_btn.SizeMode = System.Windows.Forms.PictureBoxSizeMode.StretchImage;
            this.minimize_btn.TabIndex = 21;
            this.minimize_btn.TabStop = false;
            this.minimize_btn.Click += new System.EventHandler(this.minimize_btn_Click);
```

```
            //
            // exit_btn
            //
            this.exit_btn.BackColor = System.Drawing.Color.White;
            this.exit_btn.Cursor = System.Windows.Forms.Cursors.Hand;
            this.exit_btn.Image = ((System.Drawing.Image)(resources.GetObject("exit_btn.Image")));
            this.exit_btn.Location = new System.Drawing.Point(868, 18);
            this.exit_btn.Name = "exit_btn";
            this.exit_btn.Size = new System.Drawing.Size(10, 10);
            this.exit_btn.SizeMode = System.Windows.Forms.PictureBoxSizeMode.StretchImage;
            this.exit_btn.TabIndex = 20;
            this.exit_btn.TabStop = false;
            this.exit_btn.Click += new System.EventHandler(this.exit_btn_Click);
            //
            // pictureBox1
            //
            this.pictureBox1.BackColor = System.Drawing.Color.White;
            this.pictureBox1.Cursor = System.Windows.Forms.Cursors.Hand;
            this.pictureBox1.Image =
((System.Drawing.Image)(resources.GetObject("pictureBox1.Image")));
            this.pictureBox1.Location = new System.Drawing.Point(818, 541);
            this.pictureBox1.Name = "pictureBox1";
            this.pictureBox1.Size = new System.Drawing.Size(70, 55);
            this.pictureBox1.SizeMode = System.Windows.Forms.PictureBoxSizeMode.StretchImage;
            this.pictureBox1.TabIndex = 19;
            this.pictureBox1.TabStop = false;
            this.pictureBox1.Click += new System.EventHandler(this.pictureBox1_Click);
            //
            // signup_btn
            //
            this.signup_btn.BackColor = System.Drawing.Color.Tomato;
```

```
            this.signup_btn.BackgroundColor = System.Drawing.Color.Tomato;
            this.signup_btn.BorderColor = System.Drawing.Color.Black;
            this.signup_btn.BorderRadius = 15;
            this.signup_btn.BorderSize = 0;
            this.signup_btn.Cursor = System.Windows.Forms.Cursors.Hand;
            this.signup_btn.FlatAppearance.BorderSize = 0;
            this.signup_btn.FlatStyle = System.Windows.Forms.FlatStyle.Flat;
            this.signup_btn.Font = new System.Drawing.Font("Consolas", 12F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.signup_btn.ForeColor = System.Drawing.Color.White;
            this.signup_btn.Location = new System.Drawing.Point(667, 400);
            this.signup_btn.Name = "signup_btn";
            this.signup_btn.Size = new System.Drawing.Size(145, 45);
            this.signup_btn.TabIndex = 18;
            this.signup_btn.Text = "SIGNUP";
            this.signup_btn.TextColor = System.Drawing.Color.White;
            this.signup_btn.UseVisualStyleBackColor = false;
            this.signup_btn.Click += new System.EventHandler(this.signup_btn_Click);
            //
            // login_btn
            //
            this.login_btn.BackColor = System.Drawing.Color.FromArgb(((int)(((byte)(111)))),
((int)(((byte)(247)))), ((int)(((byte)(129)))));
            this.login_btn.BackgroundColor = System.Drawing.Color.FromArgb(((int)(((byte)(111)))),
((int)(((byte)(247)))), ((int)(((byte)(129)))));
            this.login_btn.BorderColor = System.Drawing.Color.Black;
            this.login_btn.BorderRadius = 15;
            this.login_btn.BorderSize = 0;
            this.login_btn.Cursor = System.Windows.Forms.Cursors.Hand;
            this.login_btn.FlatAppearance.BorderSize = 0;
            this.login_btn.FlatStyle = System.Windows.Forms.FlatStyle.Flat;
            this.login_btn.Font = new System.Drawing.Font("Consolas", 12F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
```

```
            this.login_btn.ForeColor = System.Drawing.Color.White;
            this.login_btn.Location = new System.Drawing.Point(457, 400);
            this.login_btn.Name = "login_btn";
            this.login_btn.Size = new System.Drawing.Size(145, 45);
            this.login_btn.TabIndex = 17;
            this.login_btn.Text = "LOGIN";
            this.login_btn.TextColor = System.Drawing.Color.White;
            this.login_btn.UseVisualStyleBackColor = false;
            this.login_btn.Click += new System.EventHandler(this.login_btn_Click);
            //
            // pass_inp
            //
            this.pass_inp.BorderStyle = System.Windows.Forms.BorderStyle.FixedSingle;
            this.pass_inp.Font = new System.Drawing.Font("Consolas", 12F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.pass_inp.Location = new System.Drawing.Point(457, 326);
            this.pass_inp.Name = "pass_inp";
            this.pass_inp.PasswordChar = '*';
            this.pass_inp.Size = new System.Drawing.Size(355, 31);
            this.pass_inp.TabIndex = 16;
            //
            // pass_lbl
            //
            this.pass_lbl.AutoSize = true;
            this.pass_lbl.BackColor = System.Drawing.Color.White;
            this.pass_lbl.Font = new System.Drawing.Font("Consolas", 12F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.pass_lbl.ForeColor = System.Drawing.Color.Gray;
            this.pass_lbl.Location = new System.Drawing.Point(453, 288);
            this.pass_lbl.Name = "pass_lbl";
            this.pass_lbl.Size = new System.Drawing.Size(219, 23);
            this.pass_lbl.TabIndex = 15;
```

```
            this.pass_lbl.Text = "Enter your password";
            this.pass_lbl.Click += new System.EventHandler(this.pass_lbl_Click);
            //
            // mob_inp
            //
            this.mob_inp.BorderStyle = System.Windows.Forms.BorderStyle.FixedSingle;
            this.mob_inp.Font = new System.Drawing.Font("Consolas", 12F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.mob_inp.Location = new System.Drawing.Point(457, 222);
            this.mob_inp.Name = "mob_inp";
            this.mob_inp.Size = new System.Drawing.Size(355, 31);
            this.mob_inp.TabIndex = 14;
            //
            // mob_no
            //
            this.mob_no.AutoSize = true;
            this.mob_no.BackColor = System.Drawing.Color.White;
            this.mob_no.Font = new System.Drawing.Font("Consolas", 12F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.mob_no.ForeColor = System.Drawing.Color.Gray;
            this.mob_no.Location = new System.Drawing.Point(453, 184);
            this.mob_no.Name = "mob_no";
            this.mob_no.Size = new System.Drawing.Size(274, 23);
            this.mob_no.TabIndex = 13;
            this.mob_no.Text = "Enter your mobile number";
            this.mob_no.Click += new System.EventHandler(this.mob_no_Click);
            //
            // UserLoginUC
            //
            this.AutoScaleDimensions = new System.Drawing.SizeF(8F, 16F);
            this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
            this.Controls.Add(this.body_pnl);
```

```
            this.Name = "UserLoginUC";
            this.Size = new System.Drawing.Size(900, 600);
            ((System.ComponentModel.ISupportInitialize)(this.login_img)).EndInit();
            this.body_pnl.ResumeLayout(false);
            this.body_pnl.PerformLayout();
            ((System.ComponentModel.ISupportInitialize)(this.minimize_btn)).EndInit();
            ((System.ComponentModel.ISupportInitialize)(this.exit_btn)).EndInit();
            ((System.ComponentModel.ISupportInitialize)(this.pictureBox1)).EndInit();
            this.ResumeLayout(false);

        }

        #endregion
        private System.Windows.Forms.PictureBox login_img;
        private System.Windows.Forms.Panel body_pnl;
        private System.Windows.Forms.PictureBox minimize_btn;
        private System.Windows.Forms.PictureBox exit_btn;
        private System.Windows.Forms.PictureBox pictureBox1;
        private CustomResources.RoundButton signup_btn;
        private CustomResources.RoundButton login_btn;
        private System.Windows.Forms.TextBox pass_inp;
        private System.Windows.Forms.Label pass_lbl;
        private System.Windows.Forms.TextBox mob_inp;
        private System.Windows.Forms.Label mob_no;
    }
}
```

Login Backend Code

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using TourStay.DBUtilities;
using TourStay.FormScreens;
using TourStay.UtilityClass;

namespace TourStay.LoginUC
{
    public partial class UserLoginUC : UserControl
    {
        private FormAndUCLoader load = new FormAndUCLoader();
        private UserLoginInfoHolder info = new UserLoginInfoHolder();
        public UserLoginUC()
        {
            InitializeComponent();
        }

        private void minimize_btn_Click(object sender, EventArgs e)
        {
            if (this.ParentForm != null)
            {
```

```
        this.ParentForm.WindowState = FormWindowState.Minimized;
    }
}

private void mob_no_Click(object sender, EventArgs e)
{
    mob_inp.Focus();
}

private void pass_lbl_Click(object sender, EventArgs e)
{
    pass_inp.Focus();
}

private void pictureBox1_Click(object sender, EventArgs e)
{
    load.changeUC(new LoginOptionUC(), body_pnl);
}

private void exit_btn_Click(object sender, EventArgs e)
{
    Application.Exit();
}

private void signup_btn_Click(object sender, EventArgs e)
{
    load.changeUC(new UserSignupPage(), body_pnl);

}

private void login_btn_Click(object sender, EventArgs e)
```

```
        {
            UserInfoUpdater  upd = new UserInfoUpdater(mob_inp.Text);
            if (upd.isUser())
            {
                if(info.u_passwordGet() == pass_inp.Text)
                {
                    load.changeForm(this.ParentForm, new UserForm());
                }
                else
                {
                    MessageBox.Show("You entered wrong password");
                }
            }
            else
            {
                MessageBox.Show("No user found on this mobile number.");
            }

        }
    }
}
```

Logout Code:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
```

```
using System.Threading.Tasks;
using System.Windows.Forms;
using TourStay.FormScreens;
using TourStay.UtilityClass;

namespace TourStay.LogoutSystem
{
    public partial class LogoutForm : Form
    {
        FormAndUCLoader load = new FormAndUCLoader();
        public LogoutForm()
        {
            InitializeComponent();
        }

        private void yes_btn_Click(object sender, EventArgs e)
        {
            load.changeForm(this, new LoginScreen());
            load.closeAllForm();
            removeHotelComps();
            removeHotelDetails();
            removeNotification();
        }

        private void no_btn_Click(object sender, EventArgs e)
        {
            this.Hide();
        }
        private void removeNotification()
        {
            NotificationHolder hold = new NotificationHolder();
```

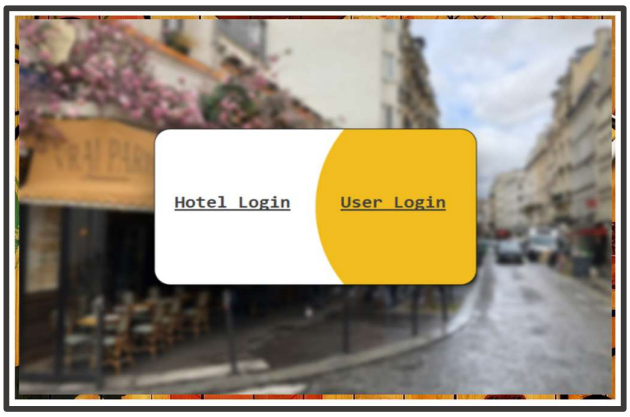
```
    hold.set_hid(00000);
    hold.set_facilities(null);
    hold.set_message(null);
    hold.set_status(null);
    hold.set_user_mobile(null);
}
private void removeHotelComps()
{
    HotelFormDataHolder hold = new HotelFormDataHolder();
    hold.set_ac("");
    hold.set_dining("");
    hold.set_food("");
    hold.set_fridge("");
    hold.set_kitchen("");
    hold.set_parking("");
    hold.set_travel("");
    hold.set_tv("");
    hold.set_view("");
    hold.set_wifi("");
}

private void removeHotelDetails()
{
    HotelInfoHolder hold = new HotelInfoHolder();
    hold.set_hid(0000);
    hold.set_pin(0000);
    hold.set_fair(0000);
    hold.set_country("");
    hold.set_img(null);
    hold.set_owner("");
    hold.set_password(null);
```

```
                hold.set_phno(null);
                hold.set_spot(null);
                hold.set_state(null);
            }
        }
    }
```

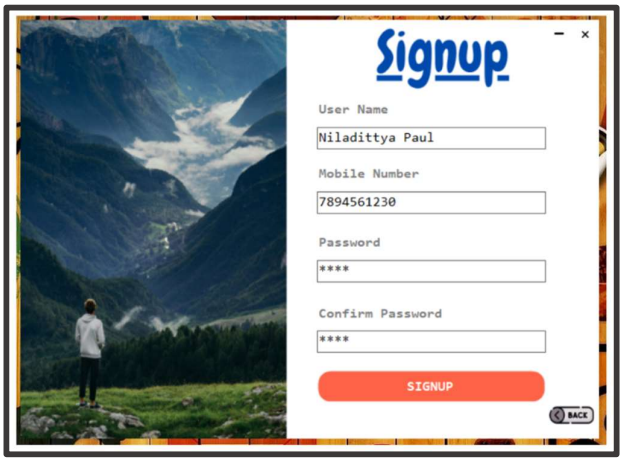
SCREENSHOTS

LOGIN Section

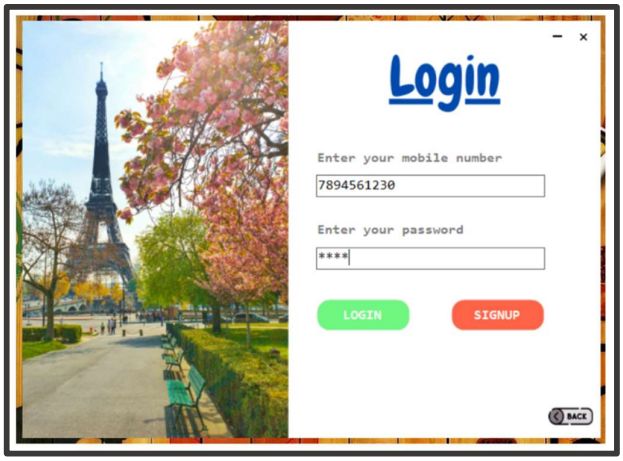
Splash Screen



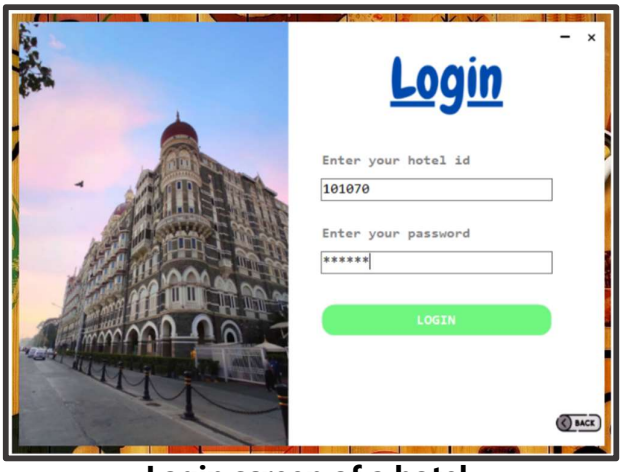
Login type screen



Signup screen of a user

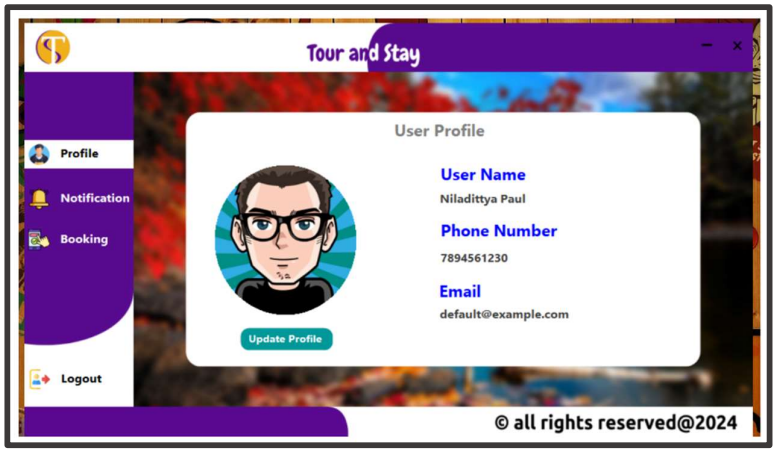


Login screen of a user.

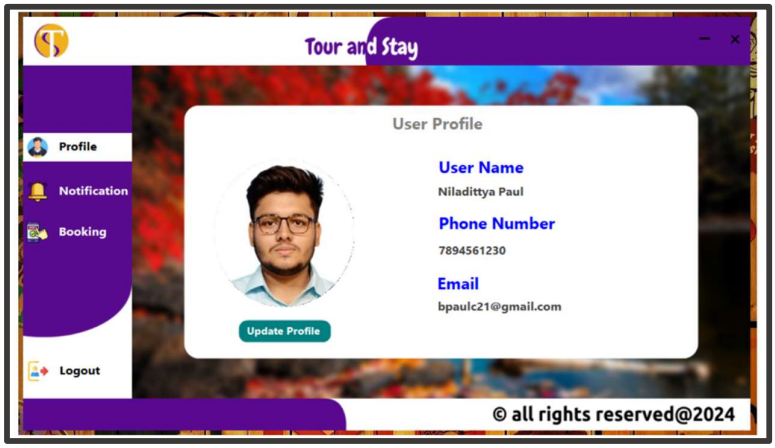


Login screen of a hotel.

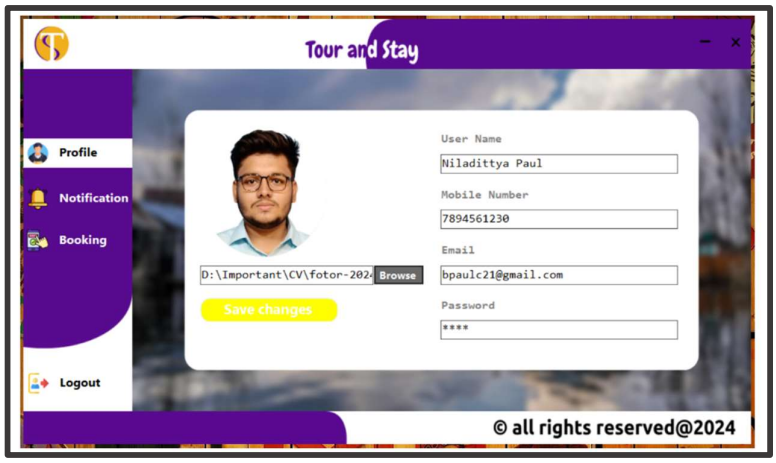
User Section



Default profile page

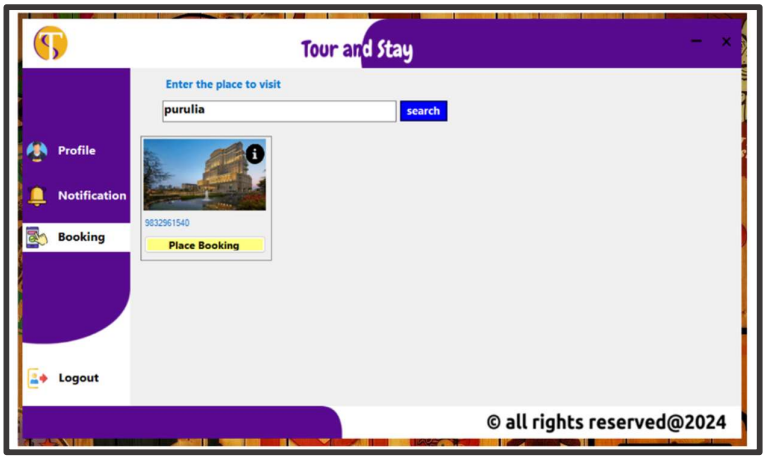


User profile page

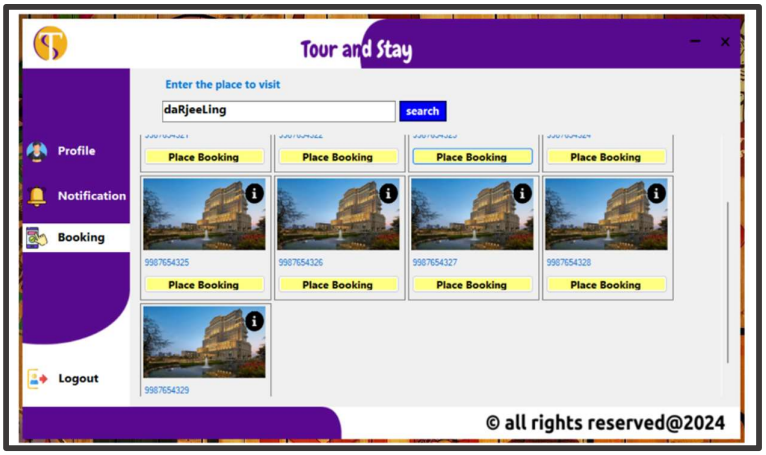


User profile update page

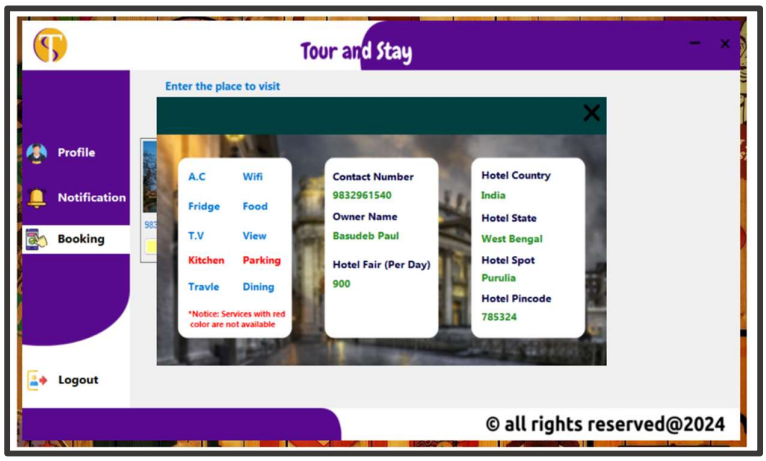
User notification page



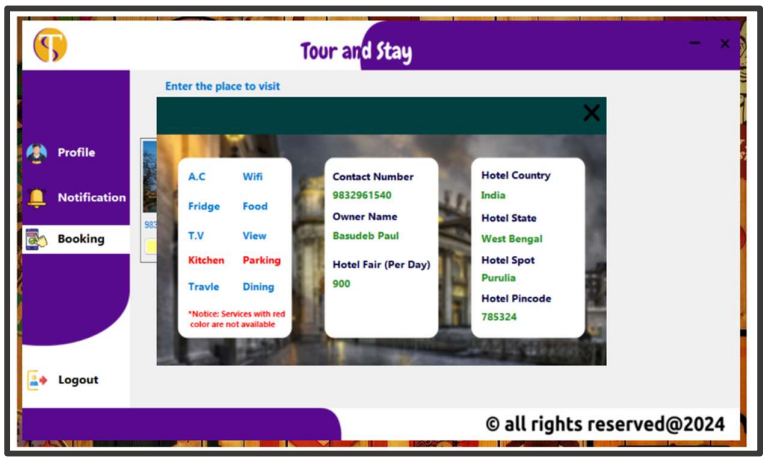
Single hotel found on searched location



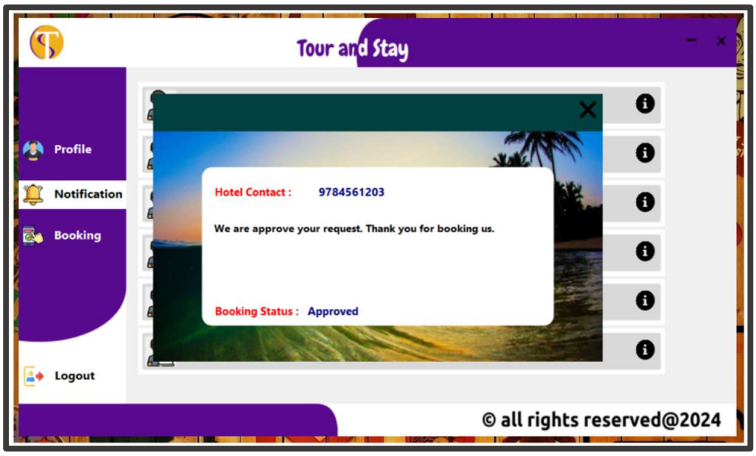
Multiple hotels found on searched location



User notification page



Information of the hotel selected from searched list



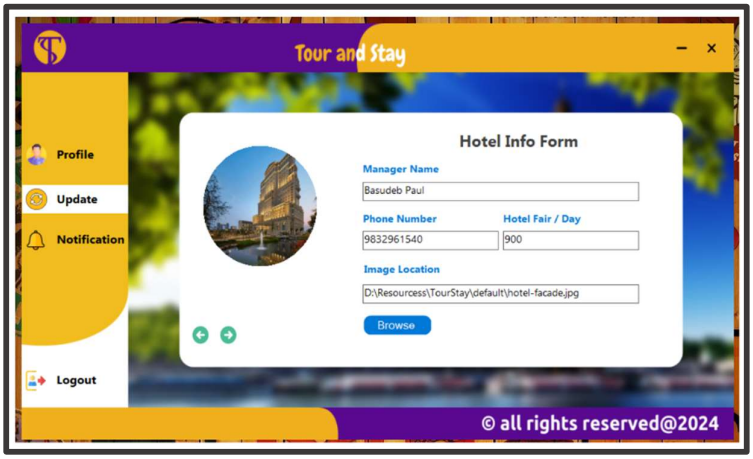
Booking approval or declined notification by hotel


Tour and Stay
Profile
Notification
Booking
Logout
User Profile
Logout
Are you sure want to logging out??
YES
NO
Paul
Number
bpaulc21@gmail.com
Update Profile
© all rights reserved@2024

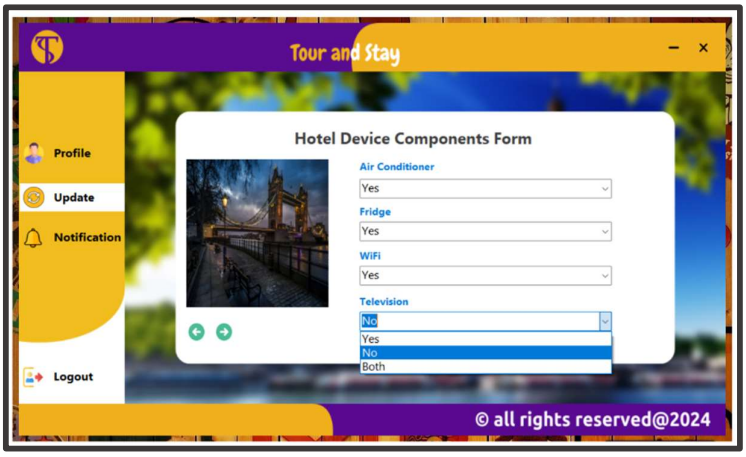
User logout

Hotel Section

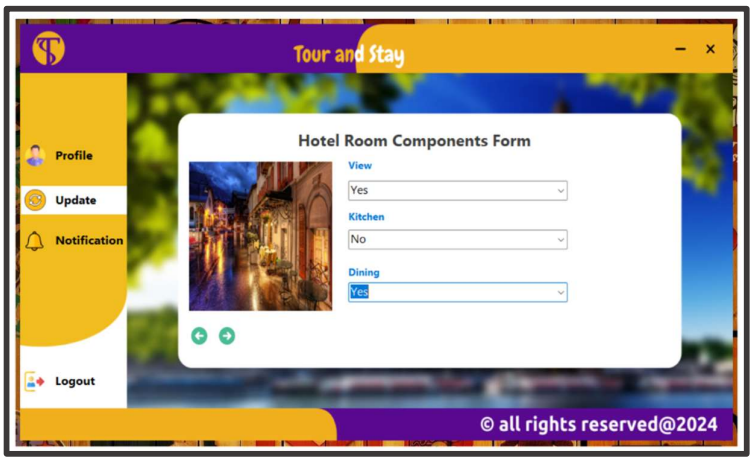
Hotel profile page



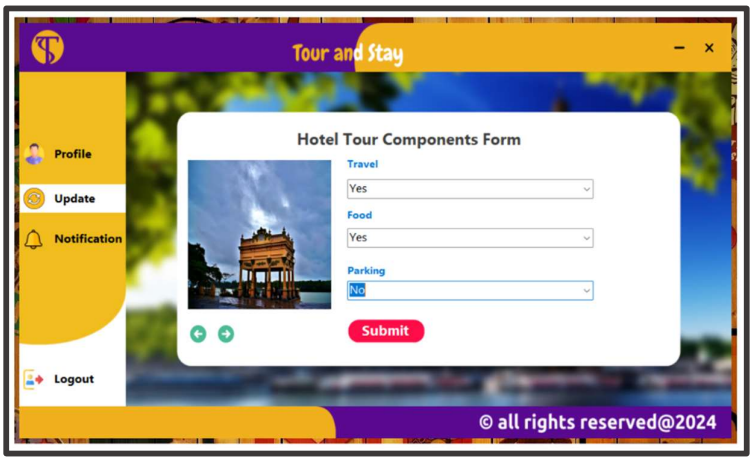
Hotel profile update page one



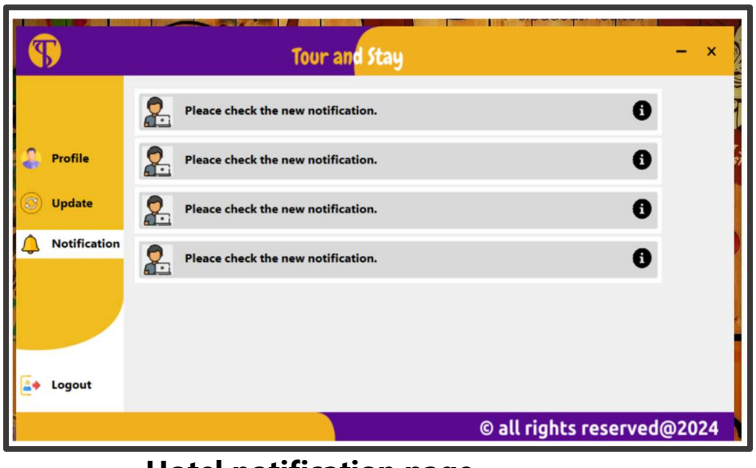
Hotel profile update page two



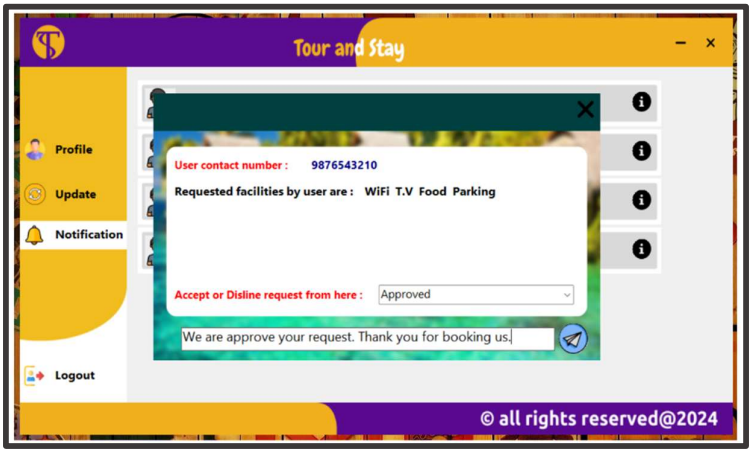
Hotel profile update page three



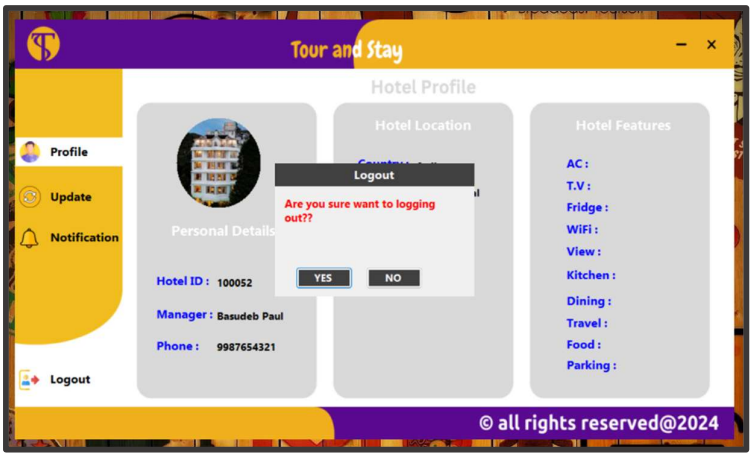
Hotel profile update page four



Hotel notification page



User booking request notification



Logout page

FUTURE SCOPE

Additional Features:

- Implement chat functionality for users and hotels.
- Implementing payments for booking hotel.
- Improve searching mechanism.

Performance Improvements:

- Optimize database queries to improve performance.
- Implement caching mechanisms to reduce server load.

CONCLUSION

- The **Tour and Stay** application represent a significant advancement in the hospitality industry by leveraging technology to simplify and enhance the hotel booking process. This Windows-based application offers a seamless and user-friendly interface that allows users to search for, book, and manage hotel reservations from the comfort of their homes.
- By partnering with franchisee hotels, Tour and Stay ensures a wide range of accommodation options, each with detailed profiles and up-to-date information on available facilities. This transparency helps users make informed decisions based on their preferences and needs.
- The real-time notification system and efficient booking management tools empower hotels to handle user requests promptly and effectively. Users receive immediate updates on their booking status, enhancing their overall experience and satisfaction.
- The robust system architecture, encompassing client-side, server-side, and database components, ensures scalability, security, and performance. By adhering to best practices in data security and privacy, the application maintains user trust and complies with relevant regulations.
- Tour and Stay's comprehensive feature set, including user account management, advanced search filters, custom facility requests, and secure payment integration, provides a holistic solution for both users and hotel partners. The continuous feedback and improvement loop ensures that the application evolves with user needs and technological advancements.
- In conclusion, **Tour and Stay** not only simplifies the hotel booking process but also fosters strong partnerships with hotels, creating a mutually beneficial ecosystem. This application is poised to become a leading platform in the hospitality industry, offering unparalleled convenience, transparency, and efficiency for users and hotels alike.