

27/03/2025

# C2P - Hotel Project Manual

C2P Hotel Management System



Authors: Mark Mukiiza, Hamed Zon, Mantas  
Bielevicius, Szymon Stepniak

## Table of Contents

User Interface	Written By: Mark Mukiiza .....	2
1.1 Home Screen.....		2
1.2 Menus.....		4
1.3 Add Screens .....		5
1.4 Delete Screens.....		7
1.5 Amend/View Screens.....		8
System Layout	Written By: Mark Mukiiza .....	9
Database Tables	Written By: Mark Mukiiza .....	10
Group Work	Written By: Mark Mukiiza .....	20
Website CSS	Written By: Mark Mukiiza .....	20
Menus and Sidebar	Written By: Mark Mukiiza .....	30
Individual Work (Code)	.....	40
Mark Mukiiza	Written By: Mark Mukiiza .....	40
Add Stock Item Screen .....		40
Delete Stock Item Screen.....		47
Amend/View Stock Item Screen.....		53
Place Orders Screen .....		62
Hamed Zon	Written By: Hamed Zon .....	73
Add Guest Screen .....		73
Amend/View Guest Screen.....		79
Delete Guest Screen.....		87
Add Extras Screen.....		95
Mantas Bilevicius	Written By: Mantas Bilevicius .....	105
Add Staff Screen .....		105
Amend/View Screen .....		111
Delete Screen .....		119
Guest Check-Out Screen.....		124
Szymon Stepniak	Written By: Szymon Stepniak.....	131
Add Room Page .....		131
Delete Room .....		136
Amend / View Page .....		139
Check in menu .....		143

# User Interface

Written By: Mark Mukiza

The following subsections describe each screen type in the system: Add screens, Delete screens, Amend/View screens, Home Screen, and Menus. Each description covers:

- **Layout and Alignment**
  - **Background and Border**
  - **Logo**
  - **Font and Font Size**
  - **Text Colour**
  - **Headings**
  - **Position of Buttons and Other Elements**
- 

## 1.1 Home Screen

The **Home Screen** consists of a central menu area containing clickable "cards" that lead to different pages (e.g., Check-in/Check-out, Extras, etc.). These cards are evenly arranged and centred both horizontally and vertically using a grid layout.

The background is a full-screen image with a subtle overlay to ensure readability. Each menu card has a dark background with slightly rounded corners and a shadow effect.

A small logo is positioned in the top-right corner. The main heading, "Main Menu," is displayed obviously. Each menu item has a large title (~24px) and a smaller description (~14px).

The text is primarily white or near white on a dark background. When a menu item is hovered over, its background slightly lightens for visual feedback.

Instead of traditional buttons, the menu items function as clickable cards containing both icons and text. A "burger menu" icon in the top-left corner allows access to a collapsible sidebar.

1. **Layout & Alignment**
    - Central "card" with "menu-item" blocks, each linking to a different page (Check-in/Check-out, Extras, etc.).
    - The entire "card" grid is centred horizontally and vertically (.card-container with `display: grid;`).
  2. **Background & Border**
    - Full-screen background image (`bed-Background.jpg`), covered with a slight overlay.
    - Each "card" has a dark background (`rgba(0,0,0,0.95)`) with rounded corners and a box-shadow.
  3. **Logo**
    - A small logo image (`C2PLogoNew.png`) anchored in the page header at the top-right corner.
  4. **Font & Font Size**
    - Large main heading: "Main Menu"
    - Each menu item uses a bigger font for the title (~24px) and a smaller subtitle (~14px).
  5. **Text Colour**
    - White or near-white text over dark background for each "menu item."
    - On hover, cards lighten slightly (`rgba(255, 255, 255, 0.2)`).
  6. **Headings**
    - "Main Menu" in the header.
    - Each menu item has an `h3` with the feature name and a small paragraph with the description.
  7. **Position of Buttons & Other Elements**
    - No typical "buttons." Instead, each card is clickable (an `<a>` tag with an icon and text).
    - The burger-menu icon is in the top-left, toggling the sidebar.
-



## 1.2 Menus

The **Menu System** includes a fixed header at the top (80px high) and a sidebar that slides in from the left when needed.

The top header has a dark background with a subtle bottom border, while the sidebar has a semi-transparent black background. By default, the sidebar remains hidden until the user activates it.

A small logo is positioned in the top-right corner of the header. Text is white, with sidebar menu items set at around 14px. When a menu item is hovered over, it highlights slightly with a light background.

The main header typically displays a title relevant to the current page. Sidebar links have a dedicated heading and are grouped into dropdown sections.

A burger menu button in the top-left corner allows users to toggle the sidebar. A logout button is positioned at the bottom. Sidebar dropdowns expand when hovered over, revealing additional options.

1. **Layout & Alignment**
  - A fixed top header (80px high) and a collapsible/slide-out sidebar on the left.
2. **Background & Border**
  - Header bar with background: `rgba(0, 0, 0, 0.8)` and a 1px bottom border.
  - Sidebar: `rgba(0,0,0,0.5)` by default, hidden off-screen until toggled.
3. **Logo**
  - A small logo image (`C2PLogoNew.png`) anchored in the page header at the top-right corner.
4. **Font & Font Size**
  - White text in both the header and sidebar menu items, ~14px.
5. **Text Colour**
  - White for main menu items, subtle highlight (`background-color: rgba(255,255,255,0.2)`) on hover.
6. **Headings**
  - The main header usually has an `<h1>` representing the page name or module name.
  - Sidebar links have headings and are grouped under dropdown sections.
7. **Position of Buttons & Other Elements**
  - **Burger menu** in the top-left corner.
  - Sidebar “exit” button (or link) at the bottom to log out.
  - Each sidebar dropdown uses a “hover to reveal” approach.

## 1.3 Add Screens

The **Add Screens** have a simple layout with a single-column form that appears in the centre of the screen. Each field is stacked vertically and clearly labelled. The form is enclosed within a Div and has a title, such as “Stock Details.” The background features a large image with a dark overlay to improve readability. The form itself has a white background with a slight shadow effect, and its edges are rounded for a softer appearance. A small logo is placed in the top-right corner of the page. The font used across the page is ‘Lato’, a clean and modern sans-serif style. Labels within the form are bold and sized between 14px and 16px, while headings are larger, around 24px to 30px, and appear in dark grey. Text is generally light in colour when placed on dark backgrounds and dark (#333) when placed on the white form. The main page heading, like “Add Stock,” appears in white against a dark bar at the top.

Buttons, such as “Add” and “Save Changes” are located at the bottom of the form. Listboxes (<select> elements) and standard input fields (text, number, etc.) are placed directly below their labels. Validation is done either within the form using HTML patterns or JavaScript.

### 1. Layout & Alignment

- Uses a single-column form layout centred on the screen.
- Form fields are stacked vertically, each labelled clearly.
- The form is enclosed in a <fieldset> with a <legend> specifying the content (e.g., “Stock Details”).

### 2. Background & Border

- Background: Large background image (bed-Background.jpg) set to cover, with a dark overlay for improved readability.
- Form container: White background (background-color: white;) with a box shadow.
- Border: Rounded corners (e.g., border-radius: 10px;) on form elements.

### 3. Logo

- A small logo image (C2PLogoNew.png) anchored in the page header at the top-right corner.

### 4. Font & Font Size

- Global font-family: 'Lato', sans-serif; .
- Standard form labels: Bold text, approximately 14px to 16px.
- Headings (H1): ~24px to 30px, centred, dark colour (#333).

### 5. Text Colour

- Body text is generally #f0f0f0 (pale white) over a dark or tinted background.
- In the form, text is #333 for high contrast against the white background.

### 6. Headings

- Main heading (“Add Stock,” etc.) displayed at the top in the page header, coloured white (color: #fff) on a dark bar.
- Sub-headings or <legend> text bold, with a slightly larger font size.

### 7. Position of Buttons & Other Elements

- **Buttons:** Typically, horizontally aligned at the bottom of the form (e.g., “Add,” “Clear”).
- **Other elements:**
  - <select> for “listboxes”.
  - Standard <input type="text">, <input type="number">, etc., each placed under its label.
  - **Validation** is either inline (e.g., HTML pattern) or via JavaScript functions (onchange="...").



## 1.4 Delete Screens

The **Delete Screens** follow a similar layout but include a listbox to select an item for deletion. Once an item is selected, its key details, such as Stock ID and Description, appear as read-only fields.

The background is the same as the Add Screens, featuring a dark overlay on a large background image. The form container remains white with rounded corners and a shadow.

A small logo is positioned in the top-right corner. Fonts and text styles remain consistent with other screens, using ‘Lato’ with form labels at 14px to 16px.

Headings are in white against a dark background, while the text within the form is black (#333) for clear visibility. The page title, like “Delete Stock,” is displayed obviously in the header.

The primary button, “Delete Item,” is placed either at the bottom or centre of the form. A confirmation popup appears before deletion is processed. The item’s details remain visible in read-only fields to ensure the correct selection before deletion.

### 1. Layout & Alignment

- Similar general layout but typically includes a listbox to select which record to delete.
- Key fields (e.g., Stock ID, Description) are displayed read-only once a user chooses an item.

### 2. Background & Border

- Same primary background (dark, with a cover image).
- Form container: White background (`background-color: white;`) with a box shadow.
- Border: Rounded corners (e.g., `border-radius: 10px;`) on form elements.

### 3. Logo

- A small logo image (C2PLLogoNew.png) anchored in the page header at the top-right corner.

### 4. Font & Font Size

- Matches the global style: `font-family: 'Lato', sans-serif;` and typical 14–16px for form text.

### 5. Text Colour

- White headings on dark background for the page header.
- Standard black text (#333) on white background for the form fields.

### 6. Headings

- Main heading (“Delete Stock,” etc.) displayed at the top in the page header, coloured white (`color: #fff`) on a dark bar.

### 7. Position of Buttons & Other Elements

- A single “Delete Item” button at the bottom or centre. JavaScript popup used to confirm deletions.
- The entity’s details (read-only) displayed in disabled `<input>` fields once selected from the listbox. This is to verify the correct entity is selected.

## 1.5 Amend/View Screens

For **Amend and View Screens**, the layout is adjusted to display more information without excessive scrolling. Instead of a single-column format, the form is divided into two columns placed side by side. These columns are organised within a grouped container inside a fieldset. The background remains the same as other screens, with a dark image overlay, while the form itself is in a white container with a shadow effect. A small logo is anchored at the top-right corner. Fonts and text styles remain consistent with other pages. Headings, such as "Amend/View [Entity]," are around 24px to 30px in size and appear at the top in the header. Text follows the standard theme, white text on a dark background in headers and black (#333) text inside the form for readability. Buttons are positioned deliberately. The "**Amend Details**" button unlocks the form fields for editing. The "**Save Changes**" button includes form validation and a confirmation popup before saving. Listboxes are used for fields such as supplier selection. Additional fields, like reorder quantity and reorder level, are placed in the right column for clarity.

### 1. Layout & Alignment

- Fields are separated into columns to reduce vertical scrolling for more data-heavy pages.
- Typically, a form with two columns side by side (`.left-column`, `.right-column` class in the CSS).
- Columns are grouped in `<div class="columns">` inside `<fieldset>`.

### 2. Background & Border

- Same primary background (dark, with a cover image).
- Form container: White background (`background-color: white;`) with a box shadow.

### 3. Logo

- A small logo image (C2PLogoNew.png) anchored in the page header at the top-right corner.

### 4. Font & Font Size

- Consistent across the site.
- Headings: ~24px to 30px for top-level.
- Form labels: ~14px, bold.

### 5. Text Colour

- White text in the header on a dark bar.
- #333 text in the form on a white background for clarity.

### 6. Headings

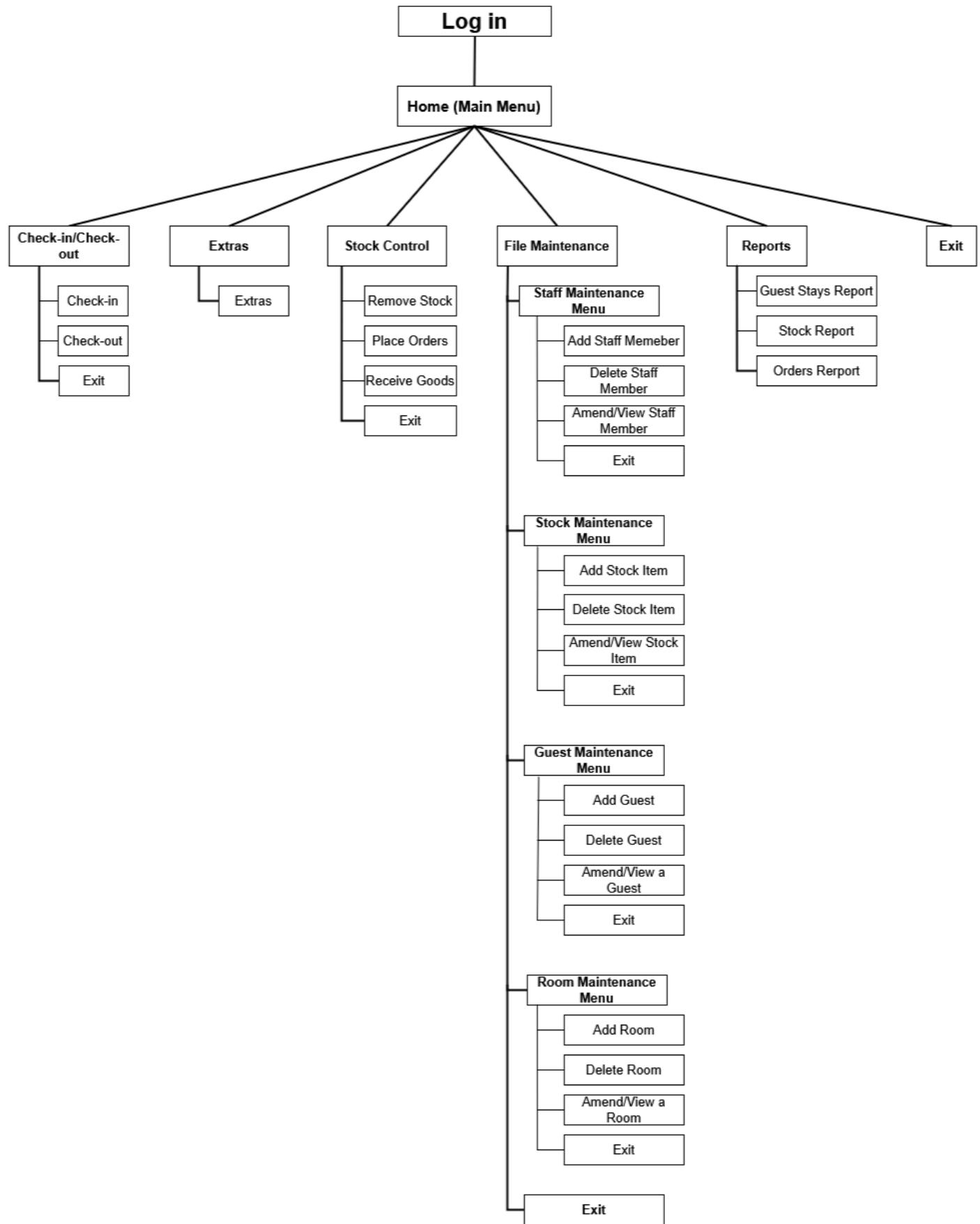
- H1: "Amend/View [Entity]" at the top, within the page header.

### 7. Position of Buttons & Other Elements

- "Amend Details" button toggles fields from *disabled* to *enabled*.
- "Save Changes" button triggers form validation plus a confirm pop-up.
- Supplier is a `<select>` listbox.
- Additional fields (e.g., reorder quantity, reorder level) are aligned to the right column.

# System Layout

Written By: Mark Mukiiiza



# Database Tables

Written By: Mark Mukiza

Name	Type	Collation	Attributes	Null	Default	Comments	Extra
StockID 	int(11)			No	None		AUTO_INCREMENT
Description	varchar(255)	utf8_general_ci		No	None		
CostPrice	decimal(10,2)			No	None		
RetailPrice	decimal(10,2)			No	None		
ReorderQuantity	int(11)			No	None		
ReorderLevel	int(11)			No	None		
SupplierStockCode	varchar(50)	utf8_general_ci		No	None		
QuantityInStockRoom	int(11)			No	None		
SupplierID 	int(11)			No	None		
DateLastUpdate	timestamp			No	current_timestamp()		
DeletedFlag	tinyint(1)			No	0		

Figure 1. Stock Table Structure (Mark)

StockID	Description	CostPrice	RetailPrice	ReorderQuantity	ReorderLevel	SupplierStockCode	QuantityInStockRoom	SupplierID	DateLastUpdate	DeletedFlag
1	Large Bath Towels	2.00	3.50	50	20	TOWL001	35	3	2025-03-18 13:26:29	0
2	Small Shampoo Bottles	1.50	8.00	70	30	SHAM001	45	2	2025-03-18 13:38:15	0
3	Toiletry Kits	5.00	12.00	15	25	TOIL001	50	1	2025-03-13 15:08:09	0
4	Taylor Originals	0.80	1.30	50	20	TATO002	25	9	2025-03-03 11:59:39	0
5	Corona Extra 330ml	1.89	2.30	100	20	CORE355	50	2	2025-02-04 11:58:56	1
6	Water Carton 500ml	1.00	3.00	50	20	CWAT001	50	2	2025-02-04 20:20:53	0
7	Chocolate Chip Cookies	1.50	3.00	50	20	CHCK001	48	2	2025-02-04 20:28:29	0

Figure 2. Stock Table Sample Data (Mark)

Name	Type	Collation	Attributes	Null	Default	Comments	Extra
SupplierID	int(11)			No	None		AUTO_INCREMENT
Name	varchar(100)	utf8_general_ci		No	None		
Address	varchar(255)	utf8_general_ci		No	None		
TelephoneNumber	varchar(15)	utf8_general_ci		No	None		
EmailAddress	varchar(100)	utf8_general_ci		No	None		
DeletedFlag	tinyint(1)			No	0		

**Figure 3. Suppliers Table Structure (Mark)**

SupplierID	Name	Address	TelephoneNumber	EmailAddress	DeletedFlag
1	Bidfood Ireland	123 River Street, Dublin 1, Ireland	01 234 56789	salses@bidfoodireland.ie	0
2	Musgrave Hospitality	321 Birch Lane45 Market Square, Cork, Ireland	01 987 65432	sales@musgravehospitality.ie	0
3	Makro Ireland	78 Industrial Estate, Limerick, Ireland	061 234 5657	sales@makoireland.ie	0
4	Hotel Supplies Direct	90 Business Park, Galway, Ireland	091 876 5432	sales@hotelsuppliesdirect.ie	0
5	The Linen Warehouse	102 Linen Road, Waterford, Ireland	051 987 6543	orders@linenwarehouse.ie	0
6	C&C Group	88 Beverage Blvd, Dublin 2, Ireland	01 765 43621	sales@ccgroup.ie	0
7	Dublin Hospitality Supplies	15 Custom House Quay, Dublin 2, Ireland	01 345 67891	orders@dublinhospitalitysupplies.ie	0
8	Emerald Isle Catering Solutions	200 Orchard Road, Cork, Ireland	021 456 78907	sales@emeraldcatering.ie	0
9	Irish Provisions & Supplies	77 Supply Lane, Limerick, Ireland	061 987 6543	contact@irishprovisions.ie	0

**Figure 4. Suppliers Table Sample Data (Mark)**

Name	Type	Collation	Attributes	Null	Default	Comments	Extra
StaffID	int(11)			No	None		AUTO_INCREMENT
FirstName	varchar(50)	utf8_general_ci		Yes	NULL		
Surname	varchar(50)	utf8_general_ci		Yes	NULL		
Address	varchar(255)	utf8_general_ci		Yes	NULL		
Eircode	varchar(10)	utf8_general_ci		Yes	NULL		
TelephoneNumber	varchar(15)	utf8_general_ci		Yes	NULL		
JobTitle	varchar(50)	utf8_general_ci		Yes	NULL		
ManagerStatus	text	utf8_general_ci		Yes	NULL		
LoginName	varchar(50)	utf8_general_ci		Yes	NULL		
Password	varchar(255)	utf8_general_ci		No	staff		
DateLastUpdate	timestamp			No	current_timestamp()		ON UPDATE CURRENT_TIMESTAMP()
DeletedFlag	tinyint(1)			Yes	0		

**Figure 5. Staff Table Structure (Mark)**

StaffID	FirstName	Surname	Address	Eircode	TelephoneNumber	JobTitle	ManagerStatus	LoginName	Password	DateLastUpdate	DeletedFlag
1	Jane	Smith	456 Oak Avenue	D02 EFGH	0869876543	Receptionist	No	jsmith	staff	2025-03-13 15:00:32	0
2	Mark	Taylor	789 Pine Road	D03 IJKL	0851234567	Housekeeping Supervisor	No	mtaylor	staff	2025-03-13 15:00:32	0
3	Sarah	Williams	321 Birch Lane	D04 MNOP	0877891234	Manager	No	swilliams	staff	2025-03-13 15:00:32	0
4	Chris	Brown	654 Maple Drive	D05 QRST	0864567890	Chef	No	cbrown	staff	2025-03-13 15:00:32	0
5	Laura	Miller	987 Cedar Court	D06 UVWX	0871122334	Front Desk Clerk	No	lmiller	staff	2025-03-13 15:00:32	0
6	Mark	Mukiiza	3 Heads Avenue	R32 RKO1	0855671234	Assistant To The Regional Manager	No	MarkMukiiza	staff	2025-03-13 15:00:32	0
7	Mantas	Bielevicius	727 Carlow Drive	D06 MOK4	0857654321	Regional Manager	Yes	C00294564	staff	2025-03-18 12:54:02	1

**Figure 6. Staff Table Sample Data (Mantas)**

Name	Type	Collation	Attributes	Null	Default	Comments	Extra
RoomID	int(11)			No	None		AUTO_INCREMENT
Type	enum('single', 'twin', 'double', 'family')	utf8_general_ci		Yes	NULL		
CostPerNight	decimal(10,2)			Yes	NULL		
TelephoneExtensionNumber	varchar(10)	utf8_general_ci		Yes	NULL		
SpecialNotes	text	utf8_general_ci		Yes	NULL		
CurrentStatus	enum('Available', 'Occupied')	utf8_general_ci		Yes	Available		
DateLastUpdate	timestamp			No	current_timestamp()		ON UPDATE CURRENT_TIMESTAMP()
DeletedFlag	tinyint(1)			Yes	0		

**Figure 7. Room Table Structure (Mark)**

RoomID	Type	CostPerNight	TelephoneExtensionNumber	SpecialNotes	CurrentStatus	DateLastUpdate	DeletedFlag
1	double	150.00	101	None	Occupied	2025-03-19 13:23:15	0
2	single	100.00	102	None	Occupied	2025-02-25 18:33:06	1
3	twin	120.00	103	None	Available	2025-03-06 15:44:39	0
4	family	200.00	104	Suitable for families	Available	2025-02-04 11:58:12	0
5	double	180.00	105	Sea view	Available	2025-03-11 10:24:36	0
6	single	120.00	106	None	Available	2025-03-10 14:35:35	1
7	double	150.00	107	None	Available	2025-03-13 11:01:25	0
8	family	150.00	108	Suitable for a small family	Available	2025-03-10 14:36:34	1
9	double	140.00	109	None	Available	2025-03-10 14:22:14	0

**Figure 8. Room Table Sample Data (Szymon)**

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	GuestID	int(11)			No	None		AUTO_INCREMENT
2	FirstName	varchar(50)	utf8_general_ci		Yes	NULL		
3	Surname	varchar(50)	utf8_general_ci		Yes	NULL		
4	Address	varchar(255)	utf8_general_ci		Yes	NULL		
5	Eircode	varchar(10)	utf8_general_ci		Yes	NULL		
6	PhoneNumber	varchar(15)	utf8_general_ci		Yes	NULL		
7	EmailAddress	varchar(100)	utf8_general_ci		Yes	NULL		
8	DateLastUpdate	timestamp			No	current_timestamp()		ON UPDATE CURRENT_TIMESTAMP()
9	DeletedFlag	tinyint(1)			Yes	0		

**Figure 9. Guest Table Structure (Mark)**

GuestID	FirstName	Surname	Address	Eircode	PhoneNumber	EmailAddress	DateLastUpdate	DeletedFlag
1	John	Doe	123 Elm Street	D01 ABCD	0871234567	john.doe@example.com	2025-03-17 15:09:11	1
2	Jane	Smith	456 Oak Avenue	D02 EFGH	0869876543	jane.smith@example.com	2025-03-17 15:08:10	1
3	Michael	Johnson	789 Pine Road	D03 IJKK	0854321789	michael.johnson@example.com	2025-03-17 15:09:23	1
4	Emily	Philip	321 Park Way	D04 MNOP	0876543210	emily.davis@example.com	2025-03-17 15:09:53	1
5	David	Wilson	654 Maple Drive	D05 QRST	0861234567	david.wilson@example.com	2025-03-17 15:11:18	1
6	Emma	Salu	24 Bullock Park Green Road Carlow	R96 SHJ2	0874519784	emmasalu@gmail.com	2025-03-17 15:11:34	1
7	Jennifer	Boahen	113 Milbrook	R93 H419	083 0965614	Jenniferboahen@gmail.com	2025-03-17 15:07:02	0
8	Timothy	Casey	Kilkenny Dean Street	R95 KTY4	0874780089	tim.casey@setu.ie	2025-03-18 09:06:42	1
9	John	Marsten	Red Dead Street	W55 HFL1	0814613387	John.Marsten@setu.ie	2025-03-14 22:50:17	1

**Figure 10. Guest Table Sample Data (Hamed)**

Name	Type	Collation	Attributes	Null	Default	Comments	Extra
OrderID	int(11)			No	None		AUTO_INCREMENT
StockID	int(11)			Yes	NULL		
DateOfOrder	date			Yes	NULL		
TotalCost	decimal(10,2)			Yes	NULL		
DateReceived	date			Yes	NULL		

Figure 11. Orders Table Structure (Mark)

OrderID	StockID	DateOfOrder	TotalCost	DateReceived
1	18	2025-03-19	94.50	NULL
2	22	2025-03-19	130.00	NULL
3	1	2025-03-19	161.25	NULL

Figure 12. Orders Table Sample Data (Mark)

Name	Type	Collation	Attributes	Null	Default	Comments	Extra
OrderItemID 	int(11)			No	None		AUTO_INCREMENT
OrderID 	int(11)			Yes	NULL		
StockID 	int(11)			No	None		
SupplierID 	int(11)			Yes	NULL		
QuantityOrdered	int(11)			Yes	NULL		

Figure 13. Order Items Table Structure (Mark)

OrderItemID	OrderID	StockID	SupplierID	QuantityOrdered
1	1	18	1	20
2	1	17	1	50
3	1	15	1	50
4	2	22	5	10
5	3	1	3	50
6	3	19	3	25

Figure 14. Order Items Table Sample Data (Mark)

Name	Type	Collation	Attributes	Null	Default	Comments	Extra
MovementID 	int(11)			No	None		AUTO_INCREMENT
StockID 	int(11)			Yes	NULL		
StaffID 	int(11)			Yes	NULL		
Date Time	datetime			Yes	NULL		
Location	varchar(255)	utf8_general_ci		Yes	NULL		
QuantityRemoved	int(11)			Yes	NULL		

Figure 15. Stock Movement Table Structure (Mark)

MovementID	StockID	StaffID	Date Time	Location	QuantityRemoved
1	1	1	2025-02-01 10:00:00	Room 101	5
2	2	2	2025-02-02 11:00:00	Room 102	2
3	3	3	2025-02-03 12:00:00	Room 103	2
4	4	4	2025-02-04 13:00:00	Room 104	2
5	5	5	2025-02-05 14:00:00	Room 105	1

Figure 16. Stock Movement Table Sample Data (Mark)

Name	Type	Collation	Attributes	Null	Default	Comments	Extra
StayID 	int(11)			No	None		AUTO_INCREMENT
GuestID 	int(11)			Yes	NULL		
RoomID 	int(11)			Yes	NULL		
CheckInDate	date			Yes	NULL		
CheckOutDate	date			Yes	NULL		
NumberInGroup	int(11)			Yes	NULL		
TotalCost	decimal(10,2)			Yes	NULL		

Figure 17. Guest Stays Table Structure (Mark)

StayID	GuestID	RoomID	CheckInDate	CheckOutDate	NumberInGroup	TotalCost
1	1	1	2025-02-01	2025-02-05	2	600.00
2	2	2	2025-02-02	2025-02-06	1	400.00
3	3	3	2025-02-03	2025-02-07	3	720.00
4	4	4	2025-02-04	2025-02-08	4	800.00
5	5	5	2025-02-05	2025-02-09	2	720.00

Figure 18. Guest Stays Table Sample Data (Szymon, Mantas)

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	<b>ExtrasID</b> 	int(11)			No	<i>None</i>		AUTO_INCREMENT
2	<b>Area</b>	varchar(50)	utf8_general_ci		Yes	<i>NULL</i>		
3	<b>Description</b>	varchar(255)	utf8_general_ci		Yes	<i>NULL</i>		
4	<b>Cost</b>	decimal(10,2)			Yes	<i>NULL</i>		
5	<b>RoomID</b> 	int(11)			Yes	<i>NULL</i>		
6	<b>StaffID</b> 	int(11)			Yes	<i>NULL</i>		
7	<b>Date Time</b>	datetime			Yes	<i>NULL</i>		

Figure 19. Extras Table Structure (Mark)

ExtrasID	Area	Description	Cost	RoomID	StaffID	Date Time
1	Room Service	Breakfast Delivery	20.00	1	1	2025-02-04 08:00:00
2	Spa	Massage Session	50.00	1	1	2025-02-03 14:30:00
3	Gym	Personal Training	30.00	3	3	2025-02-02 10:00:00
4	Concierge	Tour Booking	15.00	4	4	2025-02-01 16:00:00
5	Laundry	Same-Day Service	10.00	1	5	2025-02-05 09:00:00

Figure 20. Extras Table Sample Data (Hamed)

# Group Work

Written By: Mark Mukiiiza

## Website CSS

Written By: Mark Mukiiiza

We discussed the look and style of the website together and decided on this style. Sidebar with a header showing which page you're currently on. Later on, the sidebar was made to toggle to appear or disappear.

```

1  /* =====
2   1) General Reset and Font (Resets defaults and sets base font)
3   ===== */
4   * { /* Selects all elements on the page */
5     margin: 0; /* Removes any default margin */
6     padding: 0; /* Removes any default padding */
7     box-sizing: border-box; /* Ensures that padding and border are included in the element's total width and height */
8     font-family: 'Lato', sans-serif; /* Sets the default font to 'Lato' with a sans-serif fallback */
9   }
10 /* =====
11  2) Body & Global Text (Styles the body element and sets global text colour)
12 ===== */
13 body { /* Targets the body element */
14   background: url("bed-Background.jpg") no-repeat center center fixed; /* Sets a fixed background image, centred with no repeat */
15   background-size: cover; /* Scales the background image to cover the entire screen */
16   position: relative; /* Sets the body position to relative for positioning child elements */
17   color: #f0f0f0; /* Sets the default text colour to a light grey */
18 }
19 /* =====
20  3) Page Header (Styles the fixed header at the top of the page)
21 ===== */
22 .page-header { /* Targets the element with the class 'page-header' */
23   position: fixed; /* Fixes the header so it remains visible at the top */
24   top: 0; /* Positions the header at the very top of the screen */
25   left: 0; /* Aligns the header to the left edge of the screen */
26   width: 100%; /* Makes the header span the full width of the screen */
27   height: 80px; /* Sets the header height to 80 pixels */
28   background: rgba(0, 0, 0, 0.8); /* Applies a semi-transparent black background */
29   z-index: 5; /* Sets a high z-index to ensure the header appears above other elements */
30   display: flex; /* Uses flexbox for aligning child elements */
31   align-items: center; /* Vertically centres the header content */
32   justify-content: center; /* Horizontally centres the header content */
33   border-bottom: 1px solid #ddd; /* Adds a subtle bottom border to the header */
34 }
35 .page-header h1 { /* Targets h1 elements within the page header */
36   color: #fff; /* Sets the heading text colour to white */
37 }
38 .page-header img.logo { /* Targets the image with the class 'logo' within the header */
39   position: absolute; /* Positions the logo absolutely within the header */
40   right: 20px; /* Places the logo 20 pixels from the right edge */
41   top: 50%; /* Places the logo at 50% from the top of the header */
42   transform: translateY(-50%); /* Adjusts the logo position to vertically centre it */
43   width: 100px; /* Sets the width of the logo to 100 pixels */
44 }
45 /* Burger icon styling */
46 .burger-menu { /* Targets the element with the class 'burger-menu' */
47   position: fixed; /* Fixes the position of the burger menu on the screen */
48   top: 25px; /* Positions it 25 pixels from the top */
49   left: 20px; /* Positions it 20 pixels from the left */
50   z-index: 10; /* Sets a high z-index so it appears above most elements */
51   font-size: 30px; /* Sets the font size of the icon to 30 pixels */
52   color: #fff; /* Sets the icon colour to white */
53   cursor: pointer; /* Changes the cursor to a pointer when hovering */
54 }
```

```

55 /* =====
56 4) Sidebar (Styles the sidebar which is hidden off-screen by default)
57 ===== */
58 .sidebar { /* Targets the element with the class 'sidebar' */
59   position: fixed; /* Fixes the sidebar's position relative to the screen */
60   top: 80px; /* Positions the sidebar just below the header (80px from the top) */
61   left: 0; /* Aligns the sidebar to the left edge */
62   width: 160px; /* Sets the sidebar width to 160 pixels */
63   height: calc(100% - 80px); /* Sets the sidebar height to fill the screen minus the header height */
64   background-color: rgba(0, 0, 0, 0.5); /* Applies a semi-transparent black background */
65   z-index: 4; /* Sets a stacking order for the sidebar */
66   transform: translateX(-100%); /* Initially moves the sidebar completely off-screen to the left */
67   transition: transform 0.3s ease; /* Animate the sidebar sliding in and out over 0.3 seconds */
68 }
69 /* When the sidebar is open, slide it in */
70 .sidebar.open { /* Targets the sidebar when its 'open' */
71   transform: translateX(0); /* Resets the transform so the sidebar slides into view */
72 }
73 /* =====
74 5) Sidebar Links (Styles the links within the sidebar)
75 ===== */
76 .sidebar a { /* Targets all anchor (<a>) elements inside the sidebar */
77   display: flex; /* Uses flexbox for link layout */
78   justify-content: center; /* Centres the link content horizontally */
79   align-items: center; /* Centres the link content vertically */
80   height: 60px; /* Sets a fixed height for each link */
81   padding: 0; /* Removes any padding */
82   color: #f0f0f0; /* Sets the text colour to a light grey */
83   text-decoration: none; /* Removes the default underline from links */
84   font-size: 14px; /* Sets the font size to 14 pixels */
85   border-bottom: 1px solid #ddd; /* Adds a subtle bottom border for separation */
86 }
87 .sidebar a:hover { /* Targets sidebar links on hover */
88   background-color: rgba(255, 255, 255, 0.2); /* Applies a semi-transparent white background on hover */
89 }
90 .sidebar .exit-card { /* Targets the element with the class 'exit-card' within the sidebar */
91   position: absolute; /* Positions the exit card absolutely within the sidebar */
92   bottom: 0; /* Aligns it to the bottom of the sidebar */
93   width: 100%; /* Sets its width to fill the sidebar */
94   border-top: 1px solid #ddd; /* Adds a light grey top border for separation */
95   background-color: rgba(0, 0, 0, 0.5); /* Applies a semi-transparent black background */
96 }
97 .icon { /* Targets elements with the class 'icon' */
98   display: inline-block; /* Displays the icon as an inline block */
99   width: 20px; /* Sets a fixed width of 20 pixels */
100  height: 20px; /* Sets a fixed height of 20 pixels */
101  margin-right: 8px; /* Adds an 8-pixel margin to the right */
102  vertical-align: middle; /* Vertically aligns the icon in the middle of adjacent text */
103 }

```

```

104 /* =====
105   6) Dropdown Menus (Styles dropdown menu containers and content)
106   ===== */
107 .dropdown { /* Targets elements with the class 'dropdown' */
108   position: relative; /* Sets position to relative, to position dropdown content properly */
109 }
110 .dropdown-content { /* Targets the dropdown content */
111   display: none; /* Hides the dropdown content by default */
112   opacity: 0; /* Sets the opacity to 0 (fully transparent) by default */
113   position: absolute; /* Positions the dropdown content absolutely within its container */
114   top: 0; /* Aligns the top of the dropdown content with the container */
115   left: 100%; /* Positions the dropdown content to the right of its container */
116   width: 200px; /* Sets the width of the dropdown content to 200 pixels */
117   background-color: rgba(0, 0, 0, 0.3); /* Applies a semi-transparent black background */
118   border: 1px solid rgba(255, 255, 255, 0.3); /* Adds a semi-transparent white border */
119   border-radius: 4px; /* Rounds the corners by 4 pixels */
120   z-index: 6; /* Sets a high z-index to ensure the dropdown appears above other elements */
121   transition: opacity 0.15s ease-out; /* Transitions the opacity for a fade-in/out effect */
122 }
123 .dropdown-content.visible { /* Targets the dropdown content when it has the class 'visible' */
124   display: block; /* Changes Displays to block so it is visible */
125   opacity: 1; /* Sets the opacity to 1 (fully opaque) */
126 }
127 .dropdown-content a { /* Targets anchor (<a>) elements inside the dropdown content */
128   display: flex; /* Uses flexbox layout for the link */
129   align-items: center; /* Centres the items vertically within the link */
130   justify-content: flex-start; /* Aligns items to the start (left) horizontally */
131   gap: 12px; /* Adds a 12-pixel gap between items inside the link */
132   padding: 8px 10px; /* Adds 8 pixels of vertical padding and 10 pixels of horizontal padding */
133   font-size: 14px; /* Sets the font size to 14 pixels */
134   color: #f0f0f0; /* Sets the link text colour to a light grey */
135   text-decoration: none; /* Removes the default underline from the link */
136 }
137 /* =====
138   7) Content Area (Styles the main content area)
139   ===== */
140 .content { /* Targets elements with the class 'content' */
141   margin-top: 80px; /* Adds a top margin of 80 pixels to avoid overlap with the fixed header */
142   display: flex; /* Uses flexbox layout for the content area */
143   justify-content: center; /* Centres the content horizontally */
144   align-items: center; /* Centres the content vertically */
145   height: calc(100vh - 80px); /* Sets the height to the screen height minus 80 pixels (header height) */
146   position: relative; /* Sets position to relative for further positioning needs */
147   z-index: 1; /* Sets a z-index to establish stacking context */
148 }
149 /* =====
150   8) Card Container (Grid Layout - Styles the container that holds cards in a grid)
151   ===== */
152 .card-container { /* Targets elements with the class 'card-container' */
153   display: grid; /* Uses Grid layout to arrange cards */
154   grid-template-columns: repeat(auto-fit, minmax(250px, 1fr)); /* Defines responsive columns that fit a minimum of 250px and expand equally */
155   gap: 20px; /* Sets a 20-pixel gap between grid items */
156   max-width: 1000px; /* Limits the maximum width of the container to 1000 pixels */
157 }

```

```

158 /* =====
159 9) Card Styling (Styles individual card elements)
160 ===== */
161 .card { /* Targets elements with the class 'card' */
162   display: flex; /* Uses flexbox layout inside the card */
163   flex-direction: column; /* Stacks card contents vertically */
164   justify-content: center; /* Centres the contents vertically within the card */
165   align-items: stretch; /* Stretches child elements to fill the card's width */
166   width: 100%; /* Sets the card's width to 100% of its container */
167   max-width: 500px; /* Limits the card's maximum width to 500 pixels */
168   margin: 0; /* Removes any default margin */
169   padding: 20px 10px; /* Adds 20px padding on top/bottom and 10px on left/right */
170   background-color: rgba(0, 0, 0, 0.95); /* Sets a nearly opaque black background */
171   border-radius: 8px; /* Rounds the corners of the card by 8 pixels */
172   box-shadow: 0 4px 30px rgba(0, 0, 0, 0.8); /* Applies a box shadow for depth */
173 }
174 /* =====
175 2) Menu Items in the Card (Styles the menu items inside cards)
176 ===== */
177 .menu-item { /* Targets elements with the class 'menu-item' */
178   display: block; /* Displays the menu item as a block element */
179   width: 100%; /* Sets the menu item's width to 100% of its container */
180   background-color: rgba(255, 255, 255, 0.05); /* Apply a very subtle white background */
181   border-radius: 8px; /* Rounds the corners of the menu item */
182   margin: 10px auto; /* Adds a 10-pixel vertical margin and centre horizontally */
183   transition: background-color 0.3s; /* Smoothly transition the background colour on hover */
184   padding: 15px; /* Adds 15 pixels of padding inside the menu item */
185   box-sizing: border-box; /* Ensures padding is included in the width calculation */
186 }
187 /* Hover effect for the entire menu-item */
188 .menu-item:hover { /* When the menu item is hovered */
189   background-color: rgba(255, 255, 255, 0.2); /* Change the background colour to a slightly more opaque white */
190 }
191 /* Menu Item Link Styling */
192 .menu-item a { /* Targets anchor (<a>) elements within a menu item */
193   display: flex; /* Uses flexbox layout for the link */
194   align-items: center; /* Vertically centre the items within the link */
195   gap: 12px; /* Adds a 12-pixel gap between items inside the link */
196   color: #fff; /* Sets the link text colour to white */
197   font-size: 36px; /* Sets a large font size (36px) for the link text */
198   text-decoration: none; /* Removes the default underline from the link */
199   position: relative; /* Sets the link position to relative */
200   transition: color 0.3s; /* Smoothly transition the text colour on hover */
201 }
202 .menu-icon { /* Targets elements with the class 'menu-icon' */
203   width: 64px; /* Sets the width of the icon to 64 pixels */
204   height: 64px; /* Sets the height of the icon to 64 pixels */
205   object-fit: contain; /* Ensures the icon maintains its aspect ratio and fits inside its container */
206 }
207 .menu-text { /* Targets the container with the class 'menu-text' */
208   display: flex; /* Uses flexbox layout */
209   flex-direction: column; /* Stack text elements vertically */
210 }
211 /* Title styling */
212 .menu-title { /* Targets elements with the class 'menu-title' */
213   font-size: 24px; /* Sets the font size to 24 pixels */
214   margin: 0 0 4px 0; /* Removes margin on top/right/left and Adds 4px margin on the bottom */
215 }

```

```

216 /* Subtitle styling */
217 .menu-subtitle { /* Targets elements with the class 'menu-subtitle' */
218   font-size: 14px; /* Sets the font size to 14 pixels */
219   color: #ccc; /* Sets the subtitle text colour to a lighter grey */
220   margin: 0; /* Removes any margin from the subtitle */
221 }
222 .exit-link a { /* Targets anchor (<a>) elements within an element with the class 'exit-link' */
223   font-weight: bold; /* Makes the text bold */
224   font-size: 36px; /* Sets the font size to 36 pixels */
225   text-align: center; /* Centres the text inside the exit link */
226 }
227 /* ===== */
228 10) Default Forms & Messages Styling (Styles default forms and success/failure messages)
229 ===== */
230 form { /* Targets all form elements */
231   background-color: white; /* Sets the form background colour to white */
232   padding: 20px; /* Adds 20 pixels of padding inside the form */
233   border-radius: 10px; /* Rounds the corners of the form by 10 pixels */
234   width: 400px; /* Sets a fixed width of 400 pixels for the form */
235   margin-top: 20px; /* Adds a 20-pixel top margin to the form */
236   display: inline-block; /* Displays the form as an inline-block element */
237   text-align: left; /* Align the text inside the form to the left */
238 }
239 h1 { /* Targets all h1 elements */
240   text-align: center; /* Centres the heading text */
241   color: #333; /* Sets the heading text colour to a dark grey */
242   margin-top: 20px; /* Adds a 20-pixel top margin to the heading */
243 }
244 .inputbox { /* Targets elements with the class 'inputbox' */
245   margin-bottom: 15px; /* Adds a 15-pixel bottom margin to separate input fields */
246 }
247 .inputbox label { /* Targets label elements within an inputbox */
248   display: block; /* Displays the label as a block element */
249   margin-bottom: 5px; /* Adds a 5-pixel bottom margin for spacing */
250   font-weight: bold; /* Makes the label text bold */
251   color: #333; /* Sets the label text colour to dark grey */
252 }
253 .inputbox input[type="text"],
254 .inputbox input[type="email"],
255 .inputbox input[type="url"],
256 .inputbox input[type="date"],
257 .inputbox input[type="number"] { /* Targets various text-based input fields within an inputbox */
258   width: 100%; /* Sets the input width to 100% of its container */
259   padding: 10px; /* Adds 10 pixels of padding inside the input field */
260   border: 1px solid #ccc; /* Adds a 1-pixel solid light grey border around the input */
261   border-radius: 5px; /* Rounds the corners of the input by 5 pixels */
262   box-sizing: border-box; /* Includes the padding and border in the input's total width */
263 }
264 .inputbox select { /* Targets select elements within an inputbox */
265   width: 100%; /* Sets the select width to 100% of its container */
266   padding: 10px; /* Adds 10 pixels of padding inside the select */
267   border: 1px solid #ccc; /* Adds a 1-pixel solid light grey border around the select */
268   border-radius: 5px; /* Rounds the corners of the select by 5 pixels */
269   box-sizing: border-box; /* Includes padding and border in the select's width */
270   font-size: 14px; /* Sets the font size of the select text to 14 pixels */
271   background-color: white; /* Sets the select background colour to white */
272 }

```

```

273 fieldset { /* Targets fieldset elements, used for grouping form controls */
274   border: 1px solid #ccc; /* Adds a 1-pixel solid light grey border around the fieldset */
275   padding: 15px; /* Adds 15 pixels of padding inside the fieldset */
276   margin-bottom: 20px; /* Adds a 20-pixel bottom margin to separate it from following elements */
277   border-radius: 5px; /* Rounds the corners of the fieldset by 5 pixels */
278 }
279 legend { /* Targets legend elements within a fieldset */
280   font-weight: bold; /* Makes the legend text bold */
281   font-size: large; /* Sets a larger font size for the legend */
282   padding: 0 10px; /* Adds 10 pixels of horizontal padding to the legend */
283   color: #333; /* Sets the legend text colour to dark grey */
284 }
285 .myButton { /* Targets elements with the class 'myButton' that group buttons */
286   display: flex; /* Uses flexbox layout to arrange buttons horizontally */
287   justify-content: space-between; /* Distribute the buttons evenly with space between them */
288 }
289 .myButton input[type="submit"],
290 .myButton input[type="reset"],
291 .myButton input[type="button"] { /* Targets different types of buttons within the .myButton container */
292   background-color: #444; /* Sets a dark grey background for the buttons */
293   color: white; /* Sets the button text colour to white */
294   padding: 10px 15px; /* Adds padding inside the buttons */
295   border: none; /* Removes any default border from the buttons */
296   border-radius: 5px; /* Rounds the corners of the buttons by 5 pixels */
297   width: 48%; /* Sets the buttons' width to 48% of their container */
298   font-weight: bold; /* Makes the button text bold */
299 }
300 .myButton input[type="submit"]:hover,
301 .myButton input[type="reset"]:hover,
302 .myButton input[type="button"]:hover { /* Targets hover state for buttons */
303   background-color: #666; /* Change the background colour on hover to a lighter grey */
304 }
305 .success { /* Targets elements with the class 'success' for success messages */
306   background-color: #dff0d8; /* Sets a light green background to indicate success */
307   border: 1px solid #d6e9c6; /* Adds a matching light green border */
308   color: #3c763d; /* Sets the text colour to a dark green */
309   padding: 15px; /* Adds 15 pixels of padding inside the success message */
310   margin: 20px auto; /* Centres the message horizontally with 20 pixels vertical margin */
311   border-radius: 5px; /* Rounds the corners of the success message by 5 pixels */
312   width: 400px; /* Sets a fixed width of 400 pixels for the success message */
313   text-align: center; /* Centres the text within the success message */
314   font-size: 16px; /* Sets the font size of the success message to 16 pixels */
315 }
316 .error { /* Targets elements with the class 'error' for error messages */
317   background-color: #f2dede; /* Sets a light red background to indicate an error */
318   border: 1px solid #ebccdd; /* Adds a red border to further highlight the error */
319   color: #a94442; /* Sets the text colour to a dark red */
320   padding: 15px; /* Adds 15 pixels of padding inside the error message */
321   margin: 20px auto; /* Centres the error message horizontally with 20 pixels vertical margin */
322   border-radius: 5px; /* Rounds the corners of the error message by 5 pixels */
323   width: 400px; /* Sets a fixed width of 400 pixels for the error message */
324   text-align: center; /* Centres the text within the error message */
325   font-size: 16px; /* Sets the font size of the error message to 16 pixels */
326 }

```

```

327 /* =====
328   Delete Page (Specific styling for the Delete pages)
329 ===== */
330 .content.delete-page { /* Targets elements with both 'content' and 'delete-page' classes */
331   margin-left: auto; /* Automatically adjusts the left margin to centre the content */
332   display: flex; /* Uses flexbox layout for this content area */
333   flex-direction: column; /* Arranges child elements vertically */
334   justify-content: center; /* Centres child elements vertically */
335   align-items: center; /* Centres child elements horizontally */
336   height: calc(100vh - 60px); /* Sets the height to fill the screen minus 60 pixels */
337 }
338 .content.delete-page form { /* Targets form elements within the Delete page */
339   box-shadow: 0 4px 30px rgba(0, 0, 0, 0.8); /* Apply a box shadow to create depth for the form */
340 }
341 .content.delete-page p { /* Targets paragraph elements within the Delete page */
342   font-weight: bold; /* Makes the paragraph text bold */
343   font-style: italic; /* Italicise the paragraph text */
344   color: black; /* Sets the paragraph text colour to black */
345 }
346 /* Centres the delete button within its container */
347 .deleteButton { /* Targets elements with the class 'deleteButton' */
348   display: flex; /* Uses flexbox layout to centre the delete button */
349   justify-content: center; /* Centres the button horizontally */
350 }
351 .deleteButton input[type="submit"] { /* Targets the submit button inside the delete button container */
352   background-color: #444; /* Sets a dark grey background for the delete button */
353   color: white; /* Sets the delete button text colour to white */
354   padding: 10px 15px; /* Adds padding inside the delete button */
355   border: none; /* Removes any default border */
356   border-radius: 5px; /* Rounds the corners of the delete button by 5 pixels */
357   width: 48%; /* Sets the width of the delete button to 48% of its container */
358 }
359 .deleteButton input[type="submit"]:hover { /* Targets the hover state for the delete submit button */
360   background-color: #666; /* Darken the background colour on hover */
361 }
362 /* =====
363   Add Page (Specific styling for the Add pages)
364 ===== */
365 .content.add-page { /* Targets elements with both 'content' and 'add-page' classes */
366   margin-left: auto; /* Automatically adjusts left margin to centre the content */
367   display: flex; /* Uses flexbox layout for this content area */
368   flex-direction: column; /* Arranges child elements vertically */
369   justify-content: center; /* Centres child elements vertically */
370   align-items: center; /* Centres child elements horizontally */
371   height: calc(100vh - 60px); /* Sets the height to fill the screen minus 60 pixels */
372 }
373 .add-page { /* Targets elements with the class 'add-page' */
374   text-align: center; /* Centres the text on the Adds page */
375 }
376 .add-page form,
377 .add-page .card-container { /* Targets both form elements and card containers on the Add page */
378   margin: 20px auto; /* Centres these elements horizontally with a 20-pixel vertical margin */
379   box-shadow: 0 4px 30px rgba(0, 0, 0, 0.8); /* Apply a box shadow for visual depth */
380 }
381 .add-page .card-container { /* Further Targets the card container on the Add page */
382   margin-top: 20px; /* Adds extra top margin to separate it from preceding content */
383 }
384 /* Page Under Construction Message Styling */
385 .construction-message { /* Targets elements with the class 'construction-message' */
386   text-align: center; /* Centres the construction message text */
387   font-size: 45px; /* Sets a large font size of 45 pixels */
388   font-weight: bold; /* Sets a font weight of bold for visibility */
389   color: #333; /* Sets the text colour to a dark grey */
390   padding: 20px; /* Adds 20 pixels of padding around the message */
391   max-width: 1000px; /* Limit the maximum width of the message container to 1000 pixels */
392   margin: 0 auto; /* Centres the message container horizontally */
393 }

```

```

394 /* =====
395 Amend/View Page Form Styling (Specific styling for some of the Amend/View page forms)
396 ===== */ /* Section comment: */
397 .content.amend-view-page { /* Targets elements with both 'content' and 'amend-view-page' classes */
398   margin-left: auto; /* Automatically adjusts left margin to centre the content */
399   display: flex; /* Uses flexbox layout for the content area */
400   flex-direction: column; /* Arranges child elements vertically */
401   align-items: center; /* Centres child elements horizontally */
402   height: auto; /* Let the height adjust automatically based on content */
403   padding-top: 60px; /* Adds 60 pixels of padding at the top to separate from other elements */
404 }
405 .amend-view-page form { /* Targets form elements on the Amend/View page */
406   width: 900px; /* Sets a fixed width of 900 pixels for the form */
407   margin: 150px auto; /* Centres the form horizontally with a 150-pixel vertical margin (Works on my home display) */
408   display: block; /* Displays the form as a block element */
409   text-align: left; /* Align the text within the form to the left */
410   box-shadow: 0 4px 30px rgba(0, 0, 0, 0.8); /* Apply a box shadow to give the form depth */
411 }
412 .amend-view-page .columns { /* Targets the container that splits the form into columns */
413   display: flex; /* Uses flexbox layout for columns */
414   justify-content: space-between; /* Distribute columns evenly with space between */
415   gap: 20px; /* Adds a 20-pixel gap between the columns */
416 }
417 .amend-view-page .left-column,
418 .amend-view-page .right-column { /* Targets both left and right columns in the form */
419   display: flex; /* Uses flexbox layout within each column */
420   flex-direction: column; /* Stack the column's elements vertically */
421   gap: 10px; /* Adds a 10-pixel gap between elements in each column */
422 }
423 .amend-view-page .inputbox label { /* Targets labels within input boxes on the Amend/View page */
424   display: inline-block; /* Displays the label as an inline block element */
425   margin-bottom: 0; /* Removes any bottom margin from the label */
426   font-weight: bold; /* Makes the label text bold */
427   white-space: nowrap; /* Prevents the label text from wrapping to a new line */
428   margin-right: 15px; /* Adds a 15-pixel right margin to separate from the input field */
429 }
430 .amend-view-page .inputbox { /* Targets input boxes on the Amend/View page */
431   display: flex; /* Uses flexbox to arrange the label and input in a row */
432   justify-content: space-between; /* Spreads the label and input out with space between */
433   align-items: center; /* Vertically centres the label and input */
434   white-space: nowrap; /* Prevents wrapping within the input box */
435 }
436 .amend-view-page .inputbox input,
437 .amend-view-page .inputbox select { /* Targets input and select elements within an inputbox */
438   width: 200px; /* Sets a fixed width of 200 pixels for these fields */
439   box-sizing: border-box; /* Includes padding and border in the total width */
440   margin-left: auto; /* Automatically adjusts left margin to push the element to the right */
441 }
442
443 .amend-view-page .centered-select { /* Targets the container for centred select dropdowns */
444   display: flex; /* Uses flexbox layout for centring */
445   justify-content: center; /* Centres the dropdown horizontally */
446   margin-bottom: 20px; /* Adds a 20-pixel bottom margin for spacing */
447 }
448 .amend-view-page .centered-select select { /* Targets select elements within the centred select container */
449   width: 220px; /* Sets a fixed width of 220 pixels for the dropdown */
450 }
451 .amend-view-page .myButton { /* Targets the container for form buttons on the Amend/View page */
452   display: flex; /* Uses flexbox layout to arrange buttons horizontally */
453   justify-content: center; /* Centres the buttons horizontally */
454   gap: 20px; /* Adds a 20-pixel gap between the buttons */
455   margin-top: 20px; /* Adds a 20-pixel top margin to separate from the form content */
456 }

```

```

457 /* =====
458   Place Orders Page (Specific styling for the Place Orders page)
459   ===== */
460 .place-orders-content { /* Targets elements with the class 'place-orders-content' */
461   margin: 150px auto 0; /* Adds a 150-pixel top margin, centre horizontally, and no bottom margin */
462   max-width: 1400px; /* Limit the maximum width to 1400 pixels */
463   display: flex; /* Uses flexbox layout for this container */
464   flex-direction: column; /* Arranges child elements vertically */
465   align-items: center; /* Centres child elements horizontally */
466   position: relative; /* Sets position to relative for any absolutely positioned children */
467 }
468 .place-orders-container { /* Targets the container for order-related elements */
469   display: flex; /* Uses flexbox layout to arrange child elements */
470   justify-content: center; /* Centres the child elements horizontally */
471   gap: 30px; /* Adds a 30-pixel gap between child elements */
472   width: 100%; /* Sets the container width to 100% of its parent */
473   max-width: 1200px; /* Limit the maximum width to 1200 pixels */
474   margin: 80px 0; /* Adds an 80-pixel vertical margin */
475   box-shadow: 0 4px 30px rgba(0, 0, 0, 0.8); /* Apply a box shadow for depth */
476   border-radius: 10px; /* Rounds the corners of the container by 10 pixels */
477 }
478 .place-orders-messages { /* Targets the container for order messages */
479   width: 100%; /* Sets the width to 100% of the parent */
480   max-width: 1200px; /* Limit the maximum width to 1200 pixels */
481   margin: 20px 0; /* Adds a 20-pixel vertical margin */
482   text-align: center; /* Centres the message text */
483 }
484 /* Place Orders Error & Success Messages */
485 .place-orders-error { /* Targets error message elements on the Place Orders page */
486   background: #fff0f0; /* Sets a light, pinkish background for errors */
487   border-left: 4px solid #cc0000; /* Adds a 4-pixel solid red border on the left */
488   padding: 15px 25px; /* Adds 15 pixels vertical and 25 pixels horizontal padding */
489   color: #cc0000; /* Sets the text colour to red for errors */
490   margin-bottom: 25px; /* Adds a 25-pixel bottom margin */
491   border-radius: 4px; /* Rounds the corners by 4 pixels */
492   display: inline-block; /* Displays the error message as an inline-block element */
493 }
494 .place-orders-success { /* Targets success message elements on the Place Orders page */
495   background: #e5ffe5; /* Sets a light green background for success messages */
496   border: 1px solid green; /* Adds a 1-pixel solid green border */
497   color: darkgreen; /* Sets the text colour to dark green */
498   padding: 25px; /* Adds 25 pixels of padding */
499   margin: 20px 0; /* Adds a 20-pixel vertical margin */
500   width: 500px; /* Sets a fixed width of 500 pixels for the message container */
501   text-align: center; /* Centres the text within the success message */
502   border-radius: 8px; /* Rounds the corners by 8 pixels */
503 }
504 .place-orders-success h3 { /* Targets h3 elements within a success message */
505   margin-top: 0; /* Removes any top margin */
506   color: #2c3e50; /* Sets the header text colour to a dark blue-grey */
507   padding-bottom: 10px; /* Adds 10 pixels of padding to the bottom */
508   border-bottom: 2px solid #c8e6c9; /* Adds a 2-pixel solid bottom border for separation */
509 }
510 .place-orders-success ul { /* Targets unordered lists within the success message */
511   list-style: none; /* Removes default list bullet points */
512   padding: 0; /* Removes any padding */
513   margin: 15px 0; /* Adds a 15-pixel vertical margin */
514 }
515 .place-orders-success li { /* Targets list items within the success message */
516   padding: 8px 0; /* Adds 8 pixels of vertical padding */
517   border-bottom: 1px solid #c8e6c9; /* Adds a 1-pixel solid bottom border for separation */
518   margin: 0 auto; /* Centres the list item horizontally */
519   width: fit-content; /* Sets the width to fit the content */
520 }

```

```

521 /* Place Orders Form & Table */
522 .place-orders-form,
523 .place-orders-table { /* Targets both the form and table containers on the Place Orders page */
524   flex: 1; /* Allow these containers to expand and fill available space */
525   padding: 25px; /* Adds 25 pixels of padding inside the containers */
526   background: #ffffff; /* Sets the background colour to white */
527   border-radius: 10px; /* Rounds the corners by 10 pixels */
528   min-width: 300px; /* Sets a minimum width of 300 pixels */
529 }
530 .place-orders-form form { /* Targets the form within the Place Orders section */
531   display: block; /* Displays the form as a block element */
532   margin: 0 auto; /* Centres the form horizontally */
533   width: 80%; /* Sets the form width to 80% of its container */
534   text-align: left; /* Align the text inside the form to the left */
535 }
536 .place-orders-form .columns { /* Targets the container that splits the order form into columns */
537   display: flex; /* Uses flexbox layout for the columns */
538   justify-content: space-between; /* Distribute the columns evenly with space between */
539   gap: 20px; /* Adds a 20-pixel gap between the columns */
540 }
541 .place-orders-form .left-column,
542 .place-orders-form .right-column { /* Targets both the left and right columns in the order form */
543   display: flex; /* Uses flexbox layout within each column */
544   flex-direction: column; /* Stack the column's elements vertically */
545   gap: 10px; /* Adds a 10-pixel gap between elements in the column */
546 }
547 /* Place Orders Input Box */
548 .place-orders-inputbox { /* Targets the input box container on the Place Orders page */
549   width: 100%; /* Sets the width to 100% of its container */
550   max-width: 400px; /* Limit the maximum width to 400 pixels */
551   margin: 0 auto 20px; /* Centres the container horizontally and Adds a 20-pixel bottom margin */
552 }
553 .place-orders-inputbox label { /* Targets labels within the orders input box */
554   display: block; /* Displays the label as a block element */
555   margin-bottom: 8px; /* Adds an 8-pixel bottom margin for spacing */
556   font-weight: 500; /* Sets a medium font weight for the label */
557 }
558 #calculatedCost { /* Targets the element with the ID 'calculatedCost' */
559   color: black; /* Sets the text colour to black */
560   font-weight: bold; /* Makes the text bold */
561 }
562 /* Place Orders Table */
563 .place-orders-table h3 { /* Targets h3 elements within the orders table section */
564   text-align: center; /* Centres the header text */
565   margin-bottom: 20px; /* Adds a 20-pixel bottom margin */
566   color: #333; /* Sets the header text colour to dark grey */
567 }
568 .place-orders-table table { /* Targets table elements on the Place Orders page */
569   width: 100%; /* Sets the table width to 100% of its container */
570   border-collapse: collapse; /* Collapse table borders for a cleaner look */
571   margin-top: 15px; /* Adds a 15-pixel top margin */
572 }
573 .place-orders-table th { /* Targets table header cells */
574   background: #f8f9fa; /* Sets a light background for the header cells */
575   padding: 12px; /* Adds 12 pixels of padding inside the header cells */
576   border-bottom: 2px solid #dee2e6; /* Adds a 2-pixel solid bottom border */
577   text-align: center; /* Centres the header text */
578   vertical-align: middle; /* Vertically centre the header text */
579   color: #333; /* Sets the header text colour to dark grey */
580 }
581 .place-orders-table td { /* Targets table data cells */
582   padding: 12px; /* Adds 12 pixels of padding inside each cell */
583   border-bottom: 1px solid #dee2e6; /* Adds a 1-pixel solid bottom border for separation */
584   text-align: center; /* Centres the cell content */
585   vertical-align: middle; /* Vertically centres the cell content */
586   color: #333; /* Sets the cell text colour to dark grey */
587 }
588

```

## Menus and Sidebar Written By: Mark Mukiza

Most of the page's menus have the same layout as the main menu. They have a sidebar to the left that's hidden and is toggled by the burger icon. There's a main card in the centre that has options for other pages such as Add, Delete, Amend/View. The code is exactly the same and the only difference is the text in the cards/menu items. Below we've included Main Menu and its code as well as screenshots of the other pages without their code as it's the same as main menu.

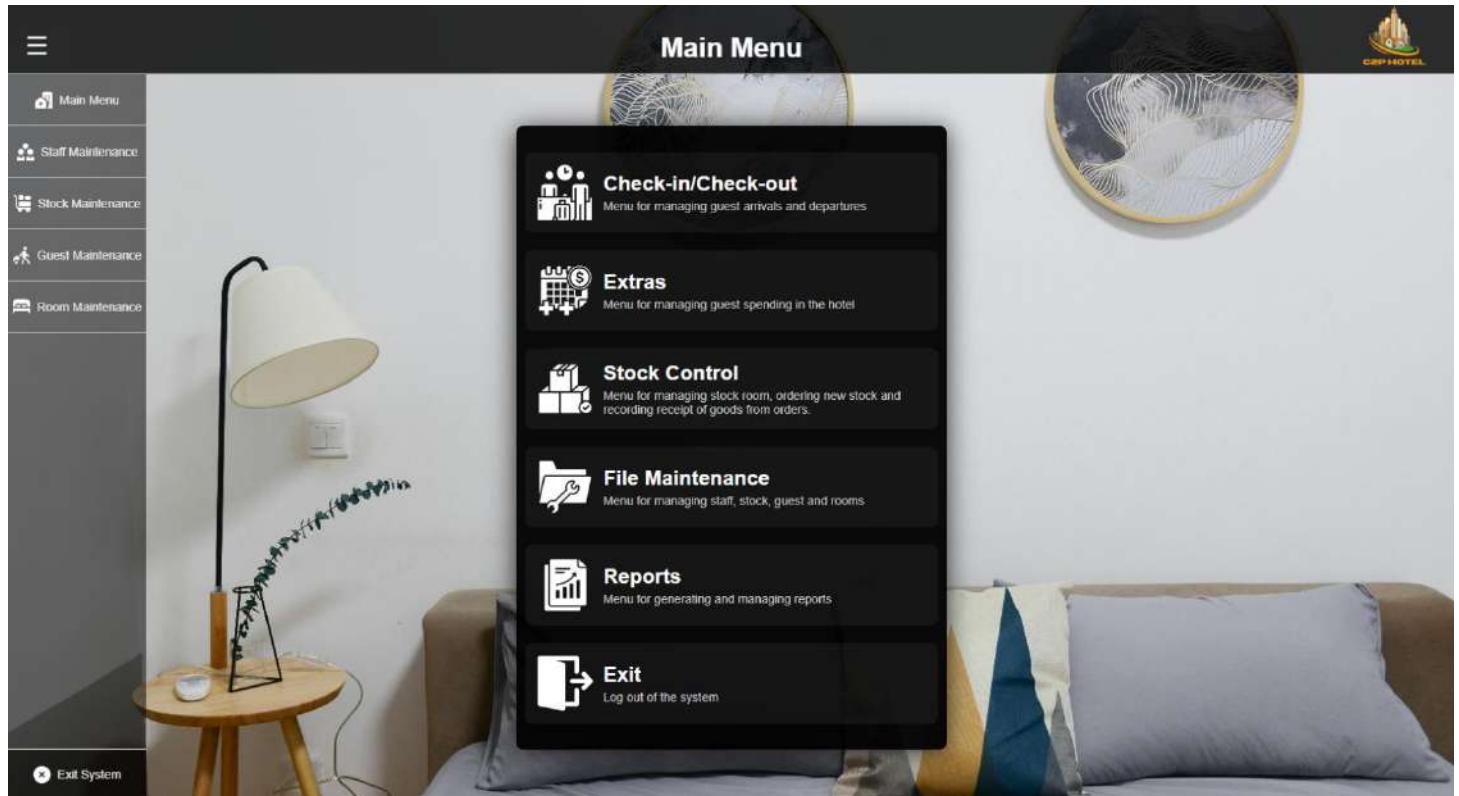


Figure 21. Main Menu and Sidebar

```
1 <!--
2     Name of screen:      home.html.php
3     Purpose of screen:  This is the main menu of the website.
4     Student ID:         C00166672
5     Name:                Mark Mukiiza
6     Date Written:       March 2025
7 -->
8 <!DOCTYPE html>
9 <html lang="en">
10 <head>
11   <meta charset="UTF-8">
12   <title>Home</title>
13   <link rel="stylesheet" href="styles.css">
14 </head>
15 <body>
16   <!-- Burger icon that will toggle the sidebar -->
17   <div class="burger-menu">&#9776;</div>
18   <div class="page-header">
19     <h1>Main Menu</h1>
20     <a href="home.html.php"></a>
21   </div>
22   <!-- Sidebar navigation -->
23   <?php include 'sidebar.php'; ?>
24   <div class="content">
25     <div class="card">
26       <div class="menu-item">
27         <a href="checkin-checkout/checkin-checkout.html">
28           
29           <div class="menu-text">
30             <h3 class="menu-title">Check-in/Check-out</h3>
31             <p class="menu-subtitle">Menu for managing guest arrivals and departures</p>
32           </div>
33         </a>
34       </div class="menu-item">
35       <div class="menu-item">
36         <a href="extra/extras.html.php">
37           
38           <div class="menu-text">
39             <h3 class="menu-title">Extras</h3>
40             <p class="menu-subtitle">Menu for managing guest spending in the hotel</p>
41           </div>
42         </a>
43       </div>
```

```
44 <div class="menu-item">
45     <a href="stockcontrol/stock_control.html.php">
46         
47         <div class="menu-text">
48             <h3 class="menu-title">Stock Control</h3>
49             <p class="menu-subtitle">Menu for managing stock room, ordering new stock and recording receipt of goods
50                 from orders.</p>
51         </div>
52     </a>
53 </div>
54 <div class="menu-item">
55     <a href="filemaintenance/file_maintenance.html.php">
56         
57         <div class="menu-text">
58             <h3 class="menu-title">File Maintenance</h3>
59             <p class="menu-subtitle">Menu for managing staff, stock, guest and rooms</p>
60         </div>
61     </a>
62 </div>
63 <div class="menu-item">
64     <a href="construction.html.php">
65         
66         <div class="menu-text">
67             <h3 class="menu-title">Reports</h3>
68             <p class="menu-subtitle">Menu for generating and managing reports</p>
69         </div>
70     </a>
71 </div>
72 <div class="menu-item">
73     <a href="login.html">
74         
75         <div class="menu-text">
76             <h3 class="menu-title">Exit</h3>
77             <p class="menu-subtitle">Log out of the system</p>
78         </div>
79     </a>
80 </div>
81 </div>
82 </div>
83 <script src="sidebar.js"></script>
84 </body>
85 </html>
```

```

1  <!--
2      Name of screen: sidebar.php
3      Purpose of screen: Holds the code for the sidebar and links to the rest of the web pages.
4      Student ID: C00166672
5      Name: Mark Mukiza
6      Date Written: March 2025
7  -->
8  <div class="sidebar">
9      <!-- Dropdown: Main Menu -->
10     <div class="dropdown" id="mainMenu">
11         <a href="/FinalWebsiteFolder/home.html.php">
12             
13             Main Menu
14         </a>
15         <div class="dropdown-content">
16             <a href="/FinalWebsiteFolder/checkin-checkout/checkin-checkout.html.php">
17                 
18                 Check-in/Check-out
19             </a>
20             <a href="/FinalWebsiteFolder/extras/">
21                 
22                 Extras
23             </a>
24             <a href="/FinalWebsiteFolder/stockcontrol/stock_control.html.php">
25                 
26                 Stock Control
27             </a>
28             <a href="/FinalWebsiteFolder/filemaintenance/file_maintenance.html.php">
29                 
30                 File Maintenance
31             </a>
32             <a href="/FinalWebsiteFolder/construction.html.php">
33                 
34                 Reports
35             </a>
36         </div>
37     </div>
38
39     <!-- Dropdown: Staff Maintenance -->
40     <div class="dropdown" id="staffMaintenance">
41         <a href="/FinalWebsiteFolder/filemaintenance/staff_maintenance.html.php">
42             
43             Staff Maintenance
44         </a>
45         <div class="dropdown-content">
46             <a href="/FinalWebsiteFolder/filemaintenance/staff/addStaff.html.php">
47                 
48                 Add Staff
49             </a>
50             <a href="/FinalWebsiteFolder/filemaintenance/staff/deleteStaff.html.php">
51                 
52                 Delete Staff
53             </a>
54             <a href="/FinalWebsiteFolder/filemaintenance/staff/amendViewStaff.html.php">
55                 
56                 Amend/View Staff
57             </a>
58         </div>
59     </div>
60

```

```

61 <!-- Dropdown: Stock Maintenance -->
62 <div class="dropdown" id="stockMaintenance">
63   <a href="/FinalWebsiteFolder/filemaintenance/stock_maintenance.html.php">
64     
65     Stock Maintenance
66   </a>
67   <div class="dropdown-content">
68     <a href="/FinalWebsiteFolder/filemaintenance/stock/addstock.html.php">
69       
70       Add Stock
71     </a>
72     <a href="/FinalWebsiteFolder/filemaintenance/stock/deleteStock.html.php">
73       
74       Delete Stock
75     </a>
76     <a href="/FinalWebsiteFolder/filemaintenance/stock/amendViewStock.html.php">
77       
78       Amend/View Stock
79     </a>
80   </div>
81 </div>
82
83 <!-- Dropdown: Guest Maintenance -->
84 <div class="dropdown" id="guestMaintenance">
85   <a href="/FinalWebsiteFolder/filemaintenance/guest_maintenance.html.php">
86     
87     Guest Maintenance
88   </a>
89   <div class="dropdown-content">
90     <a href="/FinalWebsiteFolder/filemaintenance/guest/addguest.html.php">
91       
92       Add Guest
93     </a>
94     <a href="/FinalWebsiteFolder/filemaintenance/guest/deleteGuest.html.php">
95       
96       Delete Guest
97     </a>
98     <a href="/FinalWebsiteFolder/filemaintenance/guest/ViewGuest.html.php">
99       
100      Amend/View Guest
101    </a>
102  </div>
103</div>
104
105 <!-- Dropdown: Room Maintenance -->
106 <div class="dropdown" id="roomMaintenance">
107   <a href="/FinalWebsiteFolder/filemaintenance/room_maintenance.html.php">
108     
109     Room Maintenance
110   </a>
111   <div class="dropdown-content">
112     <a href="/FinalWebsiteFolder/filemaintenance/room/addroom.html">
113       
114       Add Room
115     </a>
116     <a href="/FinalWebsiteFolder/filemaintenance/room/deleteRoom.html.php">
117       
118       Delete Room
119     </a>
120     <a href="/FinalWebsiteFolder/filemaintenance/room/AmendViewRoom.html.php">
121       
122       Amend/View Room
123     </a>
124   </div>
125 </div>
126
127 <a href="/FinalWebsiteFolder/login.html" class="exit-card">
128   
129   Exit System
130 </a>
131 </div>
132

```

```

1  /* Name of screen: sidebar.js
2  Purpose of screen: Javascript for the sidebar
3  Student ID: C00166672
4  Name: Mark Mukiza
5  Date Written: March 2025
6 */
7  document.addEventListener('DOMContentLoaded', function () {
8      // ===== SIDEBAR =====
9      const burger = document.querySelector('.burger-menu');
10     const sidebar = document.querySelector('.sidebar');
11
12     // Toggle sidebar on burger click
13     burger.addEventListener('click', (e) => {
14         e.stopPropagation();
15         sidebar.classList.toggle('open');
16     });
17
18     // Close sidebar when clicking outside
19     document.addEventListener('click', (e) => {
20         if (!sidebar.contains(e.target) && !e.target.classList.contains('burger-menu')) {
21             sidebar.classList.remove('open');
22         }
23     });
24
25     // ===== DROPPDOWNS =====
26     const dropdowns = document.querySelectorAll('.dropdown');
27     let activeDropdown = null;
28
29     dropdowns.forEach(dropdown => {
30         const content = dropdown.querySelector('.dropdown-content');
31
32         // Show dropdown on hover
33         dropdown.addEventListener('mouseenter', () => {
34             if (activeDropdown) {
35                 activeDropdown.classList.remove('visible');
36             }
37             content.classList.add('visible');
38             activeDropdown = content;
39         });
40
41         // Hide dropdown with delay
42         dropdown.addEventListener('mouseleave', () => {
43             setTimeout(() => {
44                 if (!content.matches(':hover')) {
45                     content.classList.remove('visible');
46                     activeDropdown = null;
47                 }
48             }, 10);
49         });
50
51         // Keep dropdown open when hovering its content
52         content.addEventListener('mouseenter', () => {
53             clearTimeout();
54             content.classList.add('visible');
55         });
56
57         // Hide dropdown when leaving its content
58         content.addEventListener('mouseleave', () => {
59             content.classList.remove('visible');
60             activeDropdown = null;
61         });
62     });
63 });

```

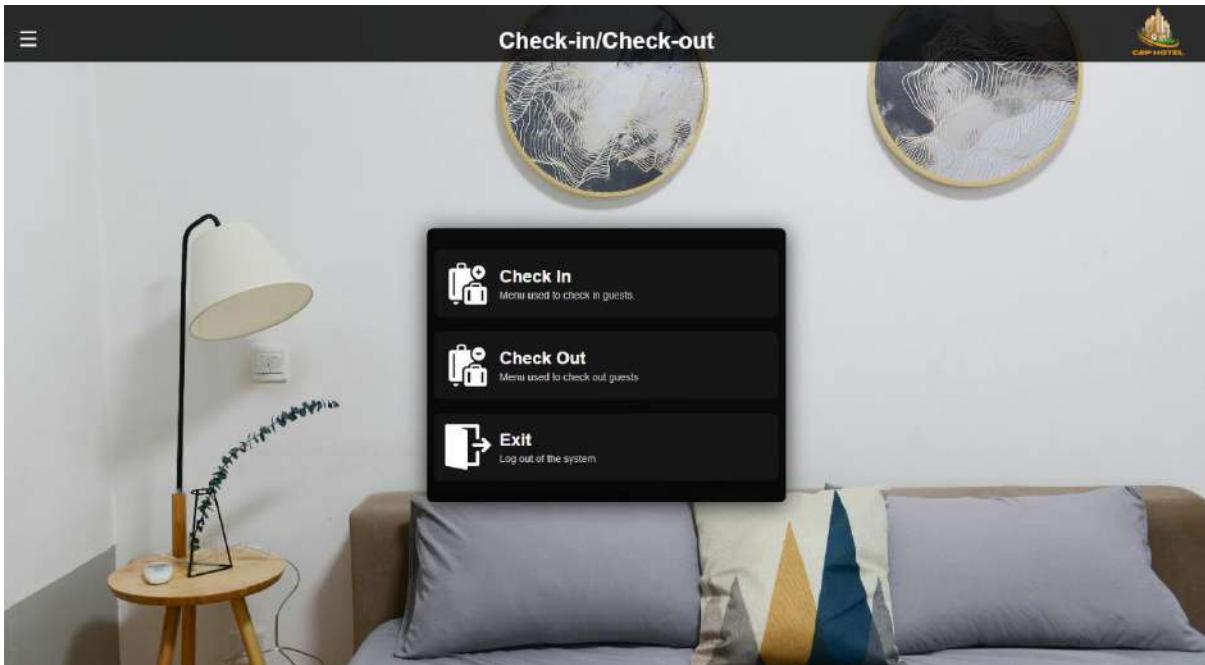


Figure 22. Check in/Check out Menu Screen

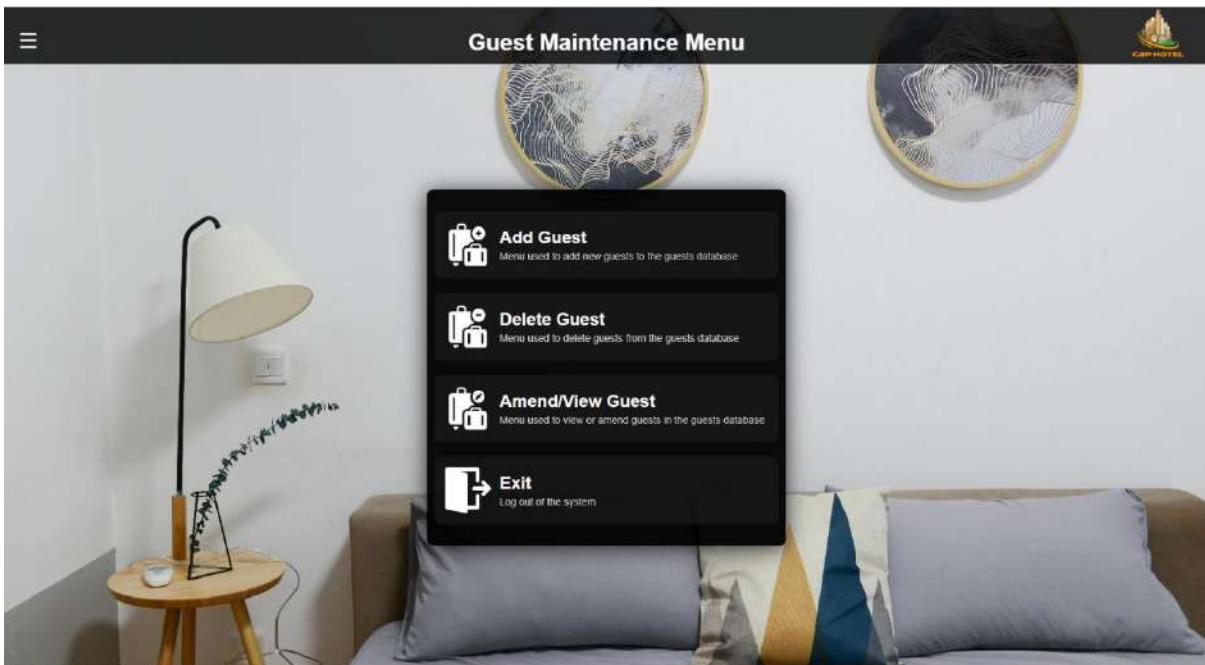


Figure 23. Guest Maintenance Menu Screen

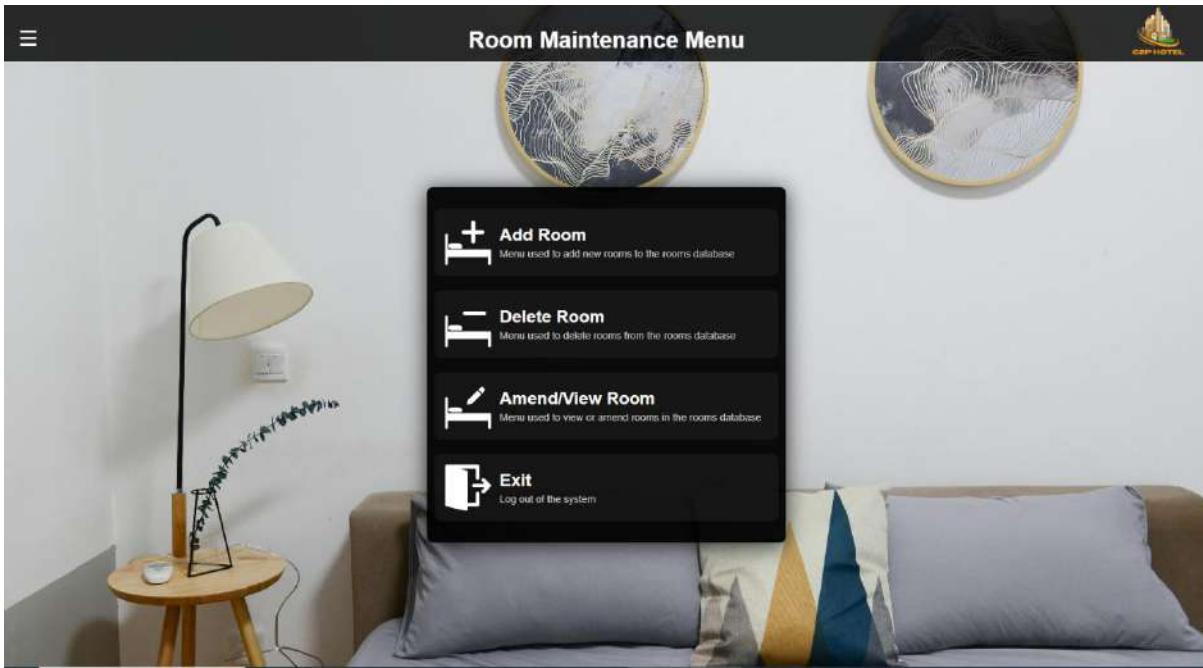


Figure 24. Room Maintenance Menu Screen

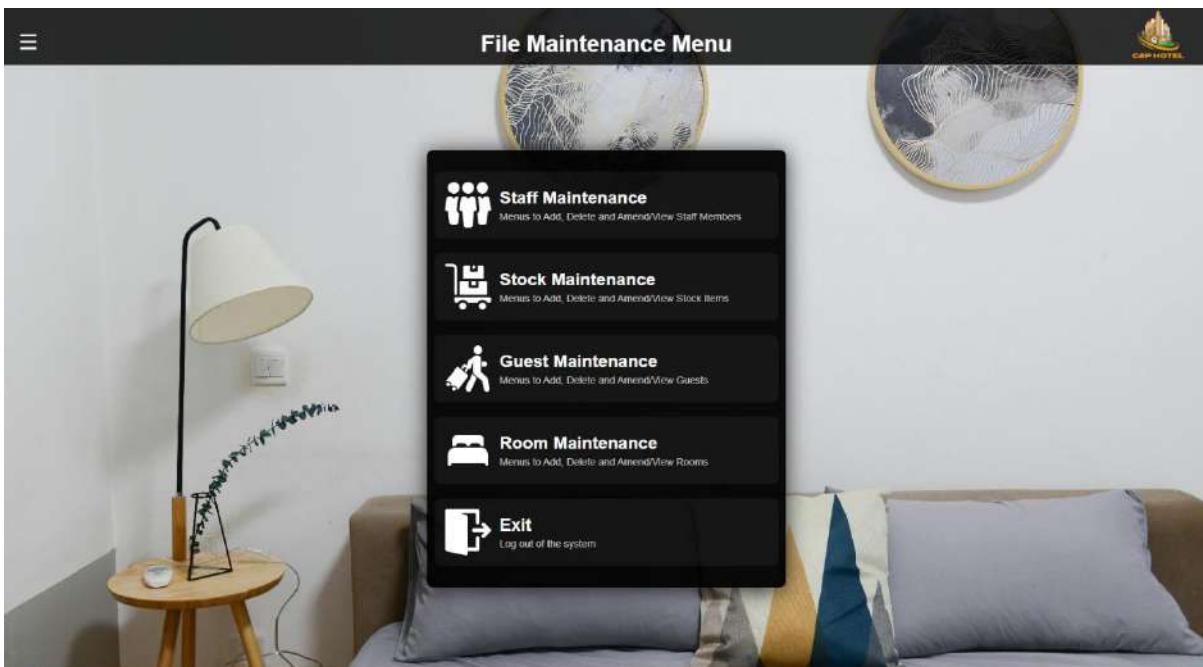


Figure 25. File Maintenance Menu Screen

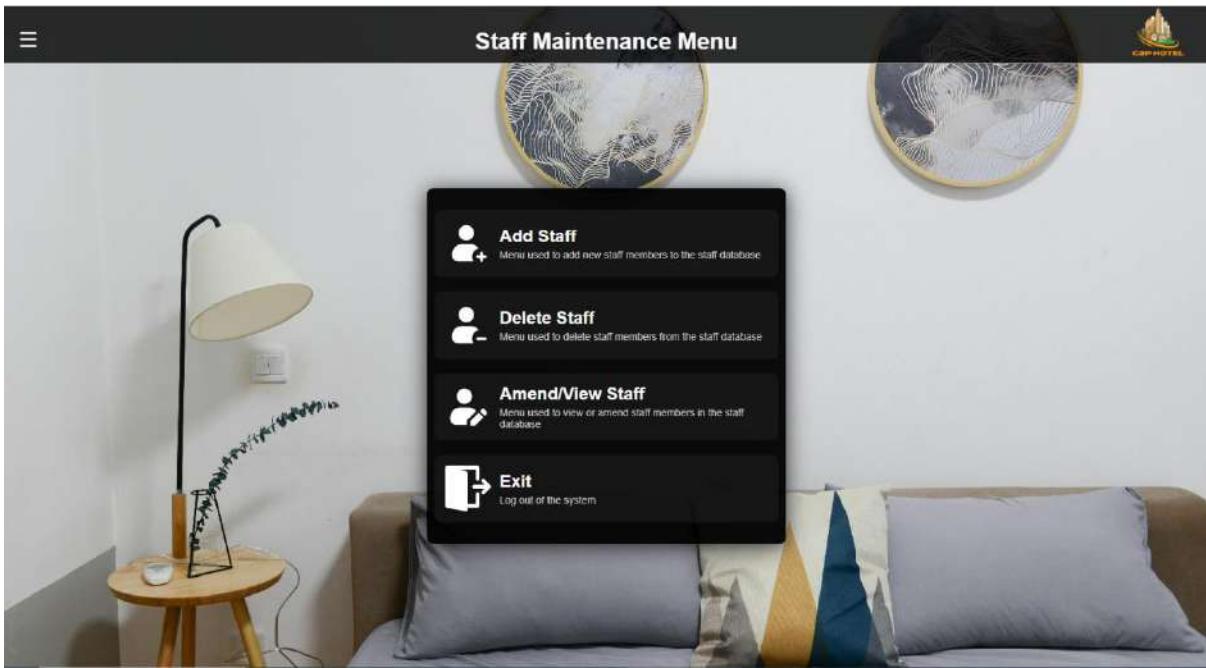


Figure 26. Staff Maintenance Menu Screen

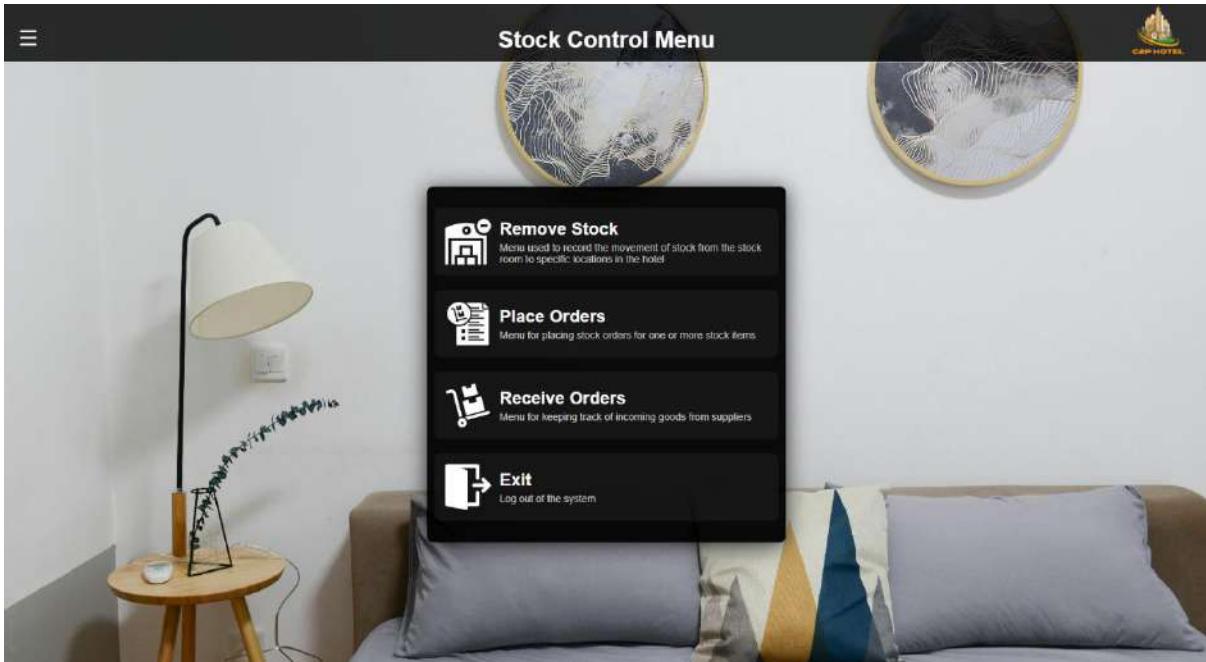


Figure 27. Stock Control Menu Screen

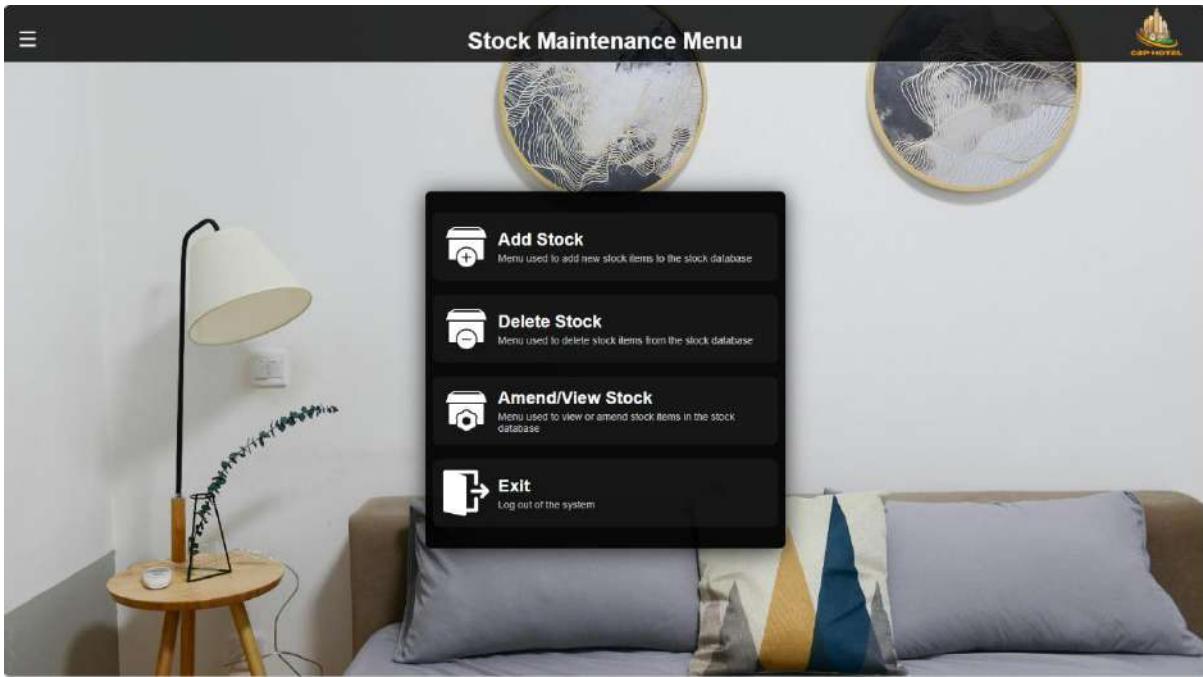


Figure 28. Stock Maintenance Menu Screen

# Individual Work (Code)

Mark Mukiiiza Written By: Mark Mukiiiza

## Add Stock Item Screen

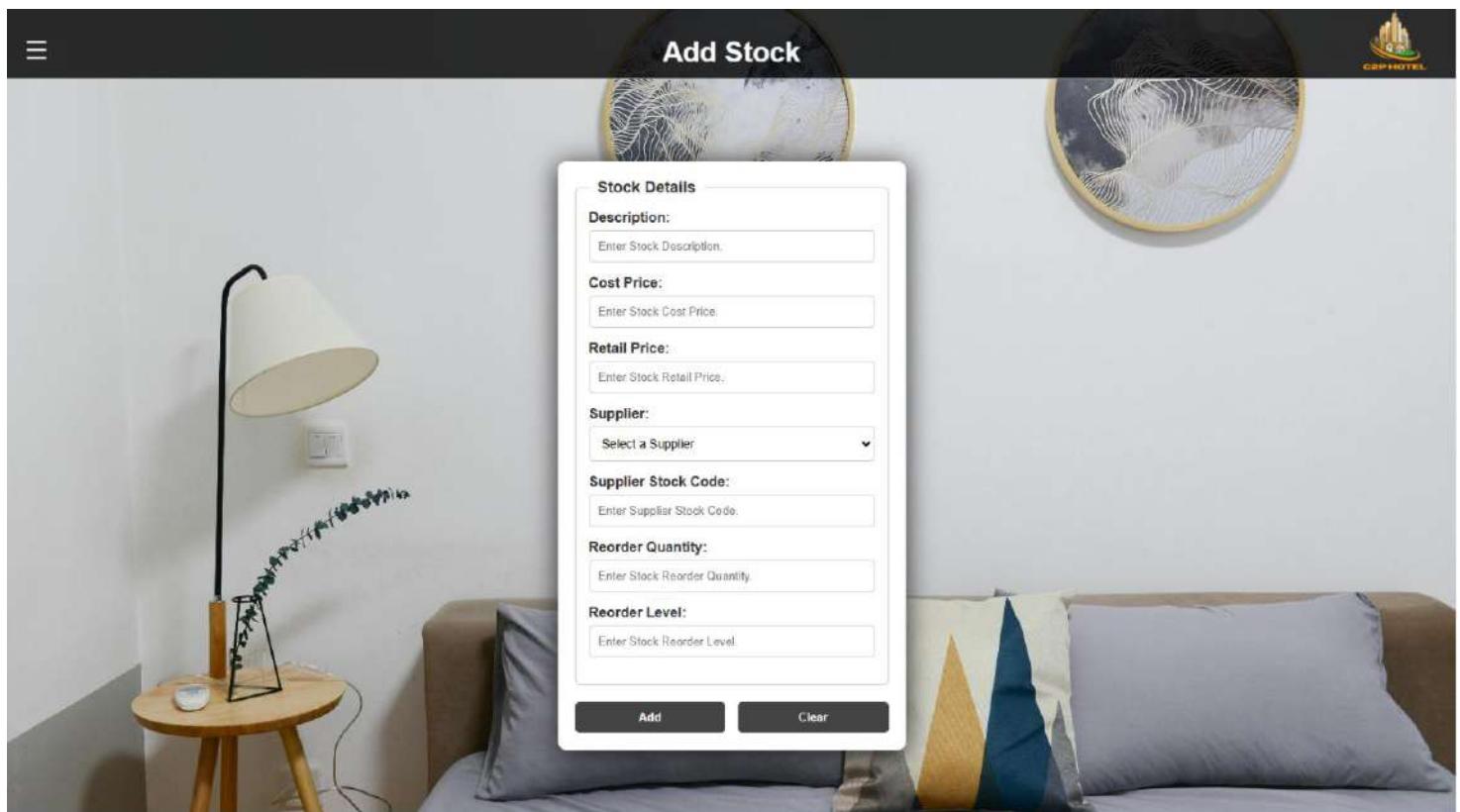


Figure 29. Add Stock Item Screen

```

1  <!--
2      Name of screen:      addstock.html.php
3      Purpose of screen: Generates a form to allow the user to add a stock item to the Hotel database.
4      Student ID:          C00166672
5      Name:                Mark Mukiza
6      Date Written:        February 2025
7  -->
8  <?php session_start(); ?>
9  <!DOCTYPE html>
10 <html lang="en">
11 <head>
12     <meta charset="UTF-8">
13     <title>Add Stock</title>
14     <!-- Link to the main stylesheet -->
15     <link rel="stylesheet" href="../../styles.css">
16     <!-- Include external JavaScript for form validation -->
17     <script src="formValidation.js"></script>
18 </head>
19 <body>
20     <!-- Burger icon that will toggle the sidebar -->
21     <div class="burger-menu">#9776;</div>
22     <!-- Page header with title and logo -->
23     <div class="page-header">
24         <h1>Add Stock</h1>
25         <a href="../../home.html.php"></a>
26     </div>
27     <!-- Sidebar navigation -->
28     <?php include '../../sidebar.php'; ?>
29     <!-- Main content area for the 'Add Stock' page -->
30     <div class="content add-page">
31         <!-- Include the flash message for success or failure to add from an external PHP file -->
32         <?php include 'php/flashMessage.php'; ?>
33         <!-- Stock addition form -->
34         <!-- The form uses both validation and a confirmation prompt on submission -->
35         <form action="php/addstock.php" method="POST" onsubmit="return validateForm() && confirmSubmit()">
36             <fieldset>
37                 <legend>Stock Details</legend>
38                 <!-- Input field for Stock Description -->
39                 <!-- https://www.regular-expressions.info/lookarounds.html
40                     https://stackoverflow.com/questions/3802192/regexp-java-for-password-validation
41                     Pattern Explanation:
42                         ^(?!\d)
43                             This negative lookahead ensures the string does not start with a digit.
44                             Example: "1Car battery" will be rejected because it starts with "1".
45                         (?=.{3,})
46                             Asserts that the entire string is at least 3 characters long.
47                         (?=.*[A-Za-z])
48                             Requires that there is at least one letter somewhere in the string.
49                         [A-Za-z0-9\s]+
50                             Matches only letters, digits, and whitespace for the entire string.-->
51             <div class="inputbox">
52                 <label for="description">Description:</label>
53                 <input type="text" name="description" id="description"
54                     pattern="^(?!\d)(?=.{3,})(?=.*[A-Za-z])[A-Za-z0-9\s]+"
55                     title="Just a short description of the stock item." placeholder="Enter Stock Description."
56                     onchange="checkDescription(this)" autofocus required />
57             </div>
58             <!-- Input field for Cost Price -->
59             <!-- Pattern enforces a number with up to 2 decimal places -->
60             <div class="inputbox">
61                 <label for="costprice">Cost Price:</label>
62                 <input type="number" name="costprice" id="costprice" placeholder="Enter Stock Cost Price."
63                     title="Enter a number with up to 2 decimal places. Max 5000. Cannot be a negative number."
64                     pattern="^\d+(\.\d{1,2})?$" min="0.10" max="5000" step="0.01" onchange="checkPrice(this)"
65                     required />
66             </div>

```

```

67      <!-- Input field for Retail Price -->
68      <!-- Pattern enforces a number with up to 2 decimal places -->
69      <div class="inputbox">
70          <label for="retailprice">Retail Price:</label>
71          <input type="number" name="retailprice" id="retailprice" placeholder="Enter Stock Retail Price."
72              title="Enter a number with up to 2 decimal places. Max 5000. Cannot be a negative number."
73              pattern="^\d{1,2}.\d{2}" min="0.10" max="5000" step="0.01" onchange="checkPrice(this)"
74              required />
75      </div>
76      <!-- Supplier dropdown -->
77      <div class="inputbox">
78          <label for="supplier">Supplier:</label>
79          <select name="supplier" id="supplier" required>
80              <option value="" disabled selected>Select a Supplier</option>
81              <!-- Include the supplier list options from an external PHP file -->
82              <?php include 'php/supplierListbox.php'; ?>
83          </select>
84      </div>
85      <!-- Input field for Supplier Stock Code -->
86      <!-- Pattern enforces 4 Uppercase Letters followed by 3 numbers -->
87      <div class="inputbox">
88          <label for="suppliercode">Supplier Stock Code:</label>
89          <input type="text" name="suppliercode" id="suppliercode" placeholder="Enter Supplier Stock Code."
90              title="Must be 4 uppercase letters followed by 3 digits. e.g. TOWL001."
91              pattern="^[A-Z]{4}[0-9]{3}$" onchange="checkSupplierCode(this)" required />
92      </div>
93      <!-- Input field for Reorder Quantity -->
94      <!-- Pattern enforces a number that's greater than one and no more than 200 -->
95      <div class="inputbox">
96          <label for="reorderqty">Reorder Quantity:</label>
97          <input type="number" name="reorderqty" id="reorderqty" placeholder="Enter Stock Reorder Quantity."
98              pattern="^1[0-9]\d*$" title="Cannot be less than 1 or greater 200." min="1" max="200"
99              onchange="checkNumber(this)" required />
100     </div>
101     <!-- Input field for Reorder Level -->
102     <!-- Pattern enforces a number that's greater than one and no more than 200 -->
103     <div class="inputbox">
104         <label for="reorderlvl">Reorder Level:</label>
105         <input type="number" name="reorderlvl" id="reorderlvl" placeholder="Enter Stock Reorder Level."
106             pattern="^1[0-9]\d*$" title="Cannot be less than 1 or greater 200." min="1" max="200"
107             onchange="checkNumber(this)" required />
108     </div>
109 </fieldset>
110 <!-- Form buttons -->
111 <div class="myButton">
112     <!-- Submit button; the form will only be sent if validation and confirmation both return true -->
113     <input type="submit" value="Add" name="submit" />
114     <!-- Reset button to clear the form -->
115     <input type="reset" value="Clear" name="reset" />
116 </div>
117 </form>
118 </div>
119 <script src="../../sidebar.js"></script>
120 </body>
121 </html>

```

```

1  <!--
2      Name of screen:      addstock.php
3      Purpose of screen:  Processes the add stock form submissions and creates a success/failure message
4                      for display on addstock.html.php
5      Student ID:        C00166672
6      Name:              Mark Mukiiza
7      Date Written:     February 2025
8  -->
9  <?php
10 // Start the session to enable flash messaging.
11 session_start();
12 // Include the database connection settings.
13 include 'db.inc.php';
14 // Set the default time zone to UTC.
15 date_default_timezone_set("UTC");
16 // Sanitize user input to prevent SQL injection.
17 // Reference: https://www.w3schools.com/php/func\_mysqli\_real\_escape\_string.asp
18 $description = mysqli_real_escape_string($con, $_POST['description']);
19 $costPrice = mysqli_real_escape_string($con, $_POST['costprice']);
20 $retailPrice = mysqli_real_escape_string($con, $_POST['retailprice']);
21 $reorderQty = mysqli_real_escape_string($con, $_POST['reorderqty']);
22 $reorderLvl = mysqli_real_escape_string($con, $_POST['reorderlvl']);
23 $supplierCode = mysqli_real_escape_string($con, $_POST['suppliercode']);
24 $supplier = mysqli_real_escape_string($con, $_POST['supplier']);
25 // Prepare the SQL query to insert the new stock record into the database.
26 $sql = "INSERT INTO Stock (Description, CostPrice, RetailPrice, ReorderQuantity, ReorderLevel, SupplierStockCode, SupplierID)
27           VALUES ('$description', '$costPrice', '$retailPrice', '$reorderQty', '$reorderLvl', '$supplierCode', '$supplier')";
28 // Execute the SQL query.
29 if (!mysqli_query($con, $sql)) {
30     // If the query fails, set a flash message indicating the error.
31     $_SESSION['msg'] = "An error occurred in the SQL Query: " . mysqli_error($con);
32     $_SESSION['msg_type'] = "error";
33     // Close the database connection.
34     mysqli_close($con);
35     // Redirect back to the add stock form page.
36     header("Location: ../addstock.html.php");
37     exit();
38 } else {
39     // If the query is successful, set a flash message indicating success.
40     $_SESSION['msg'] = "Stock Item $description has been added.";
41     $_SESSION['msg_type'] = "success";
42     // Close the database connection.
43     mysqli_close($con);
44     // Redirect back to the add stock form page.
45     header("Location: ../addstock.html.php");
46     exit();
47 }
48 ?>

```

```
1 <!--
2     Name of screen:    supplierListbox.php
3     Purpose of screen: Generates the listbox for suppliers that are in the Supplier table in the Hotel database.
4     Student ID:        C00166672
5     Name:              Mark Mukiiza
6     Date Written:      February 2025
7 -->
8 <?php
9 // Include the database connection file
10 include 'db.inc.php';
11 // Prepare the SQL query to select active suppliers (where DeletedFlag equals 0)
12 $sql = "SELECT SupplierID, Name FROM Suppliers WHERE DeletedFlag = 0";
13 // Execute the SQL query and store the result
14 $result = mysqli_query($con, $sql);
15 // Check if the query returned any rows
16 if ($result && mysqli_num_rows($result) > 0) {
17     // Loop through each supplier record
18     while ($row = mysqli_fetch_assoc($result)) {
19         // Output an HTML <option> element with the SupplierID as the value
20         // and the Supplier Name as the visible text, escaping any special characters
21         echo "<option value='" . htmlspecialchars($row['SupplierID']) . "'>" . htmlspecialchars($row['Name']) . "</option>";
22     }
23 } else {
24     // If no suppliers are found, output a disabled option indicating that no suppliers are available
25     echo "<option value='' disabled>No suppliers available.</option>";
26 }
27 ?>
```

```

1  /* Name of screen:      formValidation.js
2  Purpose of screen:    Provides validation for the addstock.html.php forms fields as well as the confirmation prompt
3  Student ID:          C00166672
4  Name:                Mark Mukiiza
5  Date Written:        February 2025
6  */
7  function checkPrice(input) {
8      const value = parseFloat(input.value); // Convert the input value to a floating-point number
9      if (isNaN(value)) {
10         alert("Please enter a valid number!"); // Inform the user that a valid number is required
11         input.style.borderColor = "red"; // Set the border colour to red to indicate an error
12         return false;
13     }
14     if (value < 0) {
15         alert("The price cannot be less than zero!"); // Inform the user that negative numbers are not allowed
16         input.style.borderColor = "red"; // Set the border colour to red to indicate an error
17         return false;
18     }
19     if (value > 5000) {
20         alert("The price cannot be more than 5000!"); // Inform the user that incredibly large numbers are not allowed
21         input.style.borderColor = "red"; // Set the border colour to red to indicate an error
22         return false;
23     }
24     // Optional: Enforce up to two decimal places
25     if (!(/^\d{1,2}\.\d{1,2}?\$/).test(input.value)) {
26         alert("Please enter a valid number with up to two decimal places!"); // Inform the user of the required format
27         input.style.borderColor = "red"; // Set the border colour to red to indicate an error
28         return false;
29     }
30     input.style.borderColor = "green"; // Set the border colour to green if the input is valid
31     return true;
32 }
33 function checkDescription(input) {
34     const descr = input.value.trim(); // Trim any extra whitespace from the input value
35     const pattern = /^(?!d)(?=.{3,})(?=.*[A-Za-z])[A-Za-z0-9\s]+$/; // Same pattern as explained in addstock.html.php
36     if (descr === "") {
37         alert('Please enter a Description for the Stock Item!'); // Prompt the user to enter a description if the field is empty
38         input.style.borderColor = "red"; // Set the border colour to red to indicate an error
39         return false;
40     } else if (!pattern.test(descr)) {
41         alert("Description can only contain letters, numbers, and spaces!"); // Inform the user of the allowed characters
42         input.style.borderColor = "red"; // Set the border colour to red to indicate an error
43         return false;
44     } else {
45         input.style.borderColor = "green"; // Set the border colour to green if the input is valid
46         return true;
47     }
48 }
49 function checkSupplierCode(input) {
50     const supCode = input.value.trim(); // Get the trimmed value of the input to remove any extra spaces
51     const pattern = /^[A-Z]{4}[0-9]{3}\$/; // Regular expression: matches exactly 4 uppercase letters followed by 3 digits
52     if (supCode === "") { // Check if the supplier code field is empty
53         alert('Please enter a Supplier Stock Code!'); // Prompt the user to enter a supplier stock code
54         input.style.borderColor = "red"; // Set the border colour to red to indicate an error
55         return false;
56     } else if (!pattern.test(supCode)) { // Validate the supplier code against the required pattern
57         alert("Supplier Stock Code must be 4 uppercase letters followed by 3 digits (e.g., TOWL001)!"); // Inform the user of the correct format
58         input.style.borderColor = "red"; // Set the border colour to red to indicate an error
59         return false;
60     } else {
61         input.style.borderColor = "green"; // Set the border colour to green if the input is valid
62         return true;
63     }
64 }

```

```

65 function checkNumber(input) {
66     const value = parseInt(input.value, 10); // Convert the input value to an integer (base 10)
67     if (isNaN(value) || !Number.isInteger(value)) {
68         alert("Please enter a valid whole number!"); // Inform the user that a valid whole number is required
69         input.style.borderColor = "red"; // Set the border colour to red to indicate an error
70         return false;
71     }
72     if (value < 1) {
73         alert("The quantity cannot be less than one!"); // Inform the user that the quantity must be at least one
74         input.style.borderColor = "red"; // Set the border colour to red to indicate an error
75         return false;
76     }
77     if (value > 200) {
78         alert("The quantity cannot be more than 200!"); // Inform the user that the quantity cannot be more than 200
79         input.style.borderColor = "red"; // Set the border colour to red to indicate an error
80         return false;
81     }
82     input.style.borderColor = "green"; // Set the border colour to green if the input is valid
83     return true;
84 }
85 function validateForm() {
86     let isValid = true;
87     // Validate the Description field
88     const descriptionInput = document.getElementById('description');
89     if (!checkDescription(descriptionInput)) {
90         isValid = false;
91     }
92     // Validate the Cost Price field
93     const costPriceInput = document.getElementById('costprice');
94     if (!checkPrice(costPriceInput)) {
95         isValid = false;
96     }
97     // Validate the Retail Price field
98     const retailPriceInput = document.getElementById('retailprice');
99     if (!checkPrice(retailPriceInput)) {
100        isValid = false;
101    }
102    // Validate the Reorder Quantity field
103    const reorderQtyInput = document.getElementById('reorderqty');
104    if (!checkNumber(reorderQtyInput)) {
105        isValid = false;
106    }
107    // Validate the Reorder Level field
108    const reorderLvlInput = document.getElementById('reorderlvl');
109    if (!checkNumber(reorderLvlInput)) {
110        isValid = false;
111    }
112    if (!isValid) {
113        alert("Please correct the errors in the form before submitting."); // Prompt the user to fix errors before submission
114        return false;
115    }
116    return true;
117 }
118 function confirmSubmit() {
119     // Display a confirmation prompt when the form is submitted.
120     return confirm("Are you sure you want to add this stock item?");
121 }

```

## Delete Stock Item Screen

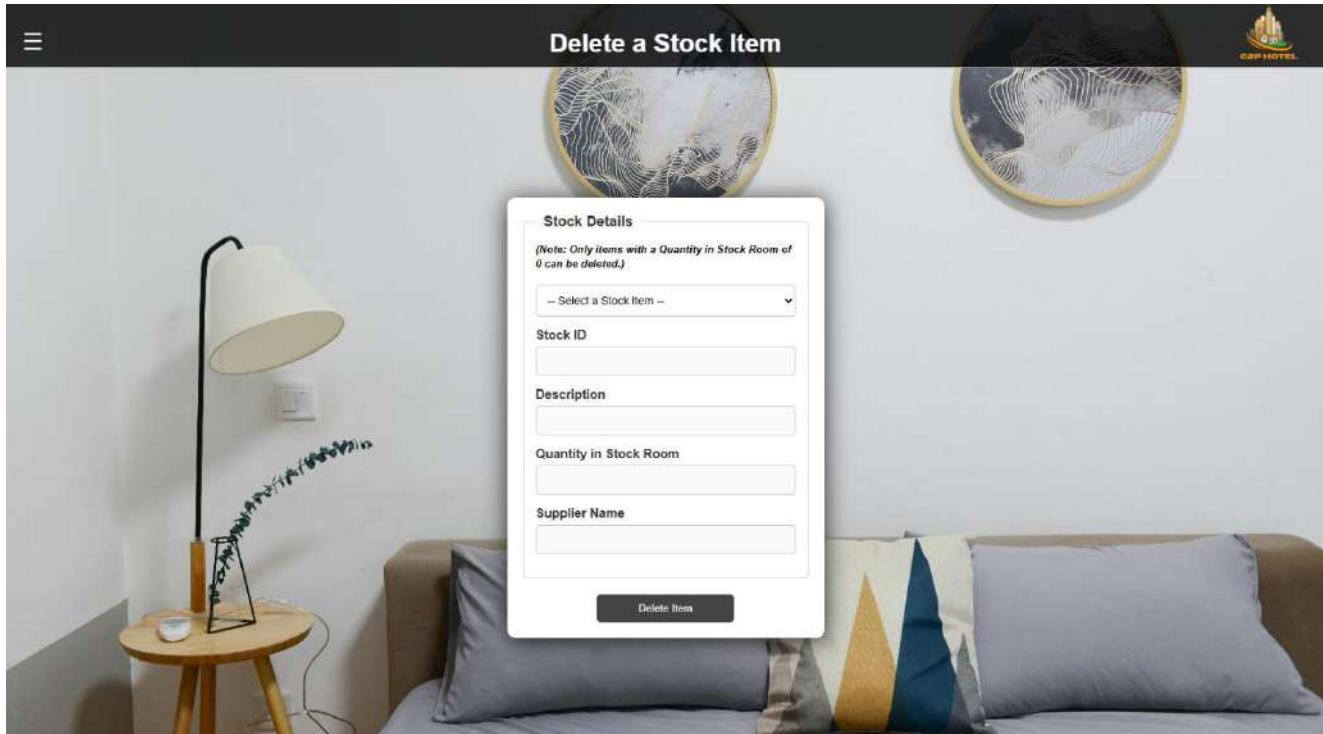


Figure 30. Delete Stock Item Screen

```

1 <!--
2     Name of screen:      deleteStock.html.php
3     Purpose of screen: Allows user to select stock from a listbox to delete from database if the quantity in store room is
4                         (Sets DeletedFlag to 1 which is deleted.)
5     Student ID:          C00166672
6     Name:                 Mark Mukiza
7     Date Written:        February 2025
8 -->
9 <?php session_start(); ?>
10 <!DOCTYPE html>
11 <html lang="en">
12 <head>
13     <meta charset="UTF-8"> <!-- Specify character encoding as UTF-8 -->
14     <title>Delete Stock</title> <!-- Title of the page -->
15     <!-- Link to the main stylesheet -->
16     <link rel="stylesheet" href="../../styles.css">
17 </head>
18 <body>
19     <!-- Burger icon that will toggle the sidebar -->
20     <div class="burger-menu">&#9776;</div>
21     <!-- Container for the page header (title and logo) -->
22     <div class="page-header">
23         <h1>Delete a Stock Item</h1> <!-- Main heading of the page -->
24         <a href="../../home.html.php"></a>
25     </div>
26     <!-- Sidebar navigation -->
27     <?php include '../../sidebar.php'; ?>
28     <!-- Main content area for the 'Delete Stock' page -->
29     <div class="content delete-page">
30         <script src="confirmDelete.js"></script>
31         <!-- Include the flash message for success or failure to add from an external PHP file -->
32         <?php include 'php/flashMessage.php'; ?>
33         <!-- Form for deleting a stock item -->
34         <form action="php/deleteStock.php" onsubmit="return confirmDelete()" method="post">
35             <fieldset> <!-- Group related form elements -->
36                 <legend>Stock Details</legend> <!-- Caption for the fieldset -->
37                 <p><small>(Note: Only items with a Quantity in Stock Room of 0 can be deleted.)</small></p>
38                 <!-- Container for the listbox populated via PHP -->
39                 <div class="inputbox">
40                     <?php include 'php/deleteStockListbox.php'; ?> <!-- Include PHP file to generate the stock listbox -->
41                 </div>
42                 <!-- Input field for Stock ID -->
43                 <div class="inputbox">
44                     <label for="stockid">Stock ID</label> <!-- Label for the Stock ID field -->
45                     <input type="text" name="stockid" id="stockid" disabled> <!-- Disabled input for Stock ID -->
46                 </div>
47                 <!-- Input field for Description -->
48                 <div class="inputbox">
49                     <label for="description">Description</label> <!-- Label for the Description field -->
50                     <input type="text" name="description" id="description" disabled>
51                     <!-- Disabled input for Description -->
52                 </div>
53                 <!-- Input field for Quantity in Stock -->
54                 <div class="inputbox">
55                     <label for="quantity">Quantity in Stock Room</label> <!-- Label for the Quantity field -->
56                     <input type="text" name="quantity" id="quantity" disabled> <!-- Disabled input for Quantity -->
57                 </div>
58                 <!-- Input field for Supplier Name -->
59                 <div class="inputbox">
60                     <label for="supplier">Supplier Name</label> <!-- Label for the Supplier Name field -->
61                     <input type="text" name="supplier" id="supplier" disabled> <!-- Disabled input for Supplier Name -->
62                 </div>
63             </fieldset>
64             <!-- Container for the delete button (single-button form) -->
65             <div class="deleteButton">
66                 <input type="submit" value="Delete Item"> <!-- Submit button to delete the stock item -->
67             </div>
68         </form>
69     </div>
70     <script src="../../sidebar.js"></script>
71 </body>
72 </html>

```

```

1 <!--
2   Name of screen:      deleteStockListbox.php
3   Purpose of screen: Generates the listbox for available stock from the Stock Table in the database.
4   Student ID:          C00166672
5   Name:                 Mark Mukiza
6   Date Written:        February 2025
7 -->
8 <?php
9 include "db.inc.php";
10 /* Retrieve stock items that have not been deleted,
11    and join with the Suppliers table to get the supplier's Name.*/
12 $sql = "SELECT stock.StockID, stock.Description, stock.QuantityInStockRoom, sup.Name AS SupplierName
13           FROM Stock stock
14             JOIN Suppliers sup ON stock.SupplierID = sup.SupplierID
15             WHERE stock.DeletedFlag = 0 AND sup.DeletedFlag = 0";
16 $result = mysqli_query($con, $sql);
17 if (!$result) {
18   die('Error in querying the database: ' . mysqli_error($con));
19 }
20 echo "<br><select name='listbox' id='listbox' onclick='populate()'>";
21 echo "<option value=''>-- Select a Stock Item --</option>";
22 while ($row = mysqli_fetch_array($result)) {
23   $stockid = $row['StockID'];
24   $description = $row['Description'];
25   $quantity = $row['QuantityInStockRoom'];
26   $supplier = $row['SupplierName'];
27   // Build a comma-separated string in the order expected by the populate() function.
28   $allText = "$stockid, $description, $quantity, $supplier";
29   echo "<option value='$allText'>$description</option>";
30 }
31 echo "</select>";
32 mysqli_close($con);
33 ?>

```

```

1  <!--
2      Name of screen:      deleteStock.php
3      Purpose of screen:  Has the php required to delete stock from database. Is called by deleteStock.php.html
4      Student ID:        C00166672
5      Name:              Mark Mukiza
6      Date Written:      February 2025
7  -->
8  <?php
9  // Start the session to allow using session variables.
10 session_start();
11 // Include the database connection file.
12 include 'db.inc.php';
13 // Set the default timezone to UTC.
14 date_default_timezone_set('UTC');
15 /* Reference: https://www.phptutorial.net/php-tutorial/php-flash-messages/
16     Check if a stock item has been selected from the form.
17     If 'stockid' is not set or is empty, set an error message and redirect.*/
18 if (!isset($_POST['stockid']) || empty($_POST['stockid'])) {
19     $_SESSION['delete_msg'] = "No stock item was selected.";
20     $_SESSION['delete_type'] = "error";
21     header("Location: ../deleteStock.html.php");
22     exit();
23 }
24 /* Sanitize the received stock ID to prevent SQL injection.
25     Reference: https://www.w3schools.com/php/func_mysqli_real_escape_string.asp*/
26 $stockid = mysqli_real_escape_string($con, $_POST['stockid']);
27 /* Prepare a query to check if the stock item exists and is not already deleted.
28     Also retrieve the 'QuantityInStockRoom' and 'Description' (stock name) for further checks and display.*/
29 $sql_check = "SELECT QuantityInStockRoom, Description
30                 FROM Stock
31                 WHERE StockID = '$stockid' AND DeletedFlag = 0";
32 // Execute the query.
33 $result = mysqli_query($con, $sql_check);
34 // If the query fails, set an error flash message and redirect.
35 if (!$result) {
36     $_SESSION['delete_msg'] = "Error: " . mysqli_error($con);
37     $_SESSION['delete_type'] = "error";
38     header("Location: ../deleteStock.html.php");
39     exit();
40 }
41 // If no matching stock item is found, set an error message and redirect.
42 if (mysqli_num_rows($result) == 0) {
43     $_SESSION['delete_msg'] = "Stock item not found or already deleted.";
44     $_SESSION['delete_type'] = "error";
45     header("Location: ../deleteStock.html.php");
46     exit();
47 }
48 // Fetch the stock item details as an associative array.
49 $row = mysqli_fetch_assoc($result);
50 // If the Quantity in Stock Room is greater than zero, deletion is not permitted.
51 if ($row['QuantityInStockRoom'] > 0) {
52     $_SESSION['delete_msg'] = "Cannot delete this stock item as Quantity in Stock Room is greater than zero.";
53     $_SESSION['delete_type'] = "error";
54     header("Location: ../deleteStock.html.php");
55     exit();
56 }
57

```

```
58 // Prepare the SQL query to mark the stock item as deleted.
59 // This sets the 'DeletedFlag' to 1 and updates the 'DateLastUpdate' to the current date.
60 $sql = "UPDATE Stock
61     SET DeletedFlag = 1, DateLastUpdate = NOW()
62     WHERE StockID = '$stockid'";
63 // Execute the deletion query.
64 if (!mysqli_query($con, $sql)) {
65     // If the query fails, set an error flash message.
66     $_SESSION['delete_msg'] = "Error: " . mysqli_error($con);
67     $_SESSION['delete_type'] = "error";
68 } else {
69     // Check if any records were actually updated.
70     if (mysqli_affected_rows($con) > 0) {
71         // If deletion was successful, set a success message that includes both the stock ID and description.
72         $_SESSION['delete_msg'] = "Stock Item ID $stockid (" . htmlspecialchars($row['Description']) . ") has been marked as deleted.";
73         $_SESSION['delete_type'] = "success";
74     } else {
75         // If no records were updated, set an error message.
76         $_SESSION['delete_msg'] = "No records were changed.";
77         $_SESSION['delete_type'] = "error";
78     }
79 }
80 // Close the database connection.
81 mysqli_close($con);
82 // Redirect back to the deleteStock.html.php page so that the flash message can be displayed.
83 header("Location: ../deleteStock.html.php");
84 exit();
85 ?>
```

```

1  /*
2   Name of screen:    confirmDelete.js
3   Purpose of screen: Populates Delete form and confirms deletion.
4   Student ID:       C00166672
5   Name:              Mark Mukiza
6   Date Written:     February 2025
7
8 When an item is selected in the listbox, populate the form fields.
9 The expected order in the option's value is:
10 StockID, Description, Quantity in Stock Room, Supplier Name.*/
11 function populate() {
12     var sel = document.getElementById("listbox"); // Get the listbox element by its ID
13     var result = sel.options[sel.selectedIndex].value; // Retrieve the value of the selected option
14     if (result === "") {
15         document.getElementById("stockid").value = ""; // Clear the Stock ID field
16         document.getElementById("description").value = ""; // Clear the Description field
17         document.getElementById("quantity").value = ""; // Clear the Quantity field
18         document.getElementById("supplier").value = ""; // Clear the Supplier Name field
19         return;
20     }
21     var stockDetails = result.split(', '); // Split the selected value into an array of details
22     document.getElementById("stockid").value = stockDetails[0]; // Populate the Stock ID field
23     document.getElementById("description").value = stockDetails[1]; // Populate the Description field
24     document.getElementById("quantity").value = stockDetails[2]; // Populate the Quantity field
25     document.getElementById("supplier").value = stockDetails[3]; // Populate the Supplier Name field
26 }
27 /* Before deletion, verify that the Quantity in Stock Room is 0.
28 If it is, prompt the user for confirmation.*/
29 function confirmDelete() {
30     var qty = parseInt(document.getElementById("quantity").value, 10); // Convert the Quantity value to an integer
31     if (qty > 0) {
32         alert("Cannot delete this stock item because the Quantity in Stock Room is greater than zero."); // Alert the user if quantity is above zero
33         return false; // Prevent form submission
34     }
35     var response = confirm('Are you sure you want to delete this item?'); // Ask the user for confirmation
36     if (response) {
37         // Re-enable the Stock ID field so its value is submitted.
38         document.getElementById("stockid").disabled = false; // Enable the Stock ID field for submission
39         return true; // Allow form submission
40     } else {
41         // If cancelled, re-populate to re-display the current values.
42         populate(); // Re-populate the form fields with the selected details
43         return false; // Prevent form submission
44     }
45 }

```

## Amend/View Stock Item Screen

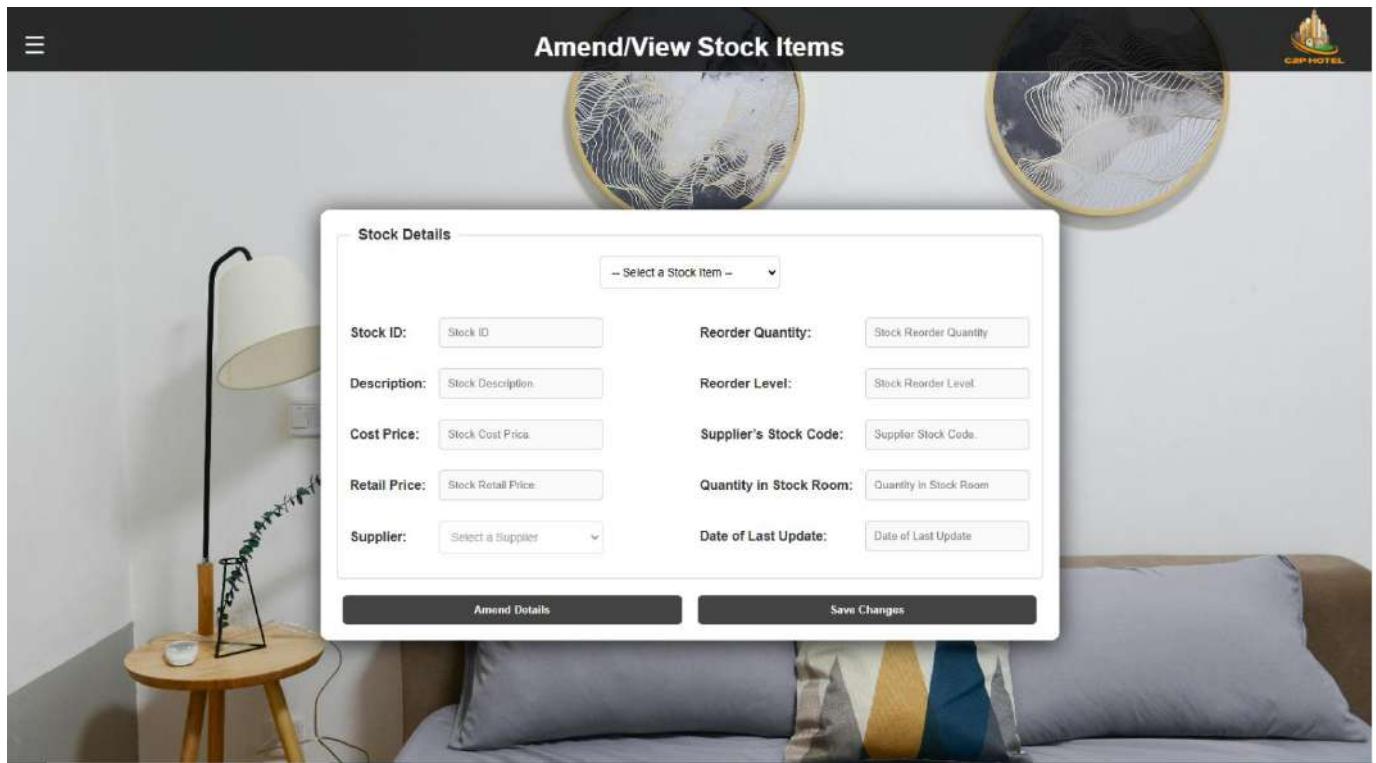


Figure 31. Amend/View Stock Item Screen

```

1  <!--
2   Name of screen: amendViewStock.html.php
3   Purpose of screen: Generates a form to allow the user to amend a stock item in the Hotel database.
4   Student ID: C00166672
5   Name: Mark Mukiiza
6   Date Written: February 2025
7  -->
8  <?php session_start(); ?>
9  <!DOCTYPE html>
10 <html lang="en">
11 <head>
12   <meta charset="UTF-8"> <!-- Specify character encoding as UTF-8 -->
13   <title>Amend/View Stock</title> <!-- Title of the page -->
14   <!-- Link to the main stylesheet -->
15   <link rel="stylesheet" href="../../styles.css">
16   <!-- Include external JavaScript for form validation -->
17   <script src="formValidation.js"></script>
18 </head>
19 <body>
20   <!-- Burger icon that will toggle the sidebar -->
21   <div class="burger-menu">#9776;</div>
22   <!-- Page header with title and logo -->
23   <div class="page-header">
24     <h1>Amend/View Stock Items</h1> <!-- Main heading of the page -->
25     <a href="../../home.html.php"></a>
26   </div>
27   <!-- Sidebar navigation -->
28   <?php include '../../sidebar.php'; ?>
29   <!-- Main content area for the 'Delete Stock' page -->
30   <div class="content amend-view-page">
31     <script src="amendStock.js"></script>
32     <!-- Include the flash message for success or failure to add from an external PHP file -->
33     <?php include 'php/flashMessage.php'; ?>
34     <form action="php/amendViewStock.php" onsubmit="confirmAmend()" method="post">
35       <fieldset>
36         <legend>Stock Details</legend>
37         <!-- Container for the stock listbox populated via PHP -->
38         <div class="centered-select">
39           <div class="inputbox">
40             <?php include 'php/amendViewStockList.php'; ?>
41             <!-- Include PHP file to generate the stock listbox -->
42           </div>
43         </div>
44         <div class="columns">
45           <div class="left-column">
46             <!-- Container for the Stock ID input field -->
47             <div class="inputbox">
48               <label for="amendStockId">Stock ID:</label>
49               <input type="text" name="amendStockId" id="amendStockId" placeholder="Stock ID" required
50                 disabled>
51             </div>
52             <!-- Input field for Stock Description -->
53             <!-- https://www.regular-expressions.info/lookarounds.html
54             https://stackoverflow.com/questions/3802192/regexp-java-for-password-validation
55             Pattern Explanation:
56               ^(?!\d)
57               This negative lookahead ensures the string does not start with a digit.
58               Example: "1Car battery" will be rejected because it starts with "1".
59               (?=.{3,})
56               Asserts that the entire string is at least 3 characters long.
57               (?=.*[A-Za-z])
58               Requires that there is at least one letter somewhere in the string.
59               [A-Za-z0-9\s]+$
60               Matches only letters, digits, and whitespace for the entire string.-->
61             <div class="inputbox">
62               <label for="amendDescription">Description:</label>
63               <input type="text" name="amendDescription" id="amendDescription"
64                 pattern="^(?!\d)(?=.{3,})(?=.*[A-Za-z])[A-Za-z0-9\s]+$"
65                 title="Just a short description of the stock item." placeholder="Stock Description."
66                 onchange="checkDescription(this)" required disabled>
67             </div>

```

```

72      <!-- Container for the Cost Price input field -->
73      <div class="inputbox">
74          <label for="amendCostPrice">Cost Price:</label>
75          <input type="text" name="amendCostPrice" id="amendCostPrice" placeholder="Stock Cost Price."
76              title="Enter a number with up to 2 decimal places. Max 5000. Cannot be a negative number."
77              pattern="^\d{1,2}(\.\d{1,2})?$" min="0.10" max="5000" step="0.01"
78              onchange="checkPrice(this)" required disabled>
79      </div>
80      <!-- Container for the Retail Price input field -->
81      <div class="inputbox">
82          <label for="amendRetailPrice">Retail Price:</label>
83          <input type="text" name="amendRetailPrice" id="amendRetailPrice"
84              placeholder="Stock Retail Price."
85              title="Enter a number with up to 2 decimal places. Max 5000. Cannot be a negative number."
86              pattern="^\d{1,2}(\.\d{1,2})?$" min="0.10" max="5000" step="0.01"
87              onchange="checkPrice(this)" required disabled>
88      </div>
89      <!-- Container for the Supplier Name listbox-->
90      <div class="inputbox">
91          <label for="amendSupplier">Supplier:</label>
92          <select name="amendSupplier" id="amendSupplier" required disabled>
93              <option value="" disabled selected>Select a Supplier</option>
94              <!-- Include the supplier list options from an external PHP file -->
95              <?php include 'php/supplierListbox.php'; ?>
96          </select>
97      </div>
98  </div>
99  <div class="right-column">
100     <!-- Container for the Reorder Quantity input field -->
101     <div class="inputbox">
102         <label for="amendReorderQuantity">Reorder Quantity:</label>
103         <input type="text" name="amendReorderQuantity" id="amendReorderQuantity"
104             onchange="checkNumber(this)" placeholder="Stock Reorder Quantity" required disabled>
105     </div>
106     <!-- Container for the Reorder Level input field -->
107     <div class="inputbox">
108         <label for="amendReorderLevel">Reorder Level:</label>
109         <input type="text" name="amendReorderLevel" id="amendReorderLevel"
110             placeholder="Stock Reorder Level." pattern="^1-9\d*$"
111             title="Cannot be less than 1 or greater 200." min="1" max="200"
112             onchange="checkNumber(this)" required disabled>
113     </div>
114     <!-- Container for the Supplier's Stock Code input field -->
115     <div class="inputbox">
116         <label for="amendSupplierStockCode">Supplier's Stock Code:</label>
117         <input type="text" name="amendSupplierStockCode" id="amendSupplierStockCode"
118             placeholder="Supplier Stock Code."
119             title="Must be 4 uppercase letters followed by 3 digits. e.g. TOWL001."
120             pattern="^([A-Z]{4})[0-9]{3}$" onchange="checkSupplierCode(this)" required disabled>
121     </div>
122     <!-- Container for the Quantity in Stock Room input field -->
123     <div class="inputbox">
124         <label for="amendQuantityInStockRoom">Quantity in Stock Room:</label>
125         <input type="text" name="amendQuantityInStockRoom" id="amendQuantityInStockRoom"
126             placeholder="Quantity in Stock Room" disabled>
127     </div>
128     <!-- Container for the Date of Last Update input field -->
129     <div class="inputbox">
130         <label for="amendDateLastUpdate">Date of Last Update:</label>
131         <input type="text" name="amendDateLastUpdate" id="amendDateLastUpdate"
132             placeholder="Date of Last Update" disabled>
133     </div>
134     </div>
135 </fieldset>
136     <!-- Container for the Amend Details and Save Changes buttons (Save changes is default submit with "Save Changes" value) -->
137     <div class="myButton">
138         <input type="button" value="Amend Details" id="amendViewbutton" onclick="toggleLock()">
139         <input type="submit" value="Save Changes" onclick="return validateForm()">
140     </div>
141     </form>
142 </div>
143     <script src="../../sidebar.js"></script>
144 </body>
145 </html>

```

```

1 <!--
2     Name of screen: amendViewStock.php
3     Purpose of screen: Process the amend/view form submissions and creates a success/failure message.
4     Student ID: C00166672
5     Name: Mark Mukiza
6     Date Written: February 2025
7 -->
8 <?php
9 // Start the session to enable flash messaging.
10 session_start();
11 // Include the database connection settings.
12 include 'db.inc.php';
13 // Set the default time zone to UTC.
14 date_default_timezone_set('UTC');
15 // Sanitize the form input values to prevent SQL injection.
16 // These values correspond to the input names in the HTML form.
17 $stockId = mysqli_real_escape_string($con, $_POST['amendStockId']);
18 $description = mysqli_real_escape_string($con, $_POST['amendDescription']);
19 $costPrice = mysqli_real_escape_string($con, $_POST['amendCostPrice']);
20 $retailPrice = mysqli_real_escape_string($con, $_POST['amendRetailPrice']);
21 $supplier = mysqli_real_escape_string($con, $_POST['amendSupplier']);
22 $reorderQty = mysqli_real_escape_string($con, $_POST['amendReorderQuantity']);
23 $reorderLvl = mysqli_real_escape_string($con, $_POST['amendReorderLevel']);
24 $supplierStockCode = mysqli_real_escape_string($con, $_POST['amendSupplierStockCode']);
25 $quantityInStock = mysqli_real_escape_string($con, $_POST['amendQuantityInStockRoom']);
26 // Prepare the SQL query to update the Stock record with the amended details.
27 // Adjust the column names if they differ in your actual database schema.
28 $sql = "UPDATE Stock SET
29             Description = '$description',
30             CostPrice = '$costPrice',
31             RetailPrice = '$retailPrice',
32             SupplierID = '$supplier',
33             ReorderQuantity = '$reorderQty',
34             ReorderLevel = '$reorderLvl',
35             SupplierStockCode = '$supplierStockCode',
36             QuantityInStockRoom = '$quantityInStock',
37             DateLastUpdate = NOW()
38         WHERE StockID = '$stockId'";
39
40 // Execute the query and check for errors.
41 if (!mysqli_query($con, $sql)) {
42     // If the query fails, set a flash message with the error details.
43     $_SESSION['msg'] = "Error: " . mysqli_error($con);
44     $_SESSION['msg_type'] = "error";
45 } else {
46     // If the query succeeds, check if any records were updated.
47     if (mysqli_affected_rows($con) > 0) {
48         // Set a success flash message including the stock ID and description.
49         $_SESSION['msg'] = "Stock Item ID $stockId (" . htmlspecialchars($description) . ") has been updated.";
50         $_SESSION['msg_type'] = "success";
51     } else {
52         // If no records were changed, set an error flash message.
53         $_SESSION['msg'] = "No records were changed.";
54         $_SESSION['msg_type'] = "error";
55     }
56 }
57 // Close the database connection.
58 mysqli_close($con);
59 // Redirect back to the Amend/View Stock page where the flash message will be displayed.
60 header("Location: ../amendViewStock.html.php");
61 exit();
62 ?>

```

```

1  <!--
2      Name of screen:      amendViewStockList.php
3      Purpose of screen: Generates the stock listbox.
4      Student ID:          C00166672
5      Name:                Mark Mukiiza
6      Date Written:        February 2025
7  -->
8  <?php
9  include "db.inc.php";
10 /* Retrieve stock items that have not been deleted,
11     and join with the Suppliers table to get the supplier's Name.*/
12 $sql = "SELECT stock.StockID, stock.Description, stock.CostPrice, stock.RetailPrice,
13             sup.Name AS SupplierName, stock.ReorderQuantity, stock.ReorderLevel,
14             stock.SupplierStockCode, stock.QuantityInStockRoom, stock.DateLastUpdate
15         FROM Stock stock
16         JOIN Suppliers sup ON stock.SupplierID = sup.SupplierID
17         WHERE stock.DeletedFlag = 0 AND sup.DeletedFlag = 0";
18 $result = mysqli_query($con, $sql);
19 if (!$result) {
20     die('Error in querying the database: ' . mysqli_error($con));
21 }
22 echo "<br><select name='listbox' id='listbox' onclick='populate()'>";
23 echo "<option value=''>-- Select a Stock Item --</option>";
24 while ($row = mysqli_fetch_array($result)) {
25     $stockid = $row['StockID'];
26     $description = $row['Description'];
27     $costprice = $row['CostPrice'];
28     $retailprice = $row['RetailPrice'];
29     $supplier = $row['SupplierName'];
30     $reorderqty = $row['ReorderQuantity'];
31     $reorderlvl = $row['ReorderLevel'];
32     $supplierstockcode = $row['SupplierStockCode'];
33     $quantity = $row['QuantityInStockRoom'];
34     $datelastupdate = $row['DateLastUpdate'];
35     // Build a comma-separated string in the order expected by the populate() function.
36     $allText = "$stockid, $description, $costprice, $retailprice, $supplier, $reorderqty,
37     $reorderlvl, $supplierstockcode, $quantity, $datelastupdate";
38     echo "<option value='$allText'$description</option>";
39 }
40 echo "</select>";
41 mysqli_close($con);
42 ?>

```

```

1  /* Name of screen:      formValidation.js
2  Purpose of screen: Provides validation for the addstock.html.php forms fields as well as the confirmation prompt
3  Student ID:          C00166672
4  Name:                 Mark Mukiza
5  Date Written:        February 2025
6 */
7 function checkPrice(input) {
8     const value = parseFloat(input.value); // Convert the input value to a floating-point number
9     if (isNaN(value)) {
10         alert("Please enter a valid number!"); // Inform the user that a valid number is required
11         input.style.borderColor = "red"; // Set the border colour to red to indicate an error
12         return false;
13     }
14     if (value < 0) {
15         alert("The price cannot be less than zero!"); // Inform the user that negative numbers are not allowed
16         input.style.borderColor = "red"; // Set the border colour to red to indicate an error
17         return false;
18     }
19     if (value > 5000) {
20         alert("The price cannot be more than 5000!"); // Inform the user that incredibly large numbers are not allowed
21         input.style.borderColor = "red"; // Set the border colour to red to indicate an error
22         return false;
23     }
24     // Optional: Enforce up to two decimal places
25     if (!(/^\d{1,2}\.\d{1,2}?\$/).test(input.value)) {
26         alert("Please enter a valid number with up to two decimal places!"); // Inform the user of the required format
27         input.style.borderColor = "red"; // Set the border colour to red to indicate an error
28         return false;
29     }
30     input.style.borderColor = "green"; // Set the border colour to green if the input is valid
31     return true;
32 }
33 function checkDescription(input) {
34     const descr = input.value.trim(); // Trim any extra whitespace from the input value
35     const pattern = /^(\?!d)(?=.{3})(?=.*[A-Za-z])[A-Za-z0-9\s]+$/; // Same pattern as explained in addstock.html.php
36     if (descr === "") {
37         alert('Please enter a Description for the Stock Item!'); // Prompt the user to enter a description if the field is empty
38         input.style.borderColor = "red"; // Set the border colour to red to indicate an error
39         return false;
40     } else if (!pattern.test(descr)) {
41         alert("Description can only contain letters, numbers, and spaces!"); // Inform the user of the allowed characters
42         input.style.borderColor = "red"; // Set the border colour to red to indicate an error
43         return false;
44     } else {
45         input.style.borderColor = "green"; // Set the border colour to green if the input is valid
46         return true;
47     }
48 }
49 function checkSupplierCode(input) {
50     const supCode = input.value.trim(); // Get the trimmed value of the input to remove any extra spaces
51     const pattern = /^[A-Z]{4}[0-9]{3}\$/; // Regular expression: matches exactly 4 uppercase letters followed by 3 digits
52     if (supCode === "") { // Check if the supplier code field is empty
53         alert('Please enter a Supplier Stock Code!'); // Prompt the user to enter a supplier stock code
54         input.style.borderColor = "red"; // Set the border colour to red to indicate an error
55         return false;
56     } else if (!pattern.test(supCode)) { // Validate the supplier code against the required pattern
57         alert("Supplier Stock Code must be 4 uppercase letters followed by 3 digits (e.g., TOWL001)!"); // Inform the user of the correct format
58         input.style.borderColor = "red"; // Set the border colour to red to indicate an error
59         return false;
60     } else {
61         input.style.borderColor = "green"; // Set the border colour to green if the input is valid
62         return true;
63     }
64 }

```

```

65 function checkNumber(input) {
66     const value = parseInt(input.value, 10); // Convert the input value to an integer (base 10)
67     if (isNaN(value) || !Number.isInteger(value)) {
68         alert("Please enter a valid whole number!"); // Inform the user that a valid whole number is required
69         input.style.borderColor = "red"; // Set the border colour to red to indicate an error
70         return false;
71     }
72     if (value < 1) {
73         alert("The quantity cannot be less than one!"); // Inform the user that the quantity must be at least one
74         input.style.borderColor = "red"; // Set the border colour to red to indicate an error
75         return false;
76     }
77     if (value > 200) {
78         alert("The quantity cannot be more than 200!"); // Inform the user that the quantity cannot be more than 200
79         input.style.borderColor = "red"; // Set the border colour to red to indicate an error
80         return false;
81     }
82     input.style.borderColor = "green"; // Set the border colour to green if the input is valid
83     return true;
84 }
85 function validateForm() {
86     let isValid = true;
87     // Validate the Description field
88     const descriptionInput = document.getElementById('description');
89     if (!checkDescription(descriptionInput)) {
90         isValid = false;
91     }
92     // Validate the Cost Price field
93     const costPriceInput = document.getElementById('costprice');
94     if (!checkPrice(costPriceInput)) {
95         isValid = false;
96     }
97     // Validate the Retail Price field
98     const retailPriceInput = document.getElementById('retailprice');
99     if (!checkPrice(retailPriceInput)) {
100        isValid = false;
101    }
102    // Validate the Reorder Quantity field
103    const reorderQtyInput = document.getElementById('reorderqty');
104    if (!checkNumber(reorderQtyInput)) {
105        isValid = false;
106    }
107    // Validate the Reorder Level field
108    const reorderLvlInput = document.getElementById('reorderlvl');
109    if (!checkNumber(reorderLvlInput)) {
110        isValid = false;
111    }
112    if (!isValid) {
113        alert("Please correct the errors in the form before submitting."); // Prompt the user to fix errors before submission
114        return false;
115    }
116    return true;
117 }
118 function confirmSubmit() {
119     // Display a confirmation prompt when the form is submitted.
120     return confirm("Are you sure you want to add this stock item?");
121 }

```

```

1  /*
2   Name of screen: amendStock.js
3   Purpose of screen: Populates form, toggles the disabled state of the input fields and also
4   confirms form submission for Amend/View Stock page.
5   Student ID: C00166672
6   Name: Mark Mukiza
7   Date Written: February 2025
8 */
9 // Populates the form fields when a stock item is selected from the listbox.
10 function populate() {
11     var sel = document.getElementById("listbox"); // Retrieve the listbox element by its ID.
12     var result = sel.options[sel.selectedIndex].value; // Get the value of the currently selected option.
13     // If nothing is selected, clear all fields
14     if (result === "") {
15         document.getElementById("amendStockId").value = ""; // Clear the Stock ID field.
16         document.getElementById("amendDescription").value = ""; // Clear the Description field.
17         document.getElementById("amendCostPrice").value = ""; // Clear the Cost Price field.
18         document.getElementById("amendRetailPrice").value = ""; // Clear the Retail Price field.
19         document.getElementById("amendSupplier").value = ""; // Clear the Supplier field.
20         document.getElementById("amendReorderQuantity").value = ""; // Clear the Reorder Quantity field.
21         document.getElementById("amendReorderLevel").value = ""; // Clear the Reorder Level field.
22         document.getElementById("amendSupplierStockCode").value = ""; // Clear the Supplier Stock Code field.
23         document.getElementById("amendQuantityInStockRoom").value = ""; // Clear the Quantity in Stock Room field.
24         document.getElementById("amendDateLastUpdate").value = ""; // Clear the Date Last Update field.
25         return; // Exit the function as there is nothing to populate.
26     }
27     // Split the selected option's value into an array (assuming comma+space separation)
28     var stockDetails = result.split(', '); // Divide the string into its constituent stock details.
29     document.getElementById("amendStockId").value = stockDetails[0]; // Set the Stock ID field.
30     document.getElementById("amendDescription").value = stockDetails[1]; // Set the Description field.
31     document.getElementById("amendCostPrice").value = stockDetails[2]; // Set the Cost Price field.
32     document.getElementById("amendRetailPrice").value = stockDetails[3]; // Set the Retail Price field.
33     document.getElementById("amendReorderQuantity").value = stockDetails[5]; // Set the Reorder Quantity field.
34     document.getElementById("amendReorderLevel").value = stockDetails[6]; // Set the Reorder Level field.
35     document.getElementById("amendSupplierStockCode").value = stockDetails[7]; // Set the Supplier Stock Code field.
36     document.getElementById("amendQuantityInStockRoom").value = stockDetails[8]; // Set the Quantity in Stock Room field.
37     document.getElementById("amendDateLastUpdate").value = stockDetails[9]; // Set the Date Last Update field.
38     /* ChatGPT:
39      This code searches through the supplier select options for the one whose visible text (the supplier name) matches the
40      name provided in your stock details. When it finds a match, it sets that option as selected. As a result, the select
41      box will show the correct supplier name, and when the form is submitted, the supplier id (stored in the option's value)
42      is sent to your PHP code.
43    */
44    // After splitting the stock details
45    var supplierName = stockDetails[4].trim(); // Extract and trim the supplier name from the details.
46    var supplierSelect = document.getElementById("amendSupplier"); // Retrieve the supplier select element.
47    // Loop through the options to find a match on the displayed text
48    for (var i = 0; i < supplierSelect.options.length; i++) {
49        if (supplierSelect.options[i].text.trim() == supplierName) { // Compare each option's text with the supplier name.
50            supplierSelect.selectedIndex = i; // Set the matching option as selected.
51            break; // Exit the loop once a match is found.
52        }
53    }
54 }

```

```

55 function toggleLock() {
56     var btn = document.getElementById("amendViewButton"); // Retrieve the amend view button element.
57     if (document.getElementById("amendViewbutton").value === "Amend Details") { // Check if the button's value is "Amend Details".
58         document.getElementById("amendDescription").disabled = false; // Enable the Description field.
59         document.getElementById("amendCostPrice").disabled = false; // Enable the Cost Price field.
60         document.getElementById("amendRetailPrice").disabled = false; // Enable the Retail Price field.
61         document.getElementById("amendSupplier").disabled = false; // Enable the Supplier field.
62         document.getElementById("amendReorderQuantity").disabled = false; // Enable the Reorder Quantity field.
63         document.getElementById("amendReorderLevel").disabled = false; // Enable the Reorder Level field.
64         document.getElementById("amendSupplierStockCode").disabled = false; // Enable the Supplier Stock Code field.
65         document.getElementById("amendViewbutton").value = "View Details"; // Change the button value to "View Details".
66     } else {
67         document.getElementById("amendDescription").disabled = true; // Disable the Description field.
68         document.getElementById("amendCostPrice").disabled = true; // Disable the Cost Price field.
69         document.getElementById("amendRetailPrice").disabled = true; // Disable the Retail Price field.
70         document.getElementById("amendSupplier").disabled = true; // Disable the Supplier field.
71         document.getElementById("amendReorderQuantity").disabled = true; // Disable the Reorder Quantity field.
72         document.getElementById("amendReorderLevel").disabled = true; // Disable the Reorder Level field.
73         document.getElementById("amendSupplierStockCode").disabled = true; // Disable the Supplier Stock Code field.
74         document.getElementById("amendViewbutton").value = "Amend Details"; // Revert the button value to "Amend Details".
75     }
76 }
77 /* Before submitting this function confirms the action.
78 If confirmed, it enables all fields so that their values are submitted.*/
79 function confirmAmend() {
80     var response = confirm('Are you sure you want to amend the details of this item?'); // Prompt the user for confirmation.
81     if (response) {
82         // Enable all fields so their values are included in the form submission.
83         document.getElementById("amendStockId").disabled = false;
84         document.getElementById("amendDescription").disabled = false;
85         document.getElementById("amendCostPrice").disabled = false;
86         document.getElementById("amendRetailPrice").disabled = false;
87         document.getElementById("amendSupplier").disabled = false;
88         document.getElementById("amendReorderQuantity").disabled = false;
89         document.getElementById("amendReorderLevel").disabled = false;
90         document.getElementById("amendSupplierStockCode").disabled = false;
91         document.getElementById("amendQuantityInStockRoom").disabled = false;
92         document.getElementById("amendDateLastUpdate").disabled = false;
93         return true; // Allow the form to be submitted.
94     } else {
95         // If the user cancels, re-populate the fields to reset any unsaved changes.
96         populate();
97         return false; // Prevent the form submission.
98     }
99 }

```

## Place Orders Screen

**Place Orders**

**Order Details**

Select Supplier: All Suppliers

Select Stock Item: Select an Item

Description:

Quantity in Stock Room:

Reorder Level:

Supplier:

Cost Price (€): 0

Order Quantity: 1

Total Cost (€): 0.00

**Current Order Items**

Description	Quantity	Cost
No items in current order		

Figure 32. Place Orders Screen

```

1  <!--
2      Name of screen:      placeOrders.html.php
3      Purpose of screen: Generates a form to allow the user to place orders for stock items.
4      Student ID:          C00166672
5      Name:                Mark Mukiza
6      Date Written:        March 2025
7  -->
8  <?php
9  session_start();
10 // Include the file that contains order information functions and variables
11 include 'php/orderInformation.php';
12 ?>
13 <!DOCTYPE html>
14 <html lang="en">
15 <head>
16     <meta charset="UTF-8">
17     <title>Place Orders</title>
18     <link rel="stylesheet" href="../styles.css">
19         <link rel="webiste icon" value="png" href="../../C2PLogoNew.png">
20     <script src="placeOrders.js"></script>
21 </head>
22 <body>
23     <!-- Burger icon that will toggle the sidebar -->
24     <div class="burger-menu">#9776;</div>
25     <div class="place-orders-content">
26         <!-- Container for the page header including the title and logo -->
27         <div class="page-header">
28             <h1>Place Orders</h1>
29             <a href="../home.html.php">
30                 
31             </a>
32         </div>
33         <!-- Sidebar navigation -->
34         <?php include '../sidebar.php'; ?>
35         <!-- Include the flash message file to display notifications -->
36         <?php include 'php/flashMessage.php'; ?>
37         <div class="place-orders-container">
38             <!-- Left side: Form area -->
39             <div class="place-orders-form">
40                 <fieldset>
41                     <legend>Order Details</legend>
42                     <!-- Filter form (GET) for supplier selection -->
43                     <form id="filterForm" method="get" action="placeOrders.html.php">
44                         <div class="centered-select">
45                             <div class="inputbox">
46                                 <label for="supplier_id">Filter by Supplier:</label>
47                                 <select name="supplier_id" id="supplier_id" onchange="this.form.submit()">
48                                     <option value="0" <?php if($supplierFilter == 0) echo 'selected'; ?>>-- All Suppliers --</option>
49                                     <?php include 'php/supplierListbox.php'; ?>
50                                 </select>
51                             </div>
52                         </div>
53                         <!-- Clear any previous stock selection when the supplier changes -->
54                         <input type="hidden" name="stock_id" value="0">
55                     </form>

```

```

56    <!-- Listbox for selecting a stock item -->
57    <div class="centered-select">
58      <div class="inputbox">
59        <label for="stockListbox">Select Stock Item:</label>
60        <select name="stockListbox" id="stockListbox" onchange="populateOrderForm()">
61          <option value="">-- Select an Item --</option>
62          <?php include 'php/stockListbox.php'; ?>
63        </select>
64      </div>
65    </div>
66    <!-- Form to submit the order -->
67    <form method="post" action="php/placeOrders.php">
68      <!-- Hidden field to store the selected StockID -->
69      <input type="hidden" name="stock_id" id="stock_id" value="">
70      <div class="inputbox">
71        <label>Description:</label>
72        <!-- Display the description of the selected stock item, if available -->
73        <input type="text" id="description" value="" disabled>
74      </div>
75      <div class="inputbox">
76        <label>Quantity in Stock Room:</label>
77        <!-- Display the quantity in the stock room for the selected item, if available -->
78        <input type="text" id="quantityInStockRoom" value="" disabled>
79      </div>
80      <div class="inputbox">
81        <label>Reorder Level:</label>
82        <!-- Display the reorder level for the selected stock item, if available -->
83        <input type="text" id="reorderLevel" value="" disabled>
84      </div>
85      <div class="inputbox">
86        <label>Supplier:</label>
87        <!-- Display the supplier name for the selected stock item, if available -->
88        <input type="text" id="supplier" value="" disabled>
89      </div>
90      <div class="inputbox">
91        <label>Cost Price (€):</label>
92        <!-- Display the cost price for the selected stock item -->
93        <input type="text" id="costPrice" value="" disabled>
94      </div>
95      <div class="inputbox">
96        <label>Order Quantity:</label>
97        <!-- Display the default order quantity for the selected stock item -->
98        <input type="number" name="order_quantity" id="order_quantity" value="1" min="1" max="200"
99          pattern="^[1-9]\d*$" title="Cannot be less than 1 or greater than 200." oninput="calculateCost()">
100     </div>
101     <div class="inputbox">
102       <label>Total Cost (€):</label>
103       <!-- Display the calculated total cost based on the cost price and order quantity -->
104       <span id="calculatedCost">0.00</span>
105     </div>
106     <!-- Submit buttons to either continue ordering or to finalise the order -->
107     <div class="myButton">
108       <input type="submit" name="action" value="Continue">
109       <input type="submit" name="action" value="Done" onclick="return confirmDone();">
110     </div>
111   </form>
112 </fieldset>
113 </div>

```

```

114 <!-- Right side: Table area -->
115 <!-- This container holds the table that displays the current order items -->
116 <div class="place-orders-table">
117   <!-- Heading for the order items table -->
118   <h3>Current Order Items</h3>
119   <!-- Begin table with the class "place-orders-table" for styling -->
120   <table class="place-orders-table">
121     <!-- Define the table header section -->
122     <thead>
123       <tr>
124         <th>Description</th>
125         <th>Quantity</th>
126         <th>Cost</th>
127       </tr>
128     </thead>
129     <tbody>
130       <?php if (!empty($orderItems)) { ?> <!-- Check if the $orderItems array is not empty -->
131         <?php foreach ($orderItems as $item) { ?> <!-- Loop through each order item in the $orderItems array -->
132           <tr>
133             <!-- Output the order item's description (escaped for safety) -->
134             <td><?php echo htmlspecialchars($item['Description']); ?></td>
135             <!-- Output the order quantity as an integer -->
136             <td><?php echo (int)$item['OrderQuantity']; ?></td>
137             <!-- Output the item cost formatted as a number with two decimals and prefixed with the Euro symbol -->
138             <td>€<?php echo number_format($item['ItemCost'], 2); ?></td>
139           </tr>
140         <?php } ?>
141       <?php } else { ?>
142         <!-- Display a single table row with a message spanning all three columns -->
143         <tr>
144           <td colspan="3">No items in current order</td>
145         </tr>
146       <?php } ?>
147     </tbody>
148   </table>
149 </div>
150   </div>
151 </div>
152 <script src="../sidebar.js"></script>
153 </body>
154 </html>
155

```

```

1  /* Name of screen: placeOrders.js
2   Purpose of screen: Javascript for populating the order form as well as calculating the total cost
3   of the item selected. Also confirms if the order is done.
4   Student ID: C00166672
5   Name: Mark Mukiza
6   Date Written: March 2025
7 */
8 // Populates the order form fields based on the selected stock item.
9 function populateOrderForm() {
10    var sel = document.getElementById("stockListbox"); // Retrieve the stock listbox element by its ID.
11    var result = sel.options[sel.selectedIndex].value; // Get the value of the currently selected option.
12    if (result === "") { // If no option is selected, clear all fields.
13        document.getElementById("stock_id").value = ""; // Clear the hidden Stock ID field.
14        document.getElementById("description").value = ""; // Clear the Description field.
15        document.getElementById("quantityInStockRoom").value = ""; // Clear the Quantity in Stock Room field.
16        document.getElementById("reorderLevel").value = ""; // Clear the Reorder Level field.
17        document.getElementById("supplier").value = ""; // Clear the Supplier field.
18        document.getElementById("costPrice").value = ""; // Clear the Cost Price field.
19        document.getElementById("order_quantity").value = "1"; // Reset the Order Quantity field to 1.
20        document.getElementById("calculatedCost").innerText = "0.00"; // Reset the Total Cost display to 0.00.
21        return; // Exit the function as there is nothing to populate.
22    }
23 // Expected format: "StockID, Description, QuantityInStockRoom, ReorderLevel, SupplierName, CostPrice, ReorderQuantity"
24 var details = result.split(", "); // Divide the string into its constituent stock details.
25 document.getElementById("stock_id").value = details[0]; // Set the Stock ID field.
26 document.getElementById("description").value = details[1]; // Set the Description field.
27 document.getElementById("quantityInStockRoom").value = details[2]; // Set the Quantity in Stock Room field.
28 document.getElementById("reorderLevel").value = details[3]; // Set the Reorder Level field.
29 document.getElementById("supplier").value = details[4]; // Set the Supplier field.
30 document.getElementById("costPrice").value = details[5]; // Set the Cost Price field.
31 document.getElementById("order_quantity").value = details[6]; // Set the Order Quantity field.
32 var totalCost = parseFloat(details[5]) * parseInt(details[6]); // Calculate the total cost by multiplying cost price by order quantity.
33 document.getElementById("calculatedCost").innerText = totalCost.toFixed(2); // Display the total cost, formatted to two decimal places.
34 }
35
36 // Recalculates the total cost when the order quantity changes.
37 function calculateCost() {
38    // Retrieve the cost price and convert it to a float, defaulting to 0.
39    var costPrice = parseFloat(document.getElementById("costPrice").value) || 0;
40    // Retrieve the order quantity and convert it to an integer, defaulting to 0.
41    var quantity = parseInt(document.getElementById("order_quantity").value) || 0;
42    var total = costPrice * quantity; // Calculate the total cost.
43    document.getElementById("calculatedCost").innerText = total.toFixed(2); // Update the Total Cost display with the calculated value.
44 }
45 // Confirmation function for the "Done" button.
46 function confirmDone() {
47    return confirm("Are you sure you want to finalise this order?"); // Prompt the user for confirmation and return the result.
48 }
49

```

```

1 <!--
2   Name of screen:      placeOrders.php
3   Purpose of screen:  Processes the order form submissions, adds items to session, completes orders,
4                       and creates an $orderSummary for display on placeOrders.html.php.
5   Student ID:         C00166672
6   Name:               Mark Mukiza
7   Date Written:       March 2025
8 -->
9 <?php
10 session_start();
11 include_once '../db.inc.php';
12 // Ensure our session order array is ready
13 if (!isset($_SESSION['order'])) {
14     $_SESSION['order'] = array();
15 }
16 $error = ''; // Will store any error messages
17 $action = isset($_POST['action']) ? $_POST['action'] : '';
18 $stockID = isset($_POST['stock_id']) ? (int) $_POST['stock_id'] : 0;
19 $orderQuantity = isset($_POST['order_quantity']) ? (int) $_POST['order_quantity'] : 0;
20 if ($action === 'Continue') {
21     if ($stockID < 1) {
22         $error = "No stock item selected. Please choose a valid stock item.";
23     } else {
24         // Fetch selected stock details from the database
25         $sql = "SELECT s.StockID, s.Description, s.CostPrice, s.SupplierID, sup.Name AS SupplierName
26             FROM Stock s
27             JOIN Suppliers sup ON s.SupplierID = sup.SupplierID
28             WHERE s.StockID = $stockID
29             AND s.DeletedFlag = 0";
30         $result = mysqli_query($con, $sql);
31         if ($result && mysqli_num_rows($result) > 0) {
32             $stock = mysqli_fetch_assoc($result);
33             // Check if the new item's supplier conflicts with existing order items
34             if (!empty($_SESSION['order'])) {
35                 $firstItem = $_SESSION['order'][0];
36                 if ($firstItem['SupplierID'] !== $stock['SupplierID']) {
37                     $error = "All items must be from the same supplier: " . htmlspecialchars($firstItem['SupplierName']);
38                 }
39             }
40             if (empty($error)) {
41                 $itemCost = $stock['CostPrice'] * $orderQuantity;
42                 $_SESSION['order'][] = array(
43                     'StockID'      => $stock['StockID'],
44                     'Description'  => $stock['Description'],
45                     'OrderQuantity' => $orderQuantity,
46                     'CostPrice'    => $stock['CostPrice'],
47                     'ItemCost'     => $itemCost,
48                     'SupplierID'   => $stock['SupplierID'],
49                     'SupplierName' => $stock['SupplierName']
50                 );
51             }
52         } else {
53             $error = "Invalid stock item selected.";
54         }
55     }
56 } elseif ($action === 'Done') {

```

```

57 // Finalize the order
58 if (empty($_SESSION['order'])) {
59     $error = "Cannot complete an empty order!";
60 } else {
61     $firstItem = $_SESSION['order'][0];
62     $stockID = $firstItem['StockID'];
63     // Calculate total cost
64     $totalCost = 0;
65     foreach ($_SESSION['order'] as $item) {
66         $totalCost += $item['ItemCost'];
67     }
68     // Insert the order into the Orders table
69     $insertOrderSql = "INSERT INTO Orders (StockID, DateOfOrder, TotalCost)
70                         VALUES ($stockID, NOW(), $totalCost)";
71     $insertOrderRes = mysqli_query($con, $insertOrderSql);
72     if (!$insertOrderRes) {
73         $error = "Failed to insert into Orders: " . mysqli_error($con);
74     } else {
75         $orderId = mysqli_insert_id($con);
76         // Insert each item into the OrderItems table
77         foreach ($_SESSION['order'] as $item) {
78             $insertItemsSql = "INSERT INTO OrderItems
79                             (OrderID, StockID, SupplierID, QuantityOrdered)
80                             VALUES (
81                                 $orderId,
82                                 " . (int)$item['StockID'] . ",
83                                 " . (int)$item['SupplierID'] . ",
84                                 " . (int)$item['OrderQuantity'] . "
85                             )";
86             $execItems = mysqli_query($con, $insertItemsSql);
87             if (!$execItems) {
88                 $error = "Failed to insert order item: " . mysqli_error($con);
89                 break;
90             }
91         }
92         if (empty($error)) {
93             // Retrieve the DateOfOrder just inserted
94             $dateQuery = "SELECT DateOfOrder FROM Orders WHERE OrderID = $orderId";
95             $dateResult = mysqli_query($con, $dateQuery);
96             $dateOfOrder = '';
97             if ($dateResult && mysqli_num_rows($dateResult) > 0) {
98                 $row = mysqli_fetch_assoc($dateResult);
99                 $dateOfOrder = $row['DateOfOrder'];
100            }
101            // Build an order summary array for display
102            $orderSummary = array(
103                'OrderID'      => $orderId,
104                'DateOfOrder'  => $dateOfOrder,
105                'SupplierName' => $firstItem['SupplierName'],
106                'TotalCost'    => $totalCost,
107                'Items'        => $_SESSION['order']
108            );
109            $_SESSION['orderSummary'] = $orderSummary;
110            // Clear the order array
111            $_SESSION['order'] = array();
112        }
113    }
114 }
115 }
116 // If there's an error, store it in session for display
117 if (!empty($error)) {
118     $_SESSION['msg'] = $error;
119     $_SESSION['msg_type'] = 'error';
120 }
121 mysqli_close($con);
122 header("Location: ../placeOrders.html.php");
123 exit;
124 ?>

```

```

1 <!--
2     Name of screen:      orderInformation.php
3     Purpose of screen:  Retrieves session-based data and database records needed by the placeOrders.html.php file.
4     Student ID:         C00166672
5     Name:               Mark Mukiza
6     Date Written:       March 2025
7 -->
8 <?php
9 session_start();
10 include_once '../../../../../db.inc.php';
11 /* Retrieve any session-based flash messages. */
12 $flashMessage = '';
13 $className = '';
14 if (isset($_SESSION['msg'])) {
15     $flashMessage = $_SESSION['msg'];
16     $className = (isset($_SESSION['msg_type']) && $_SESSION['msg_type'] === 'error') ? 'error' : 'success';
17     unset($_SESSION['msg'], $_SESSION['msg_type']);
18 }
19 /* Retrieve an $orderSummary for display if an order was just completed. */
20 $orderSummary = null;
21 if (isset($_SESSION['orderSummary'])) {
22     $orderSummary = $_SESSION['orderSummary'];
23     unset($_SESSION['orderSummary']);
24 }
25 /* Use supplier filter from GET parameters. */
26 $supplierFilter = 0;
27 if (isset($_GET['supplier_id'])) {
28     $supplierFilter = (int) $_GET['supplier_id'];
29 }
30 /* Retrieve supplier list. */
31 $supplierList = array();
32 $sqlSup = "SELECT SupplierID, Name FROM Suppliers WHERE DeletedFlag = 0 ORDER BY Name";
33 $resultSup = mysqli_query($con, $sqlSup);
34 if ($resultSup) {
35     while ($row = mysqli_fetch_assoc($resultSup)) {
36         $supplierList[] = $row;
37     }
38 }
39 /* Retrieve the stock list with all details needed for the listbox.
40     (StockID, Description, QuantityInStockRoom, ReorderLevel, CostPrice, ReorderQuantity, and SupplierName.)
41 */
42 $stockList = array();
43 $stockSql = "SELECT s.StockID, s.Description, s.QuantityInStockRoom, s.ReorderLevel,
44                     s.CostPrice, s.ReorderQuantity, sup.Name AS SupplierName
45                 FROM Stock s
46                 JOIN Suppliers sup ON s.SupplierID = sup.SupplierID
47                 WHERE s.DeletedFlag = 0";
48 if ($supplierFilter > 0) {
49     $stockSql .= " AND s.SupplierID = " . (int)$supplierFilter;
50 }
51 $stockSql .= " ORDER BY s.Description";
52 $resultStock = mysqli_query($con, $stockSql);
53 if ($resultStock) {
54     while ($row = mysqli_fetch_assoc($resultStock)) {
55         $stockList[] = $row;
56     }
57 }
58 $selectedStock = null;
59 /* Retrieve any current order items from the session to display them in the table. */
60 $orderItems = array();
61 if (isset($_SESSION['order']) && is_array($_SESSION['order'])) {
62     $orderItems = $_SESSION['order'];
63 }
64 mysqli_close($con);
65 ?>
```

```

1  <!--
2      Name of screen:      flashMessage.php
3      Purpose of screen:  Creates flash messages for order success or failure.
4      Student ID:        C00166672
5      Name:              Mark Mukiza
6      Date Written:      March 2025
7  -->
8  <!-- If there's a generic flash message (error or success), display it. -->
9  <?php if (!empty($flashMessage)) { ?>
10     <div class="<?php echo $className; ?>" id="flashMessage">
11         <?php echo htmlspecialchars($flashMessage); ?>
12     </div>
13     <script>
14         setTimeout(function () {
15             var flashDiv = document.getElementById('flashMessage');
16             if (flashDiv) {
17                 flashDiv.style.display = 'none';
18             }
19         }, 5000);
20     </script>
21 <?php } ?>
22 <!-- If there's a completed order summary, display the success message. -->
23 <?php if (!empty($orderSummary)) { ?>
24     <div class="place-orders-success">
25         <h3>Order #<?php echo htmlspecialchars($orderSummary['OrderID']); ?> Confirmation</h3>
26         <p><strong>Date Ordered:</strong>
27             <?php
28                 // Format the date: d/m/Y H:i
29                 echo date('d/m/Y H:i', strtotime($orderSummary['DateOfOrder']));
30             ?>
31         </p>
32         <p><strong>Supplier:</strong>
33             <?php echo htmlspecialchars($orderSummary['SupplierName']); ?>
34         </p>
35         <p><strong>Total Cost:</strong>
36             €<?php echo number_format($orderSummary['TotalCost'], 2); ?>
37         </p>
38         <div class="place-orders-order-items">
39             <h4>Ordered Items:</h4>
40             <ul>
41                 <?php foreach ($orderSummary['Items'] as $item) { ?>
42                     <li>
43                         <?php echo htmlspecialchars($item['Description']); ?> -
44                         Quantity: <?php echo (int) $item['OrderQuantity']; ?> -
45                         Total: €<?php echo number_format($item['ItemCost'], 2); ?>
46                     </li>
47                 <?php } ?>
48             </ul>
49         </div>
50         <script>
51             // Hide the success message after 7 seconds
52             setTimeout(function () {
53                 var successDiv = document.querySelector('.place-orders-success');
54                 if (successDiv) {
55                     successDiv.style.display = 'none';
56                 }
57             }, 7000);
58         </script>
59     <?php } ?>

```

```

1 <!--
2   Name of screen: supplierListbox.php
3   Purpose of screen: Generates the listbox for suppliers that are in the Supplier table in the Hotel database.
4   Student ID: C00166672
5   Name: Mark Mukiiza
6   Date Written: March 2025
7 -->
8 <?php
9 // Include the database connection file.
10 include '../../db.inc.php';
11 // Determine the currently selected supplier from GET parameters.
12 $supplierFilter = 0;
13 if (isset($_GET['supplier_id'])) {
14     $supplierFilter = (int) $_GET['supplier_id'];
15 }
16 // Prepare the SQL query to select active suppliers, ordered by Name.
17 $sql = "SELECT SupplierID, Name FROM Suppliers WHERE DeletedFlag = 0 ORDER BY Name";
18 $result = mysqli_query($con, $sql);
19 if ($result && mysqli_num_rows($result) > 0) {
20     while ($row = mysqli_fetch_assoc($result)) {
21         $supplierID = (int)$row['SupplierID'];
22         $supplierName = $row['Name'];
23         $selected = ($supplierID === $supplierFilter) ? "selected" : "";
24         echo "<option value='" . htmlspecialchars($supplierID) . "' $selected>" . htmlspecialchars($supplierName) . "</option>";
25     }
26 } else {
27     echo "<option value='' disabled>No suppliers available.</option>";
28 }
29 mysqli_close($con);
30 ?>
31
32
33

```

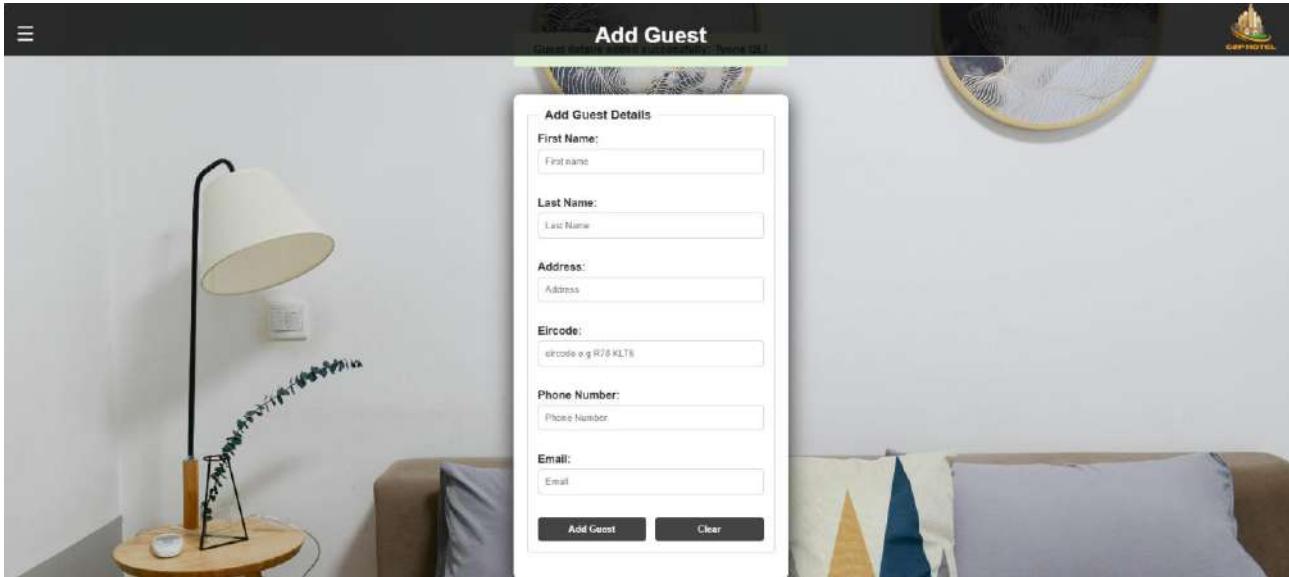
```

1  <!--
2   Name of screen:      stockListbox.php
3   Purpose of screen: Generates the listbox for stock that are in the stock table in the Hotel database.
4   Student ID:          C00166672
5   Name:                Mark Mukiza
6   Date Written:        March 2025
7 -->
8 <?php
9 // Include the database connection file.
10 include '../db.inc.php';
11 // Determine the currently selected stock ID from GET parameters.
12 $selectedStockID = 0;
13 if (isset($_GET['stock_id'])) {
14     $selectedStockID = (int) $_GET['stock_id'];
15 }
16 // Prepare the SQL query to select active stock items for the current supplier filter.
17 $sql = "SELECT s.StockID, s.Description, s.QuantityInStockRoom, s.ReorderLevel,
18         s.CostPrice, s.ReorderQuantity, sup.Name AS SupplierName
19     FROM Stock s
20     JOIN Suppliers sup ON s.SupplierID = sup.SupplierID
21     WHERE s.DeletedFlag = 0";
22 if (isset($_GET['supplier_id']) && (int)$_GET['supplier_id'] > 0) {
23     $sql .= " AND s.SupplierID = " . (int) $_GET['supplier_id'];
24 }
25 $sql .= " ORDER BY s.Description";
26 $result = mysqli_query($con, $sql);
27 if (!$result) {
28     die('Error querying the database: ' . mysqli_error($con));
29 }
30 while ($row = mysqli_fetch_assoc($result)) {
31     $stockID          = $row['StockID'];
32     $description      = $row['Description'];
33     $quantityInStockRoom = $row['QuantityInStockRoom'];
34     $reorderLevel     = $row['ReorderLevel'];
35     $supplierName     = $row['SupplierName'];
36     $costPrice        = $row['CostPrice'];
37     $reorderQuantity  = $row['ReorderQuantity'];
38     // Build a comma-separated string in the expected order.
39     $optionValue = $stockID . ", " . $description . ", " . $quantityInStockRoom . ", "
40             . $reorderLevel . ", " . $supplierName . ", " . $costPrice . ", " . $reorderQuantity;
41     $selected = ($stockID == $selectedStockID) ? "selected" : "";
42     echo "<option value='" . htmlspecialchars($optionValue) . "' $selected>" . htmlspecialchars($description) . "</option>";
43 }
44 mysqli_close($con);
45 ?>
46
47
48

```

Hamed Zon Written By: Hamed Zon

## Add Guest Screen



```

1 <?php
2 // Hamed Zon
3 // C00259228
4 // This HTML code allows you to add a guest. It includes some PHP and JavaScript to check validations.
5 //March 2025
6 // Start a PHP session to manage session variables
7 session_start();
8 ?>
9 <html lang="en">
10 <head>
11     <!-- Set the character encoding for the document --&gt;
12     &lt;meta charset="UTF-8"&gt;
13     <!-- Set the title of the page --&gt;
14     &lt;title&gt;Add Guest&lt;/title&gt;
15     <!-- Link to an external stylesheet for styling --&gt;
16     &lt;link rel="stylesheet" href="../../styles.css"&gt;
17     &lt;link rel="webiste icon" value="png" href="../../C2PLogoNew.png"&gt;
18 &lt;/head&gt;
19 &lt;body&gt;
20     <!-- Burger icon that will toggle the sidebar --&gt;
21     &lt;div class="burger-menu"&gt;&amp;#9776;&lt;/div&gt;
22     <!-- Page header with a title and logo --&gt;
23     &lt;div class="page-header"&gt;
24         &lt;h1&gt;Add Guest&lt;/h1&gt;
25         <!-- Link to the home page with the hotel logo --&gt;
26         &lt;a href="../../home.html.php"&gt;&lt;img src="../../C2PLogoNew.png" alt="C2P Hotel Logo" class="logo"&gt;&lt;/a&gt;
27     &lt;/div&gt;
28     <!-- Sidebar navigation --&gt;
29     &lt;?php include '../../sidebar.php'; ?&gt;</pre>

```

```

30 <!-- Main content area -->
31 <div class="content add-page">
32     <?php
33     // Check if a session message is set
34     if (isset($_SESSION['add_message'])) {
35         // Display the session message inside an <h1>
36         echo "<h1 class='success'>" . htmlspecialchars($_SESSION['add_message']) . "</h1>";
37         // Clear the session message after displaying it
38         unset($_SESSION['add_message']);
39     }
40 ?>
41 <!-- Form to add a guest -->
42 <form action="addGuest.php" onsubmit="return confirmCheck()" method="POST">
43     <fieldset>
44         <legend>Add Guest Details</legend>
45         <!-- JavaScript for form validation and confirmation -->
46         <script>
47             function validateForm() {
48                 // Validate First Name
49                 var firstName = document.getElementById("firstname").value; // Get the value of the first name field
50
51                 // Regex to allow only letters (no numbers, spaces, or special characters)
52                 var nameRegex = /^[A-Za-z]+$/;
53
54                 if (firstName === "" || !nameRegex.test(firstName)) { // Check if the field is empty or doesn't match the regex
55                     alert("First Name is required and should only contain letters (no numbers, spaces, or special characters)."); //
56                     Show an error message
57                     return false; // Prevent form submission
58                 }
59
60                 // Validate Last Name
61                 var lastName = document.getElementById("lastname").value; // Get the value of the last name field
62
63                 if (lastName === "" || !nameRegex.test(lastName)) { // Check if the field is empty or doesn't match the regex
64                     alert("Last Name is required and should only contain letters (no numbers, spaces, or special characters)."); //
65                     Show an error message
66                     return false; // Prevent form submission
67                 }
68
69                 // Validate Phone Number
70                 var phone = document.getElementById("phone").value;
71                 var phonePattern = /^\d{3} ?\d{7}$/; // Phone number takes 10 digits with an optional space (e.g., 087 4890954)
72                 if (!phonePattern.test(phone)) { // If phone number does not match the pattern
73                     alert("Please enter a valid phone number (e.g., 087 4890954)."); // Print message
74                     return false;
75                 }
76
77                 // Validate Eircode
78                 var eircode = document.getElementById("eircode").value;
79                 var eircodePattern = /^[A-Z][0-9]{2} ?[A-Z0-9]{4}$/; // Eircode format: 1 letter, 2 digits, optional space, 4
80                 alphanumeric characters (e.g., R95 KTX3)
81                 if (!eircodePattern.test(eircode)) { // If Eircode is not valid
82                     alert("Please enter a valid Eircode (e.g., R95 KTX3)."); // Print message
83                     return false;
84                 }

```

```

83 // Validate Email Address
84 var email = document.getElementById("email").value;
85 var emailPattern = /^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$/; // Standard email format
86 if (!emailPattern.test(email)) { // If email is not valid
87     alert("Please enter a valid email address."); // Print message
88     return false;
89 }
90
91 // Validate Address
92 var address = document.getElementById("email").value;
93 var emailPattern = /^[0-9a-zA-Z\s]+$/; // Standard email format
94 if (!emailPattern.test(address)) { // If address is not valid
95     alert("Please enter a valid address."); // Print message
96     return false;
97 }
98
99 // If all validations pass
100 return true;
101 }
102
103
104 function confirmCheck() {
105     // Ask for confirmation before submitting the form
106     var response = confirm('Are these details correct?');
107     if (response) {
108         // Enable form fields before submission
109         document.getElementById("firstname").disabled = false;
110         document.getElementById("lastname").disabled = false;
111         document.getElementById("address").disabled = false;
112         document.getElementById("eircode").disabled = false;
113         document.getElementById("phone").disabled = false;
114         document.getElementById("email").disabled = false;
115         return true;
116     }
117     return false;
118 }
119 </script>
120 <!-- Input fields for guest details -->
121 <div class="inputbox">
122     <label for="firstname">First Name:</label>
123     <input type="text" name="firstname" id="firstname" placeholder="First name" pattern="[A-Za-z\s]+" required/><br><br><!--
it only take lowcase and uppcases laters and spaces-->
124     </div>
125     <div class="inputbox">
126         <label for="lastname">Last Name:</label>
127         <input type="text" name="lastname" id="lastname" placeholder="Last Name" pattern="[A-Za-z\s]+" required/><br><br><!--
only take lowcase and uppcases laters and spaces-->
128     </div>
129     <div class="inputbox">
130         <label for="address">Address:</label>
131         <input type="text" name="address" id="address" placeholder="Address" pattern="[0-9a-zA-Z\s]+" required/><br><br> <!--
only take number from 0 to 9, lowcase and uppcases laters and spaces-->
132     </div>

```

```

133     <div class="inputbox">
134         <label for="eircode">Eircode:</label>
135         <input type="text" name="eircode" id="eircode" placeholder="eircode e.g R78 KLT6" pattern="[A-Z]{1}\d{2}\s?[A-Za-z]{3}\d{1}" required/><br><!--it only one uppercase and 2 digits and optional space and 3 low or uppercase letters and one digit-->
136         </div>
137         <div class="inputbox">
138             <label for="phone">Phone Number:</label>
139             <input type="text" name="phone" id="phone" placeholder="Phone Number" pattern="^\d{3} \d{7}$" required/><br><br><!--it
only take 10 digits and optional space-->
140             </div>
141             <div class="inputbox">
142                 <label for="email">Email:</label>
143                 <input type="email" name="email" id="email" placeholder="Email" required/><br><br>
144             </div>
145             <!-- Buttons to submit or clear the form -->
146             <div class="myButton">
147                 <input type="submit" value="Add Guest" onclick="return validateForm()"/>
148                 <input type="reset" value="Clear"/>
149             </div>
150         </fieldset>
151     </form>
152 </div>
153 <!-- JavaScript for sidebar and dropdown functionality -->
154 <script>
155     document.addEventListener('DOMContentLoaded', function () {
156         // ===== SIDEBAR =====
157         const burger = document.querySelector('.burger-menu');
158         const sidebar = document.querySelector('.sidebar');
159
160         // Toggle sidebar on burger click
161         burger.addEventListener('click', (e) => {
162             e.stopPropagation();
163             sidebar.classList.toggle('open');
164         });
165
166         // Close sidebar when clicking outside
167         document.addEventListener('click', (e) => {
168             if (!sidebar.contains(e.target) && !e.target.classList.contains('burger-menu')) {
169                 sidebar.classList.remove('open');
170             }
171         });
172
173         // ===== DROPPEDowns =====
174         const dropdowns = document.querySelectorAll('.dropdown');
175         let activeDropdown = null;
176
177         dropdowns.forEach(dropdown => {
178             const content = dropdown.querySelector('.dropdown-content');
179
180             // Show dropdown on hover
181             dropdown.addEventListener('mouseenter', () => {
182                 if (activeDropdown) {
183                     activeDropdown.classList.remove('visible');
184                 }
185                 content.classList.add('visible');
186                 activeDropdown = content;
187             });

```

```
186     activeDropdown = content;
187 });
188
189 // Hide dropdown with delay
190 dropdown.addEventListener('mouseleave', () => {
191     setTimeout(() => {
192         if (!content.matches(':hover')) {
193             content.classList.remove('visible');
194             activeDropdown = null;
195         }
196     }, 10);
197 });
198
199 // Keep dropdown open when hovering its content
200 content.addEventListener('mouseenter', () => {
201     clearTimeout();
202     content.classList.add('visible');
203 });
204
205 // Hide dropdown when leaving its content
206 content.addEventListener('mouseleave', () => {
207     content.classList.remove('visible');
208     activeDropdown = null;
209 });
210 });
211 });
212 </script>
213 </body>
214 </html>
```

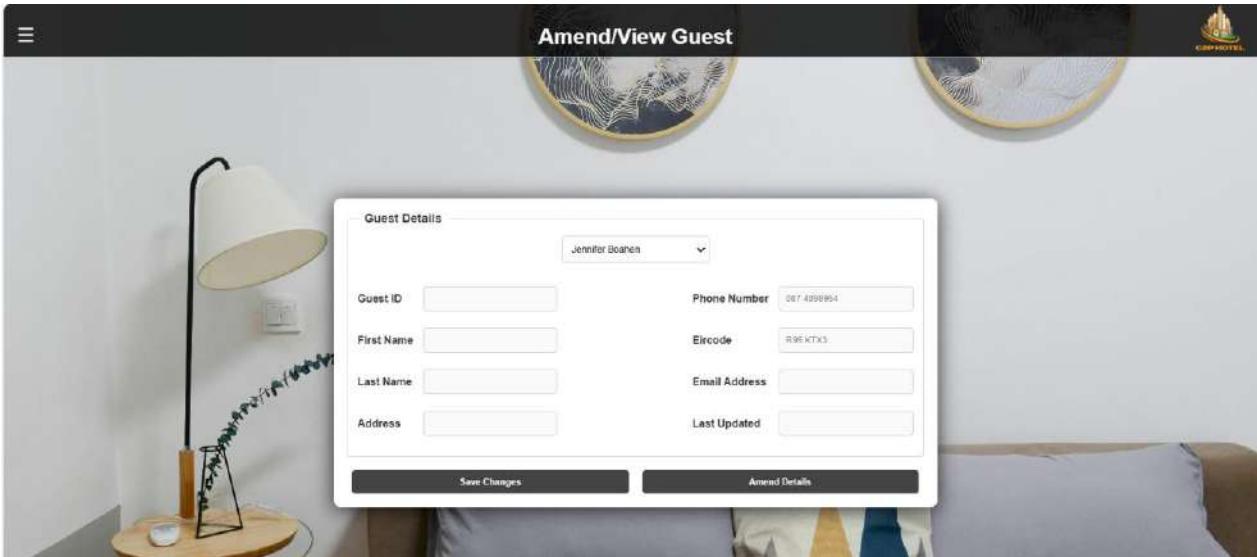
*addGuest.php*

```

1 |?php
2 // Hamed Zon
3 // c00259228
4 // This is the PHP file is to add guest details and send message to the html page
5
6 // Start a session to manage session variables
7 session_start();
8
9 // Include the database connection file
10 include 'db.inc.php';
11
12 // Check if the form was submitted using the POST method and if the 'first' field is set
13 if ($_SERVER['REQUEST_METHOD'] === 'POST' && isset($_POST['firstname'])) {
14     // Sanitize input to prevent SQL injection
15
16     //this add backslash before special characters so no one can attack the query
17     $firstName = mysqli_real_escape_string($con, $_POST['firstname']); //firstname
18     $surname = mysqli_real_escape_string($con, $_POST['lastname']); //lastname//
19     $address = mysqli_real_escape_string($con, $_POST['address']); //address
20     $eircode = mysqli_real_escape_string($con, $_POST['eircode']); //eircode
21     $phone = mysqli_real_escape_string($con, $_POST['phone']); //phne
22     $email = mysqli_real_escape_string($con, $_POST['email']); //email
23
24     // SQL query to insert a new guest
25     $sql = "INSERT INTO Guests (FirstName, Surname, Address, Eircode, TelephoneNumber, EmailAddress, DateLastUpdate)
26             VALUES ('$firstName', '$surname', '$address', '$eircode', '$phone', '$email', NOW())";
27
28     // Execute the SQL query
29     if (mysqli_query($con, $sql)) {
30         // Check if a row was actually inserted
31         if (mysqli_affected_rows($con) > 0) {
32             $_SESSION['add_message'] = "Guest details added successfully: $firstName $surname!";
33         } else {
34             $_SESSION['add_message'] = "No guest details were added.";
35         }
36     } else {
37         // Handle SQL errors
38         $_SESSION['add_message'] = "Error adding guest: " . mysqli_error($con);
39     }
40
41     // Close the database connection
42     mysqli_close($con);
43
44     // Redirect the user back to the add guest page
45     header("Location: addguest.html.php");
46     exit();
47 } else {
48     // If form submission is invalid
49     $_SESSION['add_message'] = "No guest added.";
50
51     // Redirect the user back to the add guest page
52     header("Location: addguest.html.php");
53     exit();
54 }
55 ?>

```

## Amend/View Guest Screen



```

<?php
//Hamed Zon
//C00259228
//March 2025
//This code allows you to update and view a guest
// Start a PHP session to manage session variables

session_start();
?>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="../../styles.css">
    <link rel="webiste icon" value="png" href="../../C2PLogoNew.png">
    <title>Amend/View Guest</title>
  </head>
  <body>
<!-- Burger icon that will toggle the sidebar --&gt;
  &lt;div class="burger-menu"&gt;#9776;&lt;/div&gt;
  &lt;div class="page-header"&gt;
    &lt;h1&gt;Amend/View Guest&lt;/h1&gt;
    &lt;a href="../../home.html.php"&gt;&lt;img src="../../C2PLogoNew.png" alt="C2P Hotel Logo" class="logo"&gt;&lt;/a&gt;
  &lt;/div&gt;
<!-- Sidebar navigation --&gt;
  &lt;?php include '../../sidebar.php'; ?&gt;

  &lt;div class="content amend-view-page"&gt;
    &lt;script&gt;
</pre>

```

```

29      <script>
30
31  function validateForm() {
32      // Validate First Name
33  var firstName = document.getElementById("amendfname").value; // Get the value of the first name field
34
35      // Regex to allow only letters (no numbers, spaces, or special characters)
36  var nameRegex = /^[A-Za-z]+$/;
37
38  if (firstName === "" || !nameRegex.test(firstName)) { // Check if the field is empty or doesn't match the regex
39      alert("First Name is required and should only contain letters (no numbers, spaces, or special characters)."); // Show an error
40  message
41      return false; // Prevent form submission
42  }
43
44  // Validate Last Name
45  var lastName = document.getElementById("amendsname").value; // Get the value of the last name field
46
47  if (lastName === "" || !nameRegex.test(lastName)) { // Check if the field is empty or doesn't match the regex
48      alert("Last Name is required and should only contain letters (no numbers, spaces, or special characters)."); // Show an error
49  message
50      return false; // Prevent form submission
51  }
52
53  // Validate Phone Number
54  var phone = document.getElementById("amendphone").value;
55  var phonePattern = /\d{3} ?\d{7}$/; // Phone number takes 10 digits with an optional space (e.g., 087 4890954)
56  if (!phonePattern.test(phone)) { // If phone number does not match the pattern
57      alert("Please enter a valid phone number (e.g., 087 4890954)."); // Print message
58      return false;
59  }
60
61  // Validate Eircode
62  var eircode = document.getElementById("amendeircode").value;
63  var eircodePattern = /^[A-Z][0-9]{2} ?[A-Z0-9]{4}$/; // Eircode format: 1 letter, 2 digits, optional space, 4 alphanumeric characters
64  (e.g., R95 KTX3)
65  if (!eircodePattern.test(eircode)) { // If Eircode is not valid
66      alert("Please enter a valid Eircode (e.g., R95 KTX3)."); // Print message
67      return false;
68  }
69
70  // Validate Email Address
71  var email = document.getElementById("amendemail").value;
72  var emailPattern = /^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}\$/; // Standard email format
73  if (!emailPattern.test(email)) { // If email is not valid
74      alert("Please enter a valid email address."); // Print message
75      return false;
76  }
77
78  // Validate Address
79  var address = document.getElementById("amendaddress").value;
80  var addressPattern = /^[a-zA-Z0-9\s]+$/; // Standard address format
81  if (!addressPattern.test(address)) { // If email is not valid
82      alert("Please enter a valid address."); // Print message
83      return false;
84  }
85
86  // If all validations pass

```

```

83     // If all validations pass
84     return true;
85 }
86
87
88 // Function to handle form submission
89 function confirmCheck() {
90     var response = confirm('Are these details correct?');
91     if (response) {
92         document.getElementById("amendid").disabled = false;
93         document.getElementById("amendfname").disabled = false;
94         document.getElementById("amendsname").disabled = false;
95         document.getElementById("amendaddress").disabled = false;
96         document.getElementById("amendeircode").disabled = false;
97         document.getElementById("amendphone").disabled = false;
98         document.getElementById("amendemail").disabled = false;
99         return true; // Validate before submitting
100    } else {
101        populate();
102        toggleLock();
103
104        return false;
105    }
106 }
107
108 // Function to toggle the lock/unlock on the form fields
109 function toggleLock() {
110    if (document.getElementById("amendView").value == "Amend Details") {
111        document.getElementById("amendfname").disabled = false;
112        document.getElementById("amendsname").disabled = false;
113        document.getElementById("amendaddress").disabled = false;
114        document.getElementById("amendeircode").disabled = false;
115        document.getElementById("amendphone").disabled = false;
116        document.getElementById("amendemail").disabled = false;
117        document.getElementById("amendViewbutton").value = "View Details";
118    } else {
119
120        document.getElementById("amendfname").disabled = true;
121        document.getElementById("amendsname").disabled = true;
122        document.getElementById("amendaddress").disabled = true;
123        document.getElementById("amendeircode").disabled = true;
124        document.getElementById("amendphone").disabled = true;
125        document.getElementById("amendemail").disabled = true;
126        document.getElementById("amendViewbutton").value = "View Details";
127    }
128 }
129
130 // Function to populate fields
131 function populate() {
132     var sel = document.getElementById("listbox");
133     var result = sel.options[sel.selectedIndex].value;
134     var personDetails = result.split(',');
135
136     document.getElementById("amendid").value = personDetails[0];
137     document.getElementById("amendfname").value = personDetails[1];
138     document.getElementById("amendsname").value = personDetails[2];
139     document.getElementById("amendaddress").value = personDetails[3];
140     document.getElementById("amendeircode").value = personDetails[4];
141     document.getElementById("amendphone").value = personDetails[5];

```

```

141     document.getElementById("amendphone").value = personDetails[5];
142     document.getElementById("amendemail").value = personDetails[6];
143     document.getElementById("amendlastupdate").value = personDetails[7];
144   }
145 </script>
146
147         <?php
148 // Check if a session message is set (e.g., success or error message)
149 if (isset($_SESSION['update_message'])) {
150     // Display the session message inside an <h1> element with a class for styling
151     echo "<h1 class='success'>" . htmlspecialchars($_SESSION['update_message']) . "</h1>";
152     // Clear the session message after displaying it
153     unset($_SESSION['update_message']);
154 }
155 ?>
156 <form action="amendview.php" onsubmit="confirmCheck()" method="post">
157   <fieldset>
158     <legend>Guest Details</legend>
159     <div class="centered-select">
160       <div class="inputbox"><?php include 'guestListBox.php';?></div>
161       </div>
162       <div class="columns">
163         <div class = "left-column">
164           <div class="inputbox">
165             <label for="amendid">Guest ID</label>
166             <input type="text" name="amendid" id="amendid" disabled><br><br>
167           </div>
168
169           <div class="inputbox">
170             <label for="amendfname">First Name</label>
171             <input type="text" name="amendfname" id="amendfname" disabled><br><br>
172           </div>
173
174           <div class="inputbox">
175             <label for="amendsname">Last Name</label>
176             <input type="text" name="amendsname" id="amendsname" disabled><br><br>
177           </div>
178
179           <div class="inputbox">
180             <label for="amendaddress">Address</label>
181             <input type="text" name="amendaddress" id="amendaddress" pattern="^A-Za-z0-9\s.,'-/]+$" disabled><br><br>
182           </div>
183         </div>
184         <div class = "right-column">
185           <div class="inputbox">
186             <label for="amendphone">Phone Number</label>
187             <input type="text" name="amendphone" id="amendphone" placeholder="087 4890954" pattern="\d{3} \d{7}" title="Enter a valid 10-digit phone number" disabled><br><br>
188           </div>
189
190           <div class="inputbox">
191             <label for="amendeircode">Eircode</label>
192             <input type="text" name="amendeircode" id="amendeircode" placeholder="R95 KTX3" pattern="^A-Z[0-9]{2} [A-Z0-9]{4}$" disabled>
193           <br><br>
194           </div>
195
196           <div class="inputbox">

```

```

190      <label for="amendemail">Email Address</label>
191      <input type="email" name="amendemail" id="amendemail" disabled><br><br>
192    </div>
193
194    <div class="inputbox">
195      <label for="amendlastupdate">Last Updated</label>
196      <input type="text" name="amendlastupdate" id="amendlastupdate" disabled><br><br>
197    </div>
198    </div>
199    </div>
200  </fieldset>
201
202  <div class="myButton">
203    <input type="submit" value="Save Changes" onclick="return validateForm()">
204    <input type="button" value="Amend Details" id="amendView" onclick="toggleLock()">
205  </div>
206 </form>
207 </div>
208 <script>
209   document.addEventListener('DOMContentLoaded', function () {
210     // ===== SIDEBAR =====
211     const burger = document.querySelector('.burger-menu');
212     const sidebar = document.querySelector('.sidebar');
213
214     // Toggle sidebar on burger click
215     burger.addEventListener('click', (e) => {
216       e.stopPropagation();
217       sidebar.classList.toggle('open');
218     });
219
220     // Close sidebar when clicking outside
221     if (!sidebar.contains(e.target) && !e.target.classList.contains('burger-menu')) {
222       sidebar.classList.remove('open');
223     }
224   });
225
226   // ===== DROPPEDS =====
227   const dropdowns = document.querySelectorAll('.dropdown');
228   let activeDropdown = null;
229
230   dropdowns.forEach(dropdown => {
231     const content = dropdown.querySelector('.dropdown-content');
232
233     // Show dropdown on hover
234     dropdown.addEventListener('mouseenter', () => {
235       if (activeDropdown) {
236         activeDropdown.classList.remove('visible');
237       }
238       content.classList.add('visible');
239       activeDropdown = content;
240     });
241
242     // Hide dropdown with delay
243     dropdown.addEventListener('mouseleave', () => {
244       setTimeout(() => {
245         if (!content.matches(':hover')) {
246           content.classList.remove('visible');
247           activeDropdown = null;
248         }
249       }, 10);
250     });
251   });
252
253
254
255
256
257

```

```
246     content.classList.add('visible');
247     activeDropdown = content;
248   });
249
250   // Hide dropdown with delay
251   dropdown.addEventListener('mouseleave', () => {
252     setTimeout(() => {
253       if (!content.matches(':hover')) {
254         content.classList.remove('visible');
255         activeDropdown = null;
256       }
257     }, 10);
258   });
259
260   // Keep dropdown open when hovering its content
261   content.addEventListener('mouseenter', () => {
262     clearTimeout();
263     content.classList.add('visible');
264   });
265
266   // Hide dropdown when leaving its content
267   content.addEventListener('mouseleave', () => {
268     content.classList.remove('visible');
269     activeDropdown = null;
270   });
271 });
272 });
273 </script>
274 </body>
275 </html>
```

*AmendView.php*

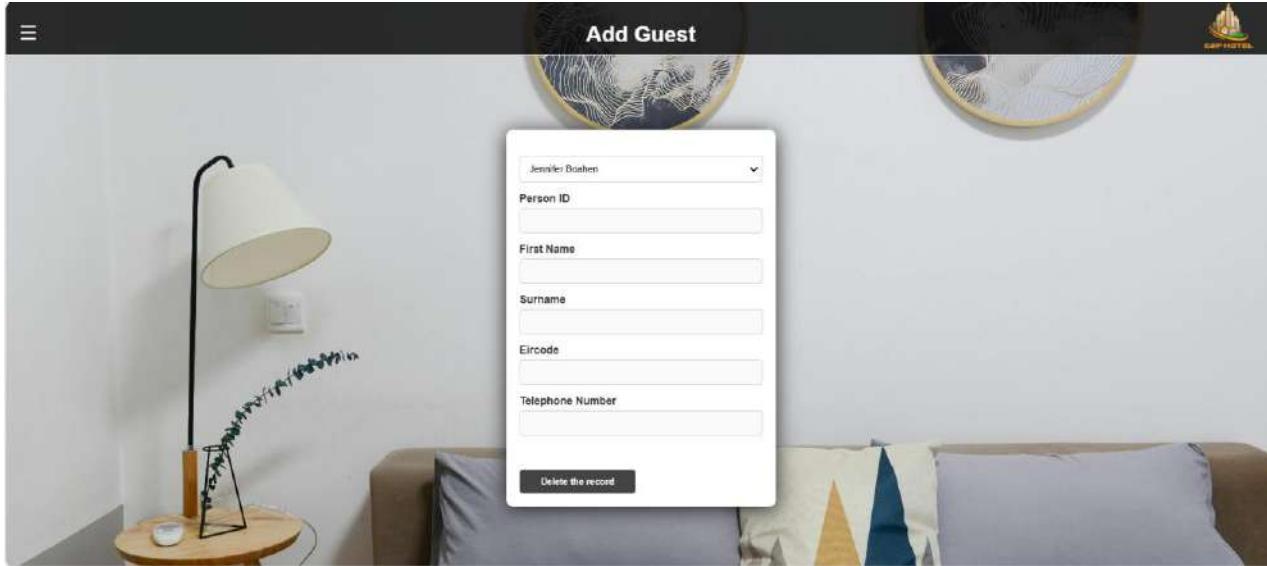
```
1| <?php
2| // Hamed Zon
3| // c00259228
4| //March 2025
5| // This is the PHP file to update and view guest details and it direct back user to the form page
6|
7|
8| // Start a session to manage session variables
9| session_start();
10|
11| // Include the database connection file to connect to the database
12| include 'db.inc.php';
13|
14| // Check if the form was submitted using the POST method and if the 'id' field is set
15| if ($_SERVER['REQUEST_METHOD'] === 'POST' && isset($_POST['amendid'])) {
16|     // Sanitize the 'id' input to prevent SQL injection
17|     $id = mysqli_real_escape_string($con, $_POST['amendid']);
18|
19|     // Sanitize the 'FirstName' input to prevent SQL injection
20|     $firstName = mysqli_real_escape_string($con, $_POST['amendfname']);
21|
22|     // Sanitize the 'Surname' input to prevent SQL injection
23|     $surname = mysqli_real_escape_string($con, $_POST['amendsname']);
24|
25|     // Sanitize the 'Address' input to prevent SQL injection
26|     $address = mysqli_real_escape_string($con, $_POST['amendaddress']);
27|
28|     // Sanitize the 'Eircode' input to prevent SQL injection
29|     $eircode = mysqli_real_escape_string($con, $_POST['amendeircode']);
30| }
```

```

31 // Sanitize the 'Phone' input to prevent SQL injection
32 $phone = mysqli_real_escape_string($con, $_POST['amendphone']);
33
34 // Sanitize the 'Email' input to prevent SQL injection
35 $email = mysqli_real_escape_string($con, $_POST['amendemail']);
36
37 // Store the current timestamp for the last update
38 $lastupdate = date('Y-m-d H:i:s');
39
40 // SQL query to update the guest details
41 $sql = "UPDATE Guests
42     SET FirstName = '$firstName',
43         Surname = '$surname',
44         Address = '$address',
45         Eircode = '$eircode',
46         TelephoneNumber = '$phone',
47         EmailAddress = '$email',
48         DateLastUpdate = '$lastupdate'
49     WHERE GuestID = '$id';";
50
51 // Execute the SQL query
52 if (mysqli_query($con, $sql)) {
53     // Check if any rows were affected by the query (i.e., if the guest details were updated)
54     if (mysqli_affected_rows($con) > 0) {
55         // Set a success message in the session variable
56         $_SESSION['update_message'] = "Guest details updated successfully: $firstName $surname!";
57     } else {
58         // Set a message indicating no changes were made
59         $_SESSION['update_message'] = "No changes were made to the guest details.";
60     }
61     if (mysqli_affected_rows($con) > 0) {
62         // Set a success message in the session variable
63         $_SESSION['update_message'] = "Guest details updated successfully: $firstName $surname!";
64     } else {
65         // Set a message indicating no changes were made
66         $_SESSION['update_message'] = "No changes were made to the guest details.";
67     }
68 } else {
69     // If there was an error with the query, set an error message in the session variable
70     $_SESSION['update_message'] = "Error updating guest: " . mysqli_error($con);
71 }
72
73 // Close the database connection
74 mysqli_close($con);
75
76 // Redirect the user back to the view guest page
77 header("Location: ViewGuest.html.php");
78 // Stop further execution of the script
79 exit();
80 }
81 }
82 ?>
83

```

## Delete Guest Screen



```

1 <?php
2 //Hamed Zon
3 //C00259228
4 //This code allows you to delete a guest by selecting from dropdown list
5 //March 2025
6 // Start a PHP session to manage session variables
7 session_start();
8 ?>
9 <html lang="en"> <!-- Define the document as HTML with English language -->
10 <head>
11   <meta charset="UTF-8">
12   <title>Delete Guest</title>
13   <link rel="stylesheet" href="../../styles.css">
14   <link rel="webiste icon" value="png" href="../../C2PLogoNew.png">
15 </head>
16 <body>
17   <!-- Burger icon that will toggle the sidebar -->
18   <div class="burger-menu">&#9776;</div>
19   <!-- Page header with a title and logo -->
20   <div class="page-header">
21     <h1>Add Guest</h1>
22     <!-- Link to the home page with the hotel logo -->
23     <a href="../../home.html.php"></a>
24   </div>
25
26   <!-- Sidebar navigation -->
27   <?php include '../../sidebar.php'; ?>
28
29   <!-- JavaScript section for handling guest selection and form submission -->
30   <script>
```

```

31 // Function to populate form fields when a guest is selected
32 function populate() {
33     // Get the selected guest from the listbox
34     var sel = document.getElementById("listbox");
35     // Get the value of the selected option
36     var result = sel.options[sel.selectedIndex].value;
37     // Split the selected value into an array of details (comma-separated)
38     var personDetails = result.split(',');
39
40     // Populate the form fields with the guest's details
41     document.getElementById("delid").value = personDetails[0]; // Person ID
42     document.getElementById("FirstName").value = personDetails[1]; // First Name
43     document.getElementById("Surname").value = personDetails[2]; // Surname
44     document.getElementById("Eircode").value = personDetails[4]; // Eircode
45     document.getElementById("PhoneNumber").value = personDetails[5]; // Telephone Number
46
47 }
48
49 // Function to confirm the deletion action
50 function confirmCheck() {
51     // Ask the user for confirmation before deleting
52     if (confirm('Are you sure you want to delete this person?')) {
53         // Enable the disabled fields so their values are submitted with the form
54         document.getElementById("delid").disabled = false;
55         document.getElementById("FirstName").disabled = false;
56         document.getElementById("Surname").disabled = false;
57         document.getElementById("Eircode").disabled = false;
58         document.getElementById("PhoneNumber").disabled = false;
59
60         // Allow the form to be submitted
61         return true;
62     }
63     // If the user cancels, prevent the form from being submitted
64     return false;
65 }
66 </script>
67
68 <div class="content add-page">
69     <?php
70     // Check if a session message is set
71     if (isset($_SESSION['delete_message'])) {
72         // Display the session message inside an <h1> element
73         echo "<h1 class='success'>" . htmlspecialchars($_SESSION['delete_message']) . "</h1>";
74         // Clear the session message after displaying it
75         unset($_SESSION['delete_message']);
76     }
77 ?>
78
79 <!-- Form for deleting a guest record -->
80 <form name="deleteForm" action="DeleteGuest2.php" onsubmit="return confirmCheck()" method="POST">
81
82
83     <!-- Container for the guest list box -->
84     <div class="inputbox">
85         <!-- Include the guest list box (likely a dropdown or list of guests) -->
86         <?php include 'guestListBox.php'; ?>
87     </div>

```

```

86         <:php include "gueslistbox.php" ;>
87     </div>
88
89     <!-- Input field for Person ID disabled by default -->
90     <div class="inputbox">
91         <label for="delid">Person ID</label>
92         <input type="text" name="delid" id="delid" disabled>
93     </div>
94
95     <!-- Input field for First Name disabled by default -->
96     <div class="inputbox">
97         <label for="FirstName">First Name</label>
98         <input type="text" name="FirstName" id="FirstName" disabled>
99     </div>
100
101    <!-- Input field for Surname disabled by default -->
102    <div class="inputbox">
103        <label for="Surname">Surname</label>
104        <input type="text" name="Surname" id="Surname" disabled>
105    </div>
106
107
108    <!-- Input field for Eircode disabled by default-->
109    <div class="inputbox">
110        <label for="Eircode">Eircode</label>
111        <input type="text" name="Eircode" id="Eircode" disabled>
112    </div>
113
114    <!-- Input field for Telephone Number disabled by default -->
115    <div class="inputbox">
116        <label for="TelephoneNumber">Telephone Number</label>
117        <input type="text" name="TelephoneNumber" id="TelephoneNumber" disabled>
118    </div>
119
120
121    <!-- Submit button for the form -->
122    <br><br>
123    <div class="Mybutton">
124        <input type="submit" value="Delete the record">
125    </div>
126    </form>
127 </div>
128
129
130 <script>
131     document.addEventListener('DOMContentLoaded', function () {
132         // ===== SIDEBAR =====
133         const burger = document.querySelector('.burger-menu');
134         const sidebar = document.querySelector('.sidebar');
135
136         // Toggle sidebar on burger click
137         burger.addEventListener('click', (e) => {
138             e.stopPropagation();
139             sidebar.classList.toggle('open');
140         });
141
142         // Close sidebar when clicking outside

```

```

142 // Close sidebar when clicking outside
143 document.addEventListener('click', (e) => {
144   if (!sidebar.contains(e.target) && !e.target.classList.contains('burger-menu')) {
145     sidebar.classList.remove('open');
146   }
147 });
148
149 // ===== DROPPDOWNS =====
150 const dropdowns = document.querySelectorAll('.dropdown');
151 let activeDropdown = null;
152
153 dropdowns.forEach(dropdown => {
154   const content = dropdown.querySelector('.dropdown-content');
155
156   // Show dropdown on hover
157   dropdown.addEventListener('mouseenter', () => {
158     if (activeDropdown) {
159       activeDropdown.classList.remove('visible');
160     }
161     content.classList.add('visible');
162     activeDropdown = content;
163   });
164
165   // Hide dropdown with delay
166   dropdown.addEventListener('mouseleave', () => {
167     setTimeout(() => {
168       if (!content.matches(':hover')) {
169         content.classList.remove('visible');
170         activeDropdown = null;
171       }
172     }, 10);
173   });
174
175   // Keep dropdown open when hovering its content
176   content.addEventListener('mouseenter', () => {
177     clearTimeout();
178     content.classList.add('visible');
179   });
180
181   // Hide dropdown when leaving its content
182   content.addEventListener('mouseleave', () => {
183     content.classList.remove('visible');
184     activeDropdown = null;
185   });
186 });
187 });
188 </script>
189
190 </body>
191 </html>

```

*DeleteGuest2.php*

```

<?php
//Hamed Zon
//c00259228
//March 2025
//This is the php to set the guest to be deleted by settintg it to 1 and send user back to main page

// Start a session to manage session variables
session_start();

// Include the database connection file to connect to the database
include 'db.inc.php';

// Check if the form was submitted using the POST method and if the delid field is set
if ($_SERVER['REQUEST_METHOD'] === 'POST' && isset($_POST['delid'])) {
    // Sanitize the delid input to prevent SQL injection
    $delid = mysqli_real_escape_string($con, $_POST['delid']);

    // Sanitize the FirstName input to prevent SQL injection
    $firstName = mysqli_real_escape_string($con, $_POST['FirstName']);

    // Sanitize the Surname input to prevent SQL injection
    $surname = mysqli_real_escape_string($con, $_POST['Surname']);

    // SQL query to mark the guest as deleted by setting the DeletedFlag to 1
    $sql = "UPDATE Guests SET DeletedFlag = 1 WHERE GuestID = '$delid'";

    // Execute the SQL query
    if (mysqli_query($con, $sql)) {
        // Check if any rows were affected by the query
        if (mysqli_affected_rows($con) > 0) {
            // Check if any rows were affected by the query
            if (mysqli_affected_rows($con) > 0) {
                // Set a success message in the session variable
                $_SESSION['delete_message'] = "Guest deleted successfully: $firstName $surname!";
            } else {
                // Set a message indicating no record was found to delete
                $_SESSION['delete_message'] = "No record found to delete.";
            }
        } else {
            // If there was an error with the query, set an error message in the session variable
            $_SESSION['delete_message'] = "Error: " . mysqli_error($con);
        }
    }

    // Close the database connection
    mysqli_close($con);

    // Redirect the user back to the delete guest page
    header("Location: deleteGuest.html.php");
    // Stop further execution of the script
    exit();
} else {
    // If the form was not submitted or no guest was selected, set a message indicating no guest was selected
    $_SESSION['delete_message'] = "No guest selected for deletion.";

    // Redirect the user back to the delete guest page
    header("Location: deleteGuest.html.php");
    // Stop further execution of the script
    exit();
}
?>

```

*GuestListBox2.php*

```

<?php
//Hamed Zon
//C00259228
//March 2025
//this code allows you to see list of names to select from , and view their data if the deletedflag is 0
    include "db.inc.php"; //connecting db
    date_default_timezone_set('UTC');
//Query to get all data where deletedflag is 0
    $sql = "SELECT GuestID, FirstName, Surname, Address, Eircode, TelephoneNumber, EmailAddress, DateLastUpdate FROM Guests WHERE DeletedFlag=0";
    if (!$result = mysqli_query($con,$sql)) //if not
    {
        die('Error in querying the database'. mysqli_error($con)); //Print message
    }

    echo "<br><select name='listbox' id='listbox' onclick= 'populate()'>";//get list box
    while($row = mysqli_fetch_array($result)) //getting all data inside the row
    {
        //getting all data from db and storing them in the new variables
        $id = $row['GuestID'];
        $fname = $row['FirstName'];
        $sname = $row['Surname'];
        $add = $row['Address'];
        $eircode = $row['Eircode'];
        $phone = $row['TelephoneNumber'];
        $email = $row['EmailAddress'];
        $update = $row['DateLastUpdate'];

        $allText = "$id,$fname,$sname,$add,$eircode,$phone,$email,$update";// sending all data in variable allText
        echo "<option value= '$allText'> $fname $sname </option>";//Show only first and last name in the list box
    }

    echo "</select>";
    mysqli_close($con); //close connection
?>

```

## Addandextra.php

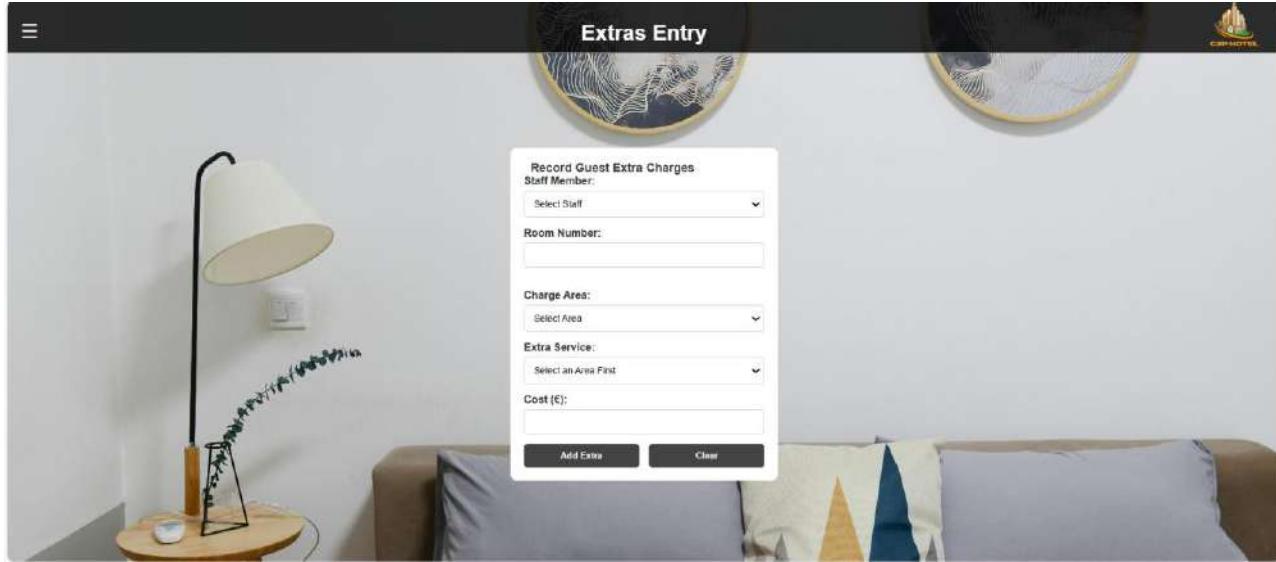
```
1 |?php
2 // Start a session to manage session variables
3 session_start();
4
5 //this php code allows you to add extra,get area details and show things that are in extra category and it send user back to the form after
6 //adding extra
7 //Hamed Zon
8 //c00259228
9 //March 2025
10
11 // Include the database connection file
12 include '../db.inc.php';
13
14 // see if the form is submitted using the POST method
15 if ($_SERVER["REQUEST_METHOD"] == "POST") {
16     // get form data using the null, operator ?? to provide default null values if fields are empty
17     $staffID = $_POST['staff'] ?? null; // Staff ID from the form
18     $roomID = $_POST['room'] ?? null; // room ID from the form
19     $areaID = $_POST['area'] ?? null; // Area ID from the form
20     $extraID = $_POST['extra'] ?? null; // extra ID from the form
21     $cost = $_POST['cost'] ?? null; // cost from the form
22     $dateTime = date('Y-m-d H:i:s'); // Current date and time in 'YYYY-MM-DD HH:MM:SS' format
23
24     // Debugging: Log all incoming data to the error log for troubleshooting
25     error_log("Received Data -> StaffID: $staffID, RoomID: $roomID, AreaID: $areaID, ExtraID: $extraID, Cost: $cost");
26
27     // Validate that all required fields are filled and cost is a positive number
28     if (!$staffID || !$roomID || !$areaID || !$extraID || $cost <= 0) {
29         // Set an error message in the session if validation fails
30         $_SESSION['error'] = "All fields are required and cost must be positive.";
31
32         // Redirect the user back to the extras form page
33         header("Location: extras.html.php");
34         exit(); // Stop further execution
35     }
36
37     // Fetch the Area Name from the Areas table using the AreaID
38     $areaQuery = "SELECT AreaName FROM Areas WHERE AreaID = ?";
39     // Prepare the SQL statement to prevent SQL injection
40     $stmt = $con->prepare($areaQuery);
41     // Bind the AreaID parameter to the SQL statement
42     $stmt->bind_param("i", $areaID);
43     // Execute the SQL statement
44     $stmt->execute();
45     // Get the result of the query
46     $areaResult = $stmt->get_result();
47     // Fetch the result as an associative array
48     $areaRow = $areaResult->fetch_assoc();
49     // Get the AreaName from the result, default to "Unknown" if not found
50     $areaName = $areaRow['AreaName'] ?? "Unknown";
51
52     // Fetch the Extra Description from the ExtrasCategory table using the ExtraID
53     $extraQuery = "SELECT ExtraName FROM ExtrasCategory WHERE ExtraID = ?";
54     // Prepare the SQL statement to prevent SQL injection
55     $stmt = $con->prepare($extraQuery);
56     // Bind the ExtraID parameter to the SQL statement
57     $stmt->bind_param("i", $extraID);
58     // Execute the SQL statement
```

```

60 // Fetch the result as an associative array
61 $extraRow = $extraResult->fetch_assoc();
62 // Get the ExtraName from the result, default to "Unknown" if not found
63 $extraDescription = $extraRow['ExtraName'] ?? "Unknown";
64
65 // Debugging: Log the fetched AreaName and ExtraDescription for troubleshooting
66 error_log("Fetched Data -> AreaName: $areaName, ExtraDescription: $extraDescription");
67
68 // Insert the new extra charge into the Extras table
69 $stmt = $con->prepare("INSERT INTO Extras (AreaID, ExtraID, Area, Description, Cost, RoomID, StaffID, DateTime) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?)");
70 // Bind all parameters to the SQL statement
71 $stmt->bind_param("iissdiis", $areaID, $extraID, $areaName, $extraDescription, $cost, $roomID, $staffID, $dateTime);
72
73 // Execute the SQL statement and check if it was successful
74 if ($stmt->execute()) {
    // Set a success message in the session if the insert was successful
    $_SESSION['success'] = "Extra charge added successfully!";
} else {
    // Set an error message in the session if the insert failed
    $_SESSION['error'] = "Error adding extra charge: " . $stmt->error;
}
80
81 // Close the prepared statement
82 $stmt->close();
83 // Close the database connection
84 $con->close();
85
86
87 // Redirect the user back to the extras form page
88 header("Location: extras.html.php");
89 exit(); // Stop further execution
90 }
91 ?>

```

## Add Extras Screen



```

1 <?php
2 //Hamed Zon
3 //March 2025
4 //this is a html and php code the form allows you to select staff and more options from dropdown lists and allows you to add extra for a
5 guest.
6 // Start a session to manage session variables
7
8 session_start();
9
10 // Include the database connection file to connect to the database
11 include '../db.inc.php';
12
13 //Hamed Zon
14 //C00259228
15 //March 2025
16 //This HTML and PHP code allows user to input extra details, room id, to see staff first and last name,guest details, areas and category
17 list to select from so that extra can be added to the db.
18
19 // Fetch staff list from the database
20 $staffQuery = "SELECT StaffID, CONCAT(FirstName, ' ', Surname) AS FullName FROM Staff WHERE DeletedFlag = 0";
21 $staffResult = $con->query($staffQuery);
22
23 // get available areas from the database
24 $areaQuery = "SELECT AreaID, AreaName FROM Areas";
25 $areaResult = $con->query($areaQuery);
26
27 <!DOCTYPE html>
28 <html lang="en">
<head>
```

```

28 <head>
29
30     <meta charset="UTF-8">
31     <!-- Set the title of the webpage -->
32     <title>Extras Entry</title>
33     <!-- Link to the external CSS file for styling -->
34     <link rel="stylesheet" href="../styles.css">
35     <!-- Set the website icon (favicon) -->
36     <link rel="website icon" type="png" href="../C2PLogoNew.png">
37 </head>
38 <body>
39     <!-- Burger icon for toggling the sidebar on smaller screens -->
40     <div class="burger-menu">#9776;</div>
41
42     <!-- Page header with a title and logo -->
43     <div class="page-header">
44         <h1>Extras Entry</h1>
45         <!-- Link to the home page with the hotel logo -->
46         <a href="../home.html.php"></a>
47     </div>
48
49     <!-- Include the sidebar navigation -->
50     <?php include '../sidebar.php'; ?>
51
52     <!-- Main content area -->
53     <div class="content">
54         <!-- Form for recording guest extra charges -->
55         <form action="addExtras.php" method="POST">
56             <form action="addExtras.php" method="POST">
57                 <!-- Form legend (title) -->
58                 <legend>Record Guest Extra Charges</legend>
59
60                 <!-- Staff Selection Dropdown list-->
61                 <div class="inputbox">
62                     <label for="staff">Staff Member:</label>
63                     <select name="staff" id="staff" required>
64                         <!-- Default option for the dropdown list -->
65                         <option value="">Select Staff</option>
66                         <!-- go through the staff results and populate the dropdown -->
67                         <?php while ($staff = $staffResult->fetch_assoc()): ?>
68                             <!-- Display the staff's full name as an option -->
69                             <option value=<?= $staff['StaffID'] ?><?= htmlspecialchars($staff['FullName']) ?></option>
70                         <?php endwhile; ?>
71                     </select>
72                 </div>
73
74                 <!-- Room Number Input -->
75                 <div class="inputbox">
76                     <label for="room">Room Number:</label>
77                     <!-- Input field for entering the room number -->
78                     <input type="number" name="room" id="room" required onchange="fetchGuestDetails()">
79                 </div>
80
81                 <!-- Guest Details Message -->
82                 <div id="guestDetails" style="color: black; font-weight: bold;"></div><br>
83
84                 <!-- Area Selection Dropdown -->
85                 <div class="inputbox">
86                     <label for="area">Area:</label>
87                     <select name="area" id="area" required>
88                         <!-- Default option for the dropdown list -->
89                         <option value="">Select Area</option>
90                         <!-- go through the area results and populate the dropdown -->
91                         <?php while ($area = $areaResult->fetch_assoc()): ?>
92                             <!-- Display the area's name as an option -->
93                             <option value=<?= $area['AreaID'] ?><?= htmlspecialchars($area['Name']) ?></option>
94                         <?php endwhile; ?>
95                     </select>
96                 </div>
97             </form>
98         </form>
99     </div>
100
101     <!-- Footer -->
102     <div class="page-footer">
103         
104         <p>C2P Hotel Project Manual</p>
105     </div>
106
107 </body>
108
109 </html>

```

```

84     <div class="inputbox">
85         <label for="area">Charge Area:</label>
86         <select name="area" id="area" required onchange="fetchExtras()">
87             <!-- Default option for the dropdown -->
88             <option value="">Select Area</option>
89             <!-- go through the area results and populate the dropdown list-->
90             <?php while ($area = $areaResult->fetch_assoc()): ?>
91                 <!-- Display the area name as an option -->
92                 <option value=<?= $area['AreaID'] ?><?= htmlspecialchars($area['AreaName']) ?>></option>
93             <?php endwhile; ?>
94         </select>
95     </div>
96
97     <!-- Extras Selection Dropdown -->
98     <div class="inputbox">
99         <label for="extra">Extra Service:</label>
100        <select name="extra" id="extra" required>
101            <!-- Default option for the dropdown -->
102            <option value="">Select an Area First</option>
103        </select>
104    </div>
105
106    <!-- Cost Input -->
107    <div class="inputbox">
108        <label for="cost">Cost (€):</label>
109        <!-- Input field for entering the cost of the extra service -->
110        <input type="number" name="cost" id="cost" step="0.01" min="0.01" required>
111    </div>
112
113    <!-- Submit and Clear Buttons -->
114    <div class="myButton">
115        <!-- Submit button to submit the form -->
116        <input type="submit" value="Add Extra">
117        <!-- Reset button to clear the form -->
118        <input type="reset" value="Clear">
119    </div>
120 </form>
121 </div>
122
123 <!-- JavaScript for dynamic functionality -->
124 <script>
125     // Function to fetch extras based on the selected area
126     function fetchExtras() {
127         // Get the selected area ID from the dropdown
128         let areaID = document.getElementById("area").value;
129         // Get the extras dropdown element
130         let extraDropdown = document.getElementById("extra");
131
132         // Check if an area is selected
133         if (!areaID) {
134             console.error("Error: No Area ID Selected!");
135             return;
136         }
137
138         // Log the selected area ID for debugging
139         console.log("Fetching extras for AreaID:", areaID);
140

```

```

141 // Fetch extras from the server using the selected area ID
142 fetch("getExtras.php?areaID=" + areaID)
143     .then(response => response.json()) // Parse the response as JSON
144     .then(data => {
145         // Log the received extras for debugging
146         console.log("Received Extras:", data);
147         // Clear the extras dropdown and add a default option
148         extraDropdown.innerHTML = "<option value='>Select Extra</option>";
149         // Loop through the fetched extras and add them to the dropdown
150         data.forEach(extra => {
151             extraDropdown.innerHTML += `<option value="${extra.id}">${extra.name}</option>`;
152         });
153     })
154     .catch(error => console.error("Error fetching extras:", error)); // Log any errors
155 }
156
157 // Function to fetch guest details based on the selected room number
158 function fetchGuestDetails() {
159     // Get the entered room number
160     let roomID = document.getElementById("room").value;
161     // Get the guest details div element
162     let guestDiv = document.getElementById("guestDetails");
163
164     // Fetch guest details from the server using the selected room number
165     fetch("getGuest.php?roomID=" + roomID)
166         .then(response => response.text()) // Parse the response as text
167         .then(data => {
168             // Display the guest details in the designated div
169             guestDiv.innerHTML = `

${data}

`;
170         })
171         .catch(error => console.error("Error fetching guest details:", error)); // Log any errors
172 }
173
174 // Event listener for when the DOM is fully loaded
175 document.addEventListener('DOMContentLoaded', function () {
176     // Get the burger menu and sidebar elements
177     const burger = document.querySelector('.burger-menu');
178     const sidebar = document.querySelector('.sidebar');
179
180     // Toggle sidebar on burger click
181     burger.addEventListener('click', (e) => {
182         e.stopPropagation(); // Prevent the event from bubbling up
183         sidebar.classList.toggle('open'); // Toggle the 'open' class on the sidebar
184     });
185
186     // Close sidebar when clicking outside
187     document.addEventListener('click', (e) => {
188         if (!sidebar.contains(e.target) && !e.target.classList.contains('burger-menu')) {
189             sidebar.classList.remove('open'); // Remove the 'open' class from the sidebar
190         }
191     });
192
193     // Sidebar dropdowns functionality
194     const dropdowns = document.querySelectorAll('.dropdown');
195     let activeDropdown = null;
196

```

```
197 // Loop through all dropdowns
198 dropdowns.forEach(dropdown => {
199   const content = dropdown.querySelector('.dropdown-content');
200
201   // Show dropdown on hover
202   dropdown.addEventListener('mouseenter', () => {
203     if (activeDropdown) {
204       activeDropdown.classList.remove('visible'); // Hide the currently active dropdown
205     }
206     content.classList.add('visible'); // Show the current dropdown
207     activeDropdown = content; // Set the current dropdown as active
208   });
209
210   // Hide dropdown with delay
211   dropdown.addEventListener('mouseleave', () => {
212     setTimeout(() => {
213       if (!content.matches(':hover')) {
214         content.classList.remove('visible'); // Hide the dropdown if not hovered
215         activeDropdown = null; // Reset the active dropdown
216       }
217     }, 10); // Delay of 10ms
218   });
219 });
220 });
221 </script>
222 </body>
223 </html>
```

*GetExtra.php*

```

1 <?php
2 // Include the database connection file to connect to the database
3 include '../db.inc.php';
4
5 //Hamed Zon|
6 //C00259228
7 //March 2025
8 //This PHP code get details from the ExtrasCategory table based on the provided AreaID
9
10 // Check if the areaID parameter is set in the GET request
11 if (isset($_GET['areaID'])) {
12     // get the areaID value from the GET request
13     $areaID = $_GET['areaID'];
14
15     // Log the received AreaID for debugging purposes
16     error_log("Fetching extras for AreaID: " . $areaID);
17
18     // Prepare a SQL query to fetch extras for the given AreaID
19     $query = "SELECT ExtraID, ExtraName FROM ExtrasCategory WHERE AreaID = ?";
20     // Prepare the SQL statement to prevent SQL injection
21     $stmt = $con->prepare($query);
22     // Bind the AreaID parameter to the SQL statement
23     $stmt->bind_param("i", $areaID);
24     // Execute the SQL statement
25     $stmt->execute();
26     // Get the result of the query
27     $result = $stmt->get_result();
28
29     // Initialize an empty array to store the fetched extras
30     $extras = [];
31     // Loop through the result set and populate the extras array
32     while ($row = $result->fetch_assoc()) {
33         $extras[] = ["id" => $row['ExtraID'], "name" => $row['ExtraName']];
34     }
35
36     // Log the fetched extras for debugging purposes
37     error_log("Fetched Extras: " . json_encode($extras));
38
39     // Return the fetched extras as a JSON response
40     echo json_encode($extras);
41 }
42 ?>

```

*Process.php*

```

1 <?php
2 // Start a session to manage session variables
3 session_start();
4
5 // Include the database connection file to connect to the database
6 include '../db.inc.php';
7
8 // Developer information
9 //Hamed Zon
10 //c00259228
11 //March 2025
12 //This PHP code allows you to get stock details based on the provided ID.
13
14 // Check if the form is submitted using the POST method
15 if ($_SERVER["REQUEST_METHOD"] == "POST") {
16     // Check if the 'stock' parameter is set in the POST request
17     if (isset($_POST['stock'])) {
18         // Retrieve the stock value from the POST request
19         $stockID = $_POST['stock'];
20
21         // Prepare a SQL query to fetch details of the selected stock item
22         $stock_sql = "SELECT * FROM Stock WHERE StockID = ?";
23         // Prepare the SQL statement to prevent SQL injection
24         $stmt = $con->prepare($stock_sql);
25         // Bind the stockID parameter to the SQL statement
26         $stmt->bind_param("i", $stockID);
27         // Execute the SQL statement
28         $stmt->execute();
29         // Get the result of the query
30         $stock_result = $stmt->get_result();
31
32         // Check if the stock item exists
33         if ($stock_result->num_rows > 0) {
34             // Fetch the stock item details as an associative array
35             $stock_data = $stock_result->fetch_assoc();
36             // Store the stock item details in the session for later use
37             $_SESSION['stock_data'] = $stock_data;
38             // Set a success message in the session
39             $_SESSION['success'] = "Stock item selected successfully.";
40         } else {
41             // Set an error message in the session if the stock item is not found
42             $_SESSION['error'] = "Stock item not found.";
43         }
44
45         // Close the prepared statement
46         $stmt->close();
47     } else {
48         // Set an error message in the session if no stock item is selected
49         $_SESSION['error'] = "No stock item selected.";
50     }
51 }
52
53 // Redirect the user back to the previous page (getStock.php)
54 header("Location: getStock.php");
55 // Stop further execution
56 exit();
57 ?>
```

*getGuest.php*

```

1 <?php
2 // Include the database connection file to connect to the database
3 include '../db.inc.php';
4
5 //Hamed Zon
6 //C00259228
7 //March 2025
8 //This code fetches and displays the name of the guest staying in the selected roomid.
9
10 // Check if the roomID parameter is set in the GET request
11 if (isset($_GET['roomID'])) {
12     // get the roomID value from the GET request
13     $roomID = $_GET['roomID'];
14
15     // query to fetch the guest's name for the given roomID
16     $query = "SELECT CONCAT(FirstName, ' ', Surname) AS GuestName FROM Guests
17             JOIN GuestStays ON Guests.GuestID = GuestStays.GuestID
18             WHERE GuestStays.RoomID = ? LIMIT 1";
19
20     // Prepare the SQL statement to prevent SQL injection
21     $stmt = $con->prepare($query);
22     // Bind the roomID parameter to the SQL statement
23     $stmt->bind_param("i", $roomID);
24     // Execute the SQL statement
25     $stmt->execute();
26     // Get the result of the query
27     $result = $stmt->get_result();
28
29     // Check if a guest is found for the given roomID
30     if ($row = $result->fetch_assoc()) {
31         // Display the guest's name in a paragraph element
32         echo "<p>Guest: " . htmlspecialchars($row['GuestName']) . "</p>";
33     } else {
34         // Display a message if no guest is found for the room
35         echo "<p>No guest found for this room.</p>";
36     }
37 }
38 ?>

```

*RoomListBox.php*

```

1 <?php
2 // Include the database connection file
3 //Hamed Zon
4 //C00259228
5 //March 2025
6 //this is a php code that get datas from guest, rooms and extras table based on the id provided
7 include '../db.inc.php';
8
9 // Start a session to manage session variables
10 session_start();
11
12 // Check if staff ID is submitted
13 if (isset($_POST['staff'])) {
14     $staffID = $_POST['staff'];
15     $_SESSION['staffID'] = $staffID; // Store staff ID in session
16 }
17
18 // Fetch room data for the dropdown list
19 $room_sql = "SELECT RoomID FROM Rooms WHERE DeletedFlag = 0";
20 $room_result = mysqli_query($con, $room_sql);
21
22 if (!$room_result) {
23     die("Error fetching rooms: " . mysqli_error($con));
24 }
25
26 // Check if a room ID is selected
27 if (isset($_POST['room'])) {
28     $roomID = $_POST['room'];
29
30     // Fetch extras data for the selected room ID
31
32     // Fetch extras data for the selected room ID
33     $extras_sql = "SELECT * FROM Extras WHERE RoomID = '$roomID'";
34     $extras_result = mysqli_query($con, $extras_sql);
35
36     if (!$extras_result) {
37         die("Error fetching extras: " . mysqli_error($con));
38     }
39
40     // Fetch guest details for the selected room ID
41     $guest_sql = "SELECT * FROM Guests Join Rooms WHERE RoomID = '$roomID'";
42     $guest_result = mysqli_query($con, $guest_sql);
43
44     if (!$guest_result) {
45         die("Error fetching guest details: " . mysqli_error($con));
46     }
47
48     $guest_data = mysqli_fetch_assoc($guest_result);
49 }
50 ?>

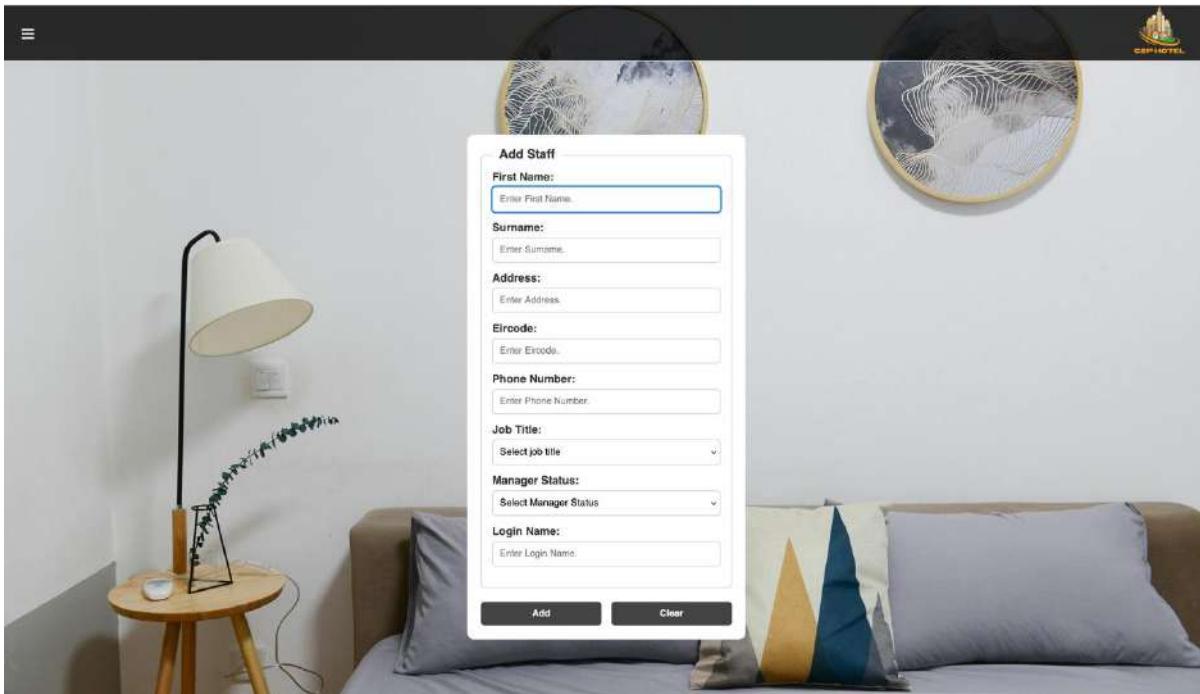
```

*staffListBox.php*

```
1 <?php
2 // Include the database connection file
3 //Hamed Zon
4 //C00259228
5 //March 2025
6 //this code get staff id firstname and lastname if they deletedflag is 0
7 include '../db.inc.php';
8
9 // Fetch staff data for the dropdown list
10 $staff_sql = "SELECT StaffID, FirstName, Surname FROM Staff WHERE DeletedFlag = 0";
11 $staff_result = mysqli_query($con, $staff_sql); //moving them to staffresult variable
12
13 if (!$staff_result) { //if not
14     die("Error fetching staff: " . mysqli_error($con)); //print this
15 }
16 ?>
```

Mantas Bielevicius Written By: Mantas Bielevicius

## Add Staff Screen



addStaff.html.php

<!DOCTYPE html>

<!--

Name Of Screen: addStaff.html.php

Purpose Of Screen: To add a new staff member. The form data is processed in addStaff.php, where the database is updated.

Student ID: C00294564

Name: Mantas Bielevicius

Date Written: March 2025

-->

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Add Staff</title>

<!-- Link to the external stylesheet -->

<link rel="stylesheet" href="../../styles.css">

<link rel="webiste icon" value="png" href="../../C2PLogoNew.png">

</head>

<body>

<!-- Burger icon for toggling the sidebar -->

<div class="burger-menu">&#9776;</div>

<!-- Page header containing logo and link to home page -->

<div class="page-header">

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



</a>

</div>

<!-- Include sidebar navigation from external PHP file -->

<?php include '../../sidebar.php'; ?>

<!-- Main content area for Add Staff form -->

```

<div class="content">
    <!-- Staff addition form. The form submits to "addStaff.php" where the database is updated. -->
    <form id="addStaffForm" action="addStaff.php" onsubmit="return confirm('Are you sure you want to add staff?');" method="POST">
        <fieldset>
            <!-- Fieldset legend for accessibility -->
            <legend>Add Staff</legend>

            <!-- Input for First Name -->
            <div class="inputbox">
                <label for="firstName">First Name:</label>
                <!-- Pattern restricts input to letters only. "autofocus" puts the cursor in this field on page load. -->
                <input type="text" pattern="[A-Za-z]+" name="firstName" id="firstName" placeholder="Enter First Name." maxlength="50" autofocus required/>
            </div>

            <!-- Input for Surname -->
            <div class="inputbox">
                <label for="surname">Surname:</label>
                <input type="text" pattern="[A-Za-z]+" name="surname" id="surname" placeholder="Enter Surname." maxlength="50" required/>
            </div>

            <!-- Input for Address -->
            <div class="inputbox">
                <label for="address">Address:</label>
                <input type="text" pattern="^[A-Za-z0-9\s.,-]+$" name="address" id="address" placeholder="Enter Address." maxlength="255" required/>
            </div>

            <!-- Input for Eircode -->
            <div class="inputbox">
                <label for="eircode">Eircode:</label>
                <input type="text" pattern="^([A-Za-z]\d{2}\s?[A-Za-z0-9]{4})$" name="eircode" id="eircode" placeholder="Enter Eircode." required/>
            </div>

            <!-- Input for Phone Number -->
            <div class="inputbox">
                <label for="phoneNumber">Phone Number:</label>
                <input type="text" pattern="[0-9 ]*" name="phoneNumber" id="phoneNumber" placeholder="Enter Phone Number." maxlength="15" required/>
            </div>

            <!-- Select field for Job Title (ordered alphabetically) -->
            <div class="inputbox">
                <label for="jobTitle">Job Title:</label>
                <select id="jobTitle" name="jobTitle">
                    <!--
                        Default option serves as a placeholder.
                        It is disabled and selected so the user must choose another option.
                    -->
                    <option value="" disabled selected>Select job title</option>
                    <!-- Job Title options ordered alphabetically -->
                    <option value="Assistant Manager">Assistant Manager</option>
                    <option value="Cashier">Cashier</option>
                    <option value="Cleaner">Cleaner</option>
                    <option value="Cook">Cook</option>
                    <option value="Front Desk Clerk">Front Desk Clerk</option>
                    <option value="Housekeeping Supervisor">Housekeeping Supervisor</option>
                    <option value="Manager">Manager</option>
                    <option value="Receptionist">Receptionist</option>
                </select>
            </div>

```

```

<option value="Regional Manager">Regional Manager</option>
<option value="Waiter">Waiter</option>
</select>
</div>

<!-- Select field for Manager Status -->
<div class="inputbox">
    <label for="managerStatus">Manager Status:</label>
    <select name="managerStatus" id="managerStatus" required>
        <!-- Default option acting as a placeholder -->
        <option value="" disabled selected>Select Manager Status</option>
        <option value="Yes">Yes</option>
        <option value="No">No</option>
    </select>
</div>

<!-- Input for Login Name -->
<div class="inputbox">
    <label for="loginName">Login Name:</label>
    <input type="text" name="loginName" id="loginName" placeholder="Enter Login Name." maxlength="50"
required>
</div>

<!-- Hidden input for Password (default set to "staff") -->
<div>
    <input type="hidden" name="password" id="password" value="staff"/>
</div>
</fieldset>

<!-- Submit and Reset buttons -->
<div class="myButton">
    <!--
        Submit button.
        The onsubmit calls a confirm() dialog to verify the user's intention.
        If the user cancels, the form won't submit.
    -->
    <input type="submit" value="Add" name="submit"/>
    <input type="reset" value="Clear" name="reset"/>
</div>
</form>
</div>

<!-- Include external JavaScript for sidebar functionality -->
<script src="../../sidebar.js"></script>
</body>
</html>

```

## addStaff.php

```

<?php
//Name Of Screen: addStaff.php
//Purpose Of Screen:
//Student ID: C00294564

```

```

//Name: Mantas Bielevicius
//Date Written: March 2025

// Include the database connection file
include '../db.inc.php';

// Set the default timezone to UTC
date_default_timezone_set("UTC");

// Display the details sent from the form
echo "The details sent down are: <br>";

echo "First Name: " . $_POST['firstName'] . "<br>";
echo "Surname : " . $_POST['surname'] . "<br>";
echo "Address: " . $_POST['address'] . "<br>";
echo "Eircode: " . $_POST['eircode'] . "<br>";
echo "Phone Number: " . $_POST['phoneNumber'] . "<br>";
echo "Job Title: " . $_POST['jobTitle'] . "<br>";
echo "Manager Status: " . $_POST['managerStatus'] . "<br>";
echo "Login Name: " . $_POST['loginName'] . "<br>";

// SQL query to insert the staff details into the database
$sql = "INSERT INTO Staff (FirstName, Surname, Address, Eircode, TelephoneNumber, JobTitle, ManagerStatus, LoginName)
VALUES ('$_POST[firstName]', '$_POST[surname]', '$_POST[address]', '$_POST[eircode]', '$_POST[phoneNumber]',
'$_POST[jobTitle]', '$_POST[managerStatus]', '$_POST[loginName]')";

// Execute the query and check for errors
if (!mysqli_query($con, $sql)) {
    die("An Error in the SQL Query: " . mysqli_error($con));
}

// Confirms the record has been added
echo "<br>A record has been added for " . $_POST['firstName'] . " " . $_POST['surname'];

// Closes the database connection
mysqli_close($con);

// Redirects to add staff page
header('Location: addStaff.html.php');
exit();
?>

```

## addStaff.js

```

// Name Of Screen: addStaff.js
// Purpose Of Screen: Client-side validation for adding a new staff member, including confirming the action before submission.
// Student ID: C00294564
// Name: Mantas Bielevicius
// Date Written: March 2025

// Added an event listener to the "addStaffForm" to perform validation before form submission
document.getElementById('addStaffForm').addEventListener('submit', function(event) {
    // Fetch input values and trim whitespace for validation
    let firstName = document.getElementById('firstName').value.trim();
    let surname = document.getElementById('surname').value.trim();
    let address = document.getElementById('address').value.trim();
    let eircode = document.getElementById('eircode').value.trim();
    let phoneNumber = document.getElementById('phoneNumber').value.trim();
    let jobTitle = document.getElementById('jobTitle').value.trim();
    let loginName = document.getElementById('loginName').value.trim();
    let managerStatus = document.getElementById('managerStatus').value;

    // With .value it would be working with the HTML element itself rather than the data the user entered.
    // Without .value it returns the input element, but adding .value retrieves the text that the user has typed in.

    // Pattern for names: letters and spaces only
    let namePattern = /^[A-Za-z ]+$/;
    // Pattern for address: letters, numbers, spaces, commas, periods, and hyphens allowed
    let addressPattern = /^[A-Za-z0-9\s,.]+$/;
    // Pattern for Eircode: e.g., R32ABDD (adjust as necessary for your country's format. Also allows option to have in R32 ABDD
    // format with a space.)
    let eircodePattern = /^[A-Za-z]d{2}\s?[A-Za-z0-9]{4}\$/;
    // Pattern for phone numbers: numbers and spaces only, minimum 7 digits and maximum 15 digits
    let phonePattern = /^[0-9 ]{7,15}\$/;

    // Validation checks for each field
    if (!namePattern.test(firstName)) {
        alert("First name should only include letters and spaces.");
        event.preventDefault(); // Prevent form submission if validation fails
        return;
    }

    if (!namePattern.test(surname)) {
        alert("Surname should only include letters and spaces.");
        event.preventDefault();
        return;
    }

    if (!addressPattern.test(address)) {
        alert("Enter a valid address with letters, numbers, and common symbols.");
        event.preventDefault();
        return;
    }

    if (!eircodePattern.test(eircode)) {
        alert("Enter a valid Eircode (e.g., R32ABDD).");
        event.preventDefault();
        return;
    }

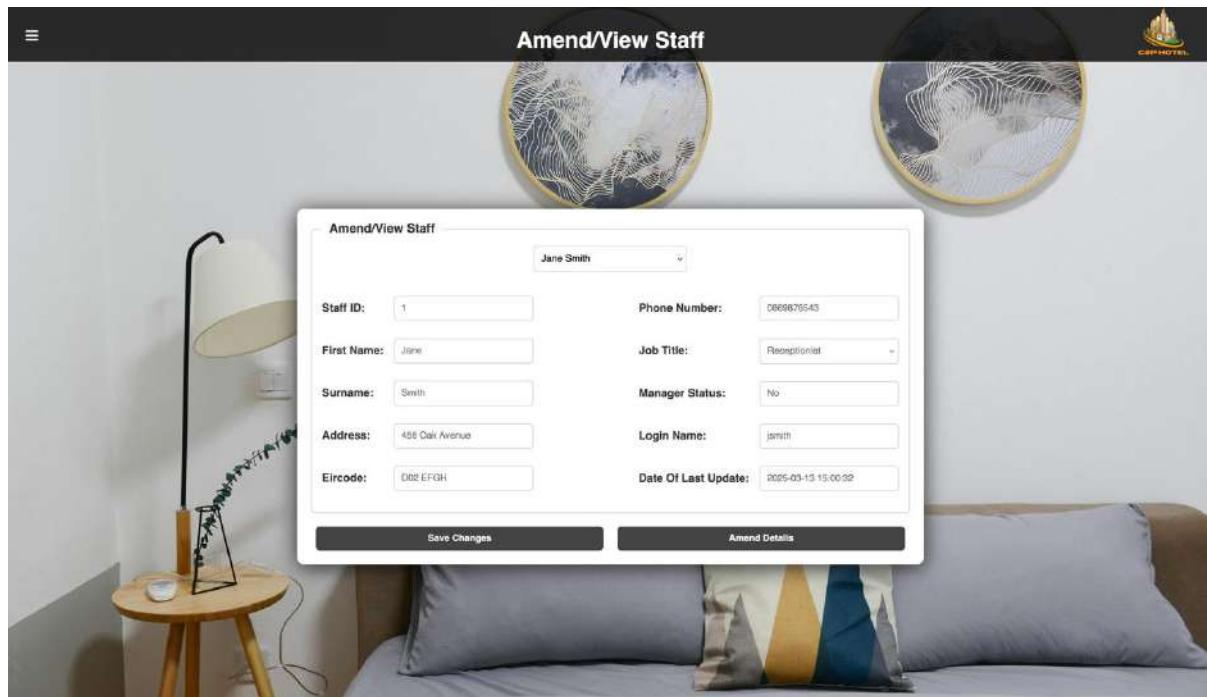
    if (!phonePattern.test(phoneNumber)) {
        alert("Enter a valid phone number (7-15 digits).");
        event.preventDefault();
        return;
    }

    // Validates job title against the same pattern as names (if appropriate)
}

```

```
if (!namePattern.test(jobTitle)) {  
    alert("Job title should only include letters and spaces.");  
    event.preventDefault();  
    return;  
}  
  
// Makes sure login name is not empty  
if (loginName === "") {  
    alert("Login name cannot be empty.");  
    event.preventDefault();  
    return;  
}  
  
// Makes sure a manager status is selected  
if (managerStatus === "") {  
    alert("Please select if the staff member is a manager.");  
    event.preventDefault();  
    return;  
}  
});
```

## Amend/View Screen



## amendViewStaff.html.php

```

<!DOCTYPE html>
<!--
Name Of Screen: amendViewStaff.html.php
Purpose Of Screen: Display and allow amendment/viewing of staff details.
Student ID: C00294564
Name: Mantas Bilevicius
Date Written: March 2025
-->
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Amend/View Staff</title>
<!-- Link to external CSS for styling -->
<link rel="stylesheet" href="../../styles.css">
<link rel="webiste icon" value="png" href="../../C2PLogoNew.png">
</head>
<body>
<!-- Burger icon for toggling the sidebar -->
<div class="burger-menu">&#9776;</div>

<!-- Page header with title and logo -->
<div class="page-header">
<h1>Amend/View Staff</h1>
<a href="../../home.html">
  
</a>
</div>

<!-- Include sidebar navigation from an external PHP file -->
<?php include '../../sidebar.php'; ?>

<!-- Main content area for Amend/View Staff form -->
<div class="content amend-view-page">

<!--
  JavaScript functions to populate and manage the form fields.
  These functions help update the form based on user selection,
  toggle the fields between editable and non-editable, and confirm changes.
-->
<script>
/*
Function: populate
Purpose: Populates form fields with the details of a selected staff member.
It Retrieves the value from the listbox, splits it by commas,
and assigns each detail to the corresponding form field.
*/
function populate() {
  // Gets listbox element that contains staff details
  var sel = document.getElementById("listbox");
  // Gets selected option's value
  var result = sel.options[sel.selectedIndex].value;
  // Splits string by commas to separate individual details
  var personDetails = result.split(',');
  // Populate form fields with staff details
  document.getElementById("amendStaffID").value = personDetails[0];
  // Hidden field to preserve original staff ID for database updates because it can't
  // be submitted if the input field is disabled
  document.getElementById("hiddenAmendStaffID").value = personDetails[0];
  document.getElementById("amendFirstName").value = personDetails[1];
}

```

```

document.getElementById("amendSurname").value = personDetails[2];
document.getElementById("amendAddress").value = personDetails[3];
document.getElementById("amendEircode").value = personDetails[4];
document.getElementById("amendPhoneNumber").value = personDetails[5];
// Populates the select field for job title (Value must match one of the options)
document.getElementById("amendJobTitle").value = personDetails[6];
document.getElementById("amendManagerStatus").value = personDetails[7];
document.getElementById("amendLoginName").value = personDetails[8];
document.getElementById("amendDateLastUpdated").value = personDetails[9];
}

/*
Function: toggleLock
Purpose: Toggles form fields between editable (enabled) and non-editable (disabled) states.
- When the "Amend Details" button is clicked, form fields become editable.
- When the "View Details" button is clicked, form fields become disabled.
*/
function toggleLock() {
    // Checks current value of the toggle button
    if (document.getElementById("amendViewbutton").value == "Amend Details") {
        // Enables fields for editing
        document.getElementById("amendFirstName").disabled = false;
        document.getElementById("amendSurname").disabled = false;
        document.getElementById("amendAddress").disabled = false;
        document.getElementById("amendEircode").disabled = false;
        document.getElementById("amendPhoneNumber").disabled = false;
        document.getElementById("amendJobTitle").disabled = false;
        // Changes the button text to "View Details"
        document.getElementById("amendViewbutton").value = "View Details";
    } else {
        // Disables fields for viewing
        document.getElementById("amendFirstName").disabled = true;
        document.getElementById("amendSurname").disabled = true;
        document.getElementById("amendAddress").disabled = true;
        document.getElementById("amendEircode").disabled = true;
        document.getElementById("amendPhoneNumber").disabled = true;
        document.getElementById("amendJobTitle").disabled = true;
        // Changes the button text back to "Amend Details"
        document.getElementById("amendViewbutton").value = "Amend Details";
    }
}

/*
Function: confirmCheck
Purpose: Asks for confirmation before saving changes.
- Displays a confirmation dialog.
- If confirmed, re-enables all fields for submission.
- If cancelled, re-populates the form with original data and toggles fields to non-editable.
- Returns true if the user confirms, false otherwise.
*/
function confirmCheck() {
    var response = confirm("Are you sure you want to save these changes?");
    if (response) {
        // Enables all form fields to ensure data is submitted
        document.getElementById("amendStaffID").disabled = false;
        document.getElementById("amendFirstName").disabled = false;
        document.getElementById("amendSurname").disabled = false;
        document.getElementById("amendAddress").disabled = false;
        document.getElementById("amendEircode").disabled = false;
        document.getElementById("amendPhoneNumber").disabled = false;
        document.getElementById("amendJobTitle").disabled = false;
    }
}

```

```

document.getElementById("amendManagerStatus").disabled = false;
document.getElementById("amendLoginName").disabled = false;
document.getElementById("amendDateLastUpdated").disabled = false;

return true; // Allow form submission
} else {
    // If cancelled, restore original data and set fields to non-editable
    populate();
    toggleLock();
    return false; // Prevent form submission
}
}

</script>

<!-- Staff amendment/view form -->
<!-- The form submits to amendViewStaff.php, where the database is updated -->
<form class="content.amend-view-page" id="addStaffForm" action="amendViewStaff.php" onsubmit="return
confirmCheck()" method="POST">
    <fieldset>
        <!-- Fieldset legend describing the form section -->
        <legend>Amend/View Staff</legend>

        <!-- Container for the select listbox -->
        <div class="centered-select">
            <!--
The onchange event here calls the populate() function to update the form fields based on the selected staff member. -->
            <div class="inputbox" onchange="populate()">
                <?php include 'amendViewStaffListbox.php'; ?>
            </div>
        </div>

        <!-- Two-column layout for form fields -->
        <div class="columns">
            <div class="left-column">
                <!-- Staff ID (displayed but disabled) -->
                <div class="inputbox">
                    <label for="amendStaffID">Staff ID: </label>
                    <input type="text" name="amendStaffID" id="amendStaffID" placeholder="Staff ID" disabled/>
                    <!-- Hidden field to hold the staff ID for submission -->
                    <input type="hidden" name="amendStaffID" id="hiddenAmendStaffID"/>
                </div>
                <!-- First Name -->
                <div class="inputbox">
                    <label for="amendFirstName">First Name: </label>
                    <input type="text" pattern="[A-Za-z]+" name="amendFirstName" id="amendFirstName" title="First name should
only include letters." placeholder="Enter First Name" maxlength="50" autofocus required disabled />
                </div>
                <!-- Surname -->
                <div class="inputbox">
                    <label for="amendSurname">Surname:</label>
                    <input type="text" pattern="[A-Za-z]+" name="amendSurname" id="amendSurname" placeholder="Enter
Surname" title="Surname should only include letters." maxlength="50" required disabled />
                </div>
                <!-- Address -->
                <div class="inputbox">
                    <label for="amendAddress">Address:</label>
                    <input type="text" pattern="^([A-Za-z0-9\s,-]+)$" name="amendAddress" id="amendAddress" placeholder="Enter
Address" title="Enter a valid address with letters, numbers, and common symbols." maxlength="255" required disabled />
                </div>
                <!-- Eircode -->
                <div class="inputbox">
                    <label for="amendEircode">Eircode:</label>
                </div>
            </div>
        </div>
    </fieldset>

```

The onchange event here calls the populate() function to update the form fields based on the selected staff member. -->

```

            <div class="inputbox" onchange="populate()">
                <?php include 'amendViewStaffListbox.php'; ?>
            </div>
        </div>

```

```

        <!-- Two-column layout for form fields -->
        <div class="columns">
            <div class="left-column">
                <!-- Staff ID (displayed but disabled) -->
                <div class="inputbox">
                    <label for="amendStaffID">Staff ID: </label>
                    <input type="text" name="amendStaffID" id="amendStaffID" placeholder="Staff ID" disabled/>
                    <!-- Hidden field to hold the staff ID for submission -->
                    <input type="hidden" name="amendStaffID" id="hiddenAmendStaffID"/>
                </div>
                <!-- First Name -->
                <div class="inputbox">
                    <label for="amendFirstName">First Name: </label>
                    <input type="text" pattern="[A-Za-z]+" name="amendFirstName" id="amendFirstName" title="First name should
only include letters." placeholder="Enter First Name" maxlength="50" autofocus required disabled />
                </div>
                <!-- Surname -->
                <div class="inputbox">
                    <label for="amendSurname">Surname:</label>
                    <input type="text" pattern="[A-Za-z]+" name="amendSurname" id="amendSurname" placeholder="Enter
Surname" title="Surname should only include letters." maxlength="50" required disabled />
                </div>
                <!-- Address -->
                <div class="inputbox">
                    <label for="amendAddress">Address:</label>
                    <input type="text" pattern="^([A-Za-z0-9\s,-]+)$" name="amendAddress" id="amendAddress" placeholder="Enter
Address" title="Enter a valid address with letters, numbers, and common symbols." maxlength="255" required disabled />
                </div>
                <!-- Eircode -->
                <div class="inputbox">
                    <label for="amendEircode">Eircode:</label>
                </div>
            </div>
        </div>
    </fieldset>

```

```

<input type="text" pattern="^([A-Za-z]{2})[0-9]{4}$" name="amendEircode" id="amendEircode"
placeholder="Enter Eircode" title="Enter a valid Eircode (e.g R32ABDD)." required disabled />
</div>
</div>

<div class="right-column">
    <!-- Phone Number -->
    <div class="inputbox">
        <label for="amendPhoneNumber">Phone Number:</label>
        <input type="text" pattern="[0-9 ]*" name="amendPhoneNumber" id="amendPhoneNumber" placeholder="Enter
Phone Number" title="Only numbers and spaces are allowed." maxlength="15" required disabled/>
    </div>
    <!-- Job Title (select field) -->
    <div class="inputbox">
        <label for="amendJobTitle">Job Title:</label>
        <select id="amendJobTitle" name="amendJobTitle" disabled>
            <!-- Default placeholder option -->
            <option value="" disabled selected>Select job title</option>
            <!-- Options ordered alphabetically -->
            <option value="Assistant Manager">Assistant Manager</option>
            <option value="Cashier">Cashier</option>
            <option value="Cleaner">Cleaner</option>
            <option value="Cook">Cook</option>
            <option value="Front Desk Clerk">Front Desk Clerk</option>
            <option value="Housekeeping Supervisor">Housekeeping Supervisor</option>
            <option value="Manager">Manager</option>
            <option value="Receptionist">Receptionist</option>
            <option value="Regional Manager">Regional Manager</option>
            <option value="Waiter">Waiter</option>
        </select>
    </div>
    <!-- Manager Status (input field for simplicity) -->
    <div class="inputbox">
        <label for="amendManagerStatus">Manager Status:</label>
        <input type="text" name="amendManagerStatus" id="amendManagerStatus" placeholder="Manager Status"
title="Select if the staff is a manager." required disabled>
    </div>
    <!-- Login Name -->
    <div class="inputbox">
        <label for="amendLoginName">Login Name:</label>
        <input type="text" name="amendLoginName" id="amendLoginName" placeholder="Login Name" title="Enter a
unique login name for the staff." maxlength="50" required disabled/>
    </div>
    <!-- Date of Last Update -->
    <div class="inputbox">
        <label for="amendDateLastUpdated">Date Of Last Update:</label>
        <input type="text" name="amendDateLastUpdated" id="amendDateLastUpdated" placeholder="Date Of Last
Update" disabled/>
    </div>
</div>
</div>
</fieldset>
<!-- Submit and reset buttons -->
<div class="myButton">
    <!-- When submitted, the confirmCheck() function runs to validate the changes -->
    <input type="submit" value="Save Changes" name="submit"/>
    <!-- Button to toggle between "Amend Details" and "View Details" -->
    <input type="button" value="Amend Details" id="amendViewbutton" onclick="toggleLock()">
</div>
</form>
</div>
<!-- Link to external JavaScript file for additional functionality -->

```

```
<script src="addStaff.js" defer></script>
<script src="../../sidebar.js"></script>
</body>
</html>
```

```

amendViewStaff.php
<?php
//Name Of Screen: amendViewStaff.php
//Purpose Of Screen:
//Student ID: C00294564
//Name: Mantas Bilevicius
//Date Written: March 2025

// Include the database connection details from the 'db.inc.php' file
include '../db.inc.php';

// Set the default timezone to UTC for correct date formatting
date_default_timezone_set('UTC');
// SQL query to update the staff member's details in the 'Staff' table
$sql = "UPDATE Staff SET
    FirstName = '$_POST[amendFirstName]',
    Surname = '$_POST[amendSurname]',
    Address = '$_POST[amendAddress]',
    Eircode = '$_POST[amendEircode]',
    TelephoneNumber = '$_POST[amendPhoneNumber]',
    JobTitle = '$_POST[amendJobTitle]'
    WHERE StaffID = '$_POST[amendStaffID]'";
echo $sql;

// Execute the SQL query
if (!mysqli_query($con, $sql)) {
    // If the query execution fails, output an error message along with the MySQL error
    echo "Error: " . mysqli_error($con);
} else {
    // If the query executes successfully, check if any rows were affected by the update
    if (mysqli_affected_rows($con) != 0) {
        // If rows were affected (i.e., updated), display how many records were updated
        echo mysqli_affected_rows($con) . " record(s) updated <br>";
    } else {
        // If no rows were affected, output a message indicating that no records were changed
        echo "No records were changed";
    }
}
// Closes database connection to free resources
mysqli_close($con);
// Redirects to delete staff page
header('Location: amendViewStaff.html.php');
exit();
?>

```

```

amendViewStaffListbox.php
<?php
// Name Of Screen: amendViewStaffListbox.php
// Purpose Of Screen: Retrieve staff details from the database and populate a listbox (select element) for viewing or editing staff records.
// Student ID: C00294564
// Name: Mantas Bielevicius
// Date Written: March 2025

// Include the database connection details from the 'db.inc.php' file
include "../db.inc.php"; // Establish database connection

// Set the default timezone to UTC
date_default_timezone_set('UTC');

// SQL query to retrieve active (non-deleted) staff details
$sql = "SELECT StaffID, FirstName, Surname, Address, Eircode, TelephoneNumber, JobTitle, ManagerStatus, LoginName, DateLastUpdate FROM Staff WHERE DeletedFlag = 0";

// Execute the query and check if it is successful
if (!$result = mysqli_query($con, $sql)) {
    // If the query fails, output an error message and terminate the script
    die('Error in querying the database: ' . mysqli_error($con));
}

// Start the HTML code for a drop-down list (select box)
// The "onclick" attribute is used to call the "populate()" JavaScript function
echo "<br><select name='listbox' id='listbox' onclick='populate()>";

// Loop through each row returned by the query
while ($row = mysqli_fetch_array($result)) {
    // Retrieve individual columns from the current row
    $StaffID      = $row['StaffID'];
    $FirstName    = $row['FirstName'];
    $Surname      = $row['Surname'];
    $Address      = $row['Address'];
    $Eircode      = $row['Eircode'];
    $TelephoneNumber = $row['TelephoneNumber'];
    $JobTitle     = $row['JobTitle'];
    $ManagerStatus = $row['ManagerStatus'];
    $LoginName    = $row['LoginName'];
    $DateLastUpdate = $row['DateLastUpdate'];

    // Combines all the details into a single comma-separated string.
    // This string is used as the value attribute of the option, so that the JavaScript "populate()"
    // function can parse and assign the values to the corresponding form fields.
    $alltext =
"$StaffID,$FirstName,$Surname,$Address,$Eircode,$TelephoneNumber,$JobTitle,$ManagerStatus,$LoginName,$DateLastUpdate";

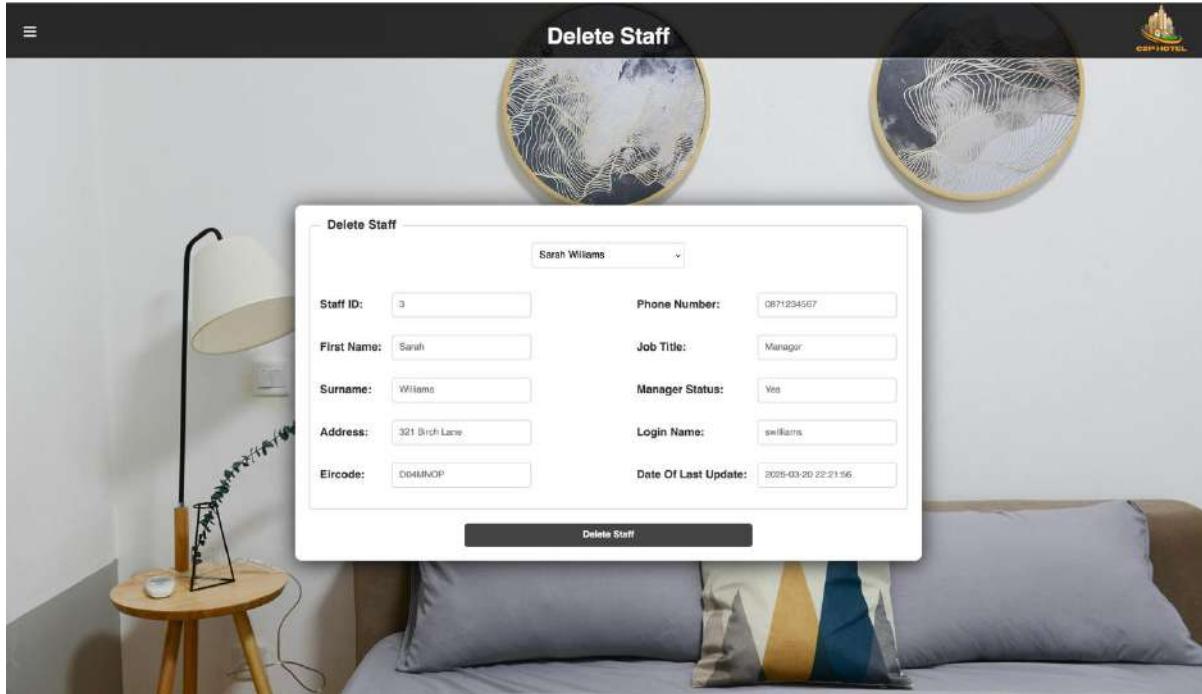
    // Output an option element for the current staff member.
    // The visible text displays the staff member's first and last name.
    echo "<option value='$alltext'$>$FirstName $Surname</option>";
}

// Close the select box tag after populating it with all options
echo "</select>";

// Close the database connection to free resources
mysqli_close($con);
?>

```

## Delete Screen



delete.html.php

<!DOCTYPE html>

<!--

Name Of Screen: deleteStaff.html.php

Purpose Of Screen: Display a form to delete a staff member, pre-populated with that member's details.

Student ID: C00294564

Name: Mantas Bilevicius

Date Written: March 2025

-->

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Delete Staff</title>

<!-- Link to external CSS for styling -->

<link rel="stylesheet" href="../../styles.css">

<link rel="webiste icon" value="png" href="../../C2PLogoNew.png">

</head>

<body>

<!-- Burger icon to toggle the sidebar -->

<div class="burger-menu">&#9776;</div>

<!-- Page header with title and logo -->

<div class="page-header">

<h1>Delete Staff</h1>

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



</a>

</div>

<!-- Sidebar navigation from external PHP file -->

<?php include '../../sidebar.php'; ?>

<!-- Main content area -->

<div class="content amend-view-page">

<!-- JavaScript functions to populate and manage the form fields -->

```

<script>
/*
Function: populate
Purpose: Populates the form fields with the selected staff member's details.
Retrieves the value from the listbox and splits the comma-separated string.
Assigns each part of the string to the appropriate input field.
*/
function populate() {
    // Get the listbox element that contains staff member options
    var sel = document.getElementById("listbox");
    // Retrieve the selected option's value
    var result = sel.options[sel.selectedIndex].value;
    // Split the string into individual details based on commas
    var personDetails = result.split(',');
}

// Populate the form fields with the details from the selected option
document.getElementById("delStaffID").value = personDetails[0];
document.getElementById("hiddenDelStaffID").value = personDetails[0];
document.getElementById("delFirstName").value = personDetails[1];
document.getElementById("delSurname").value = personDetails[2];
document.getElementById("delAddress").value = personDetails[3];
document.getElementById("delEircode").value = personDetails[4];
document.getElementById("delPhoneNumber").value = personDetails[5];
document.getElementById("delJobTitle").value = personDetails[6];
document.getElementById("delManagerStatus").value = personDetails[7];
document.getElementById("delLoginName").value = personDetails[8];
document.getElementById("delDateLastUpdated").value = personDetails[9];
}

/*
Disable Input Fields Immediately on Page Load
Purpose: Ensure that all input fields are disabled so they cannot be manually altered.
These lines run as soon as the script is parsed.
*/
document.getElementById("delStaffID").disabled = true;
document.getElementById("delFirstName").disabled = true;
document.getElementById("delSurname").disabled = true;
document.getElementById("delAddress").disabled = true;
document.getElementById("delEircode").disabled = true;
document.getElementById("delPhoneNumber").disabled = true;
document.getElementById("delJobTitle").disabled = true;
document.getElementById("delManagerStatus").disabled = true;
document.getElementById("delLoginName").disabled = true;
document.getElementById("delDateLastUpdated").disabled = true;

/*
Function: confirmCheck
Purpose: Prompts the user for confirmation before deleting the staff member.
- If confirmed, re-enables the fields so their values can be submitted.
- If not confirmed, re-populates the fields and prevents form submission.
- @returns {boolean} true if the user confirms deletion, false otherwise.
*/
function confirmCheck() {
    // Display confirmation dialog
    var response = confirm('Are you sure you want to delete this staff member?');

    if (response) {
        // Enable all fields so that their values are submitted with the form
        document.getElementById("delStaffID").disabled = false;
        document.getElementById("delFirstName").disabled = false;
        document.getElementById("delSurname").disabled = false;
        document.getElementById("delAddress").disabled = false;
    }
}

```

```

document.getElementById("delEircode").disabled = false;
document.getElementById("delPhoneNumber").disabled = false;
document.getElementById("delJobTitle").disabled = false;
document.getElementById("delManagerStatus").disabled = false;
document.getElementById("delLoginName").disabled = false;
document.getElementById("delDateLastUpdated").disabled = false;

    return true; // Allow form submission
} else {
    // If cancelled, re-populate the form with the original data
    populate();
    return false; // Prevent form submission
}
}

</script>


<!-- Form action points to deleteStaff.php where the deletion is processed --&gt;
&lt;form class="content.amend-view-page" id="deleteStaff" action="deleteStaff.php" onsubmit="return confirmCheck()" method="POST"&gt;
&lt;fieldset&gt;
    &lt;!-- Fieldset legend for grouping the form controls --&gt;
    &lt;legend&gt;Delete Staff&lt;/legend&gt;
    &lt;div class="centered-select"&gt;
&lt;!-- When an option is selected, the populate() function is called --&gt;
        &lt;div class="inputbox" onchange="populate()"&gt;
            &lt;?php include 'amendViewStaffListbox.php'; ?&gt;
        &lt;/div&gt;
    &lt;/div&gt;

    &lt;!-- Two-column layout for form fields --&gt;
    &lt;div class="columns"&gt;
        &lt;div class="left-column"&gt;
&lt;!-- Staff ID: displayed for information, stored in a hidden field for submission --&gt;
        &lt;div class="inputbox"&gt;
            &lt;label for="delStaffID"&gt;Staff ID: &lt;/label&gt;
            &lt;input type="text" name="delStaffID" id="delStaffID" placeholder="Staff ID" disabled/&gt;
            &lt;input type="hidden" name="delStaffID" id="hiddenDelStaffID" disabled/&gt;
        &lt;/div&gt;
        &lt;!-- First Name --&gt;
        &lt;div class="inputbox"&gt;
            &lt;label for="delFirstName"&gt;First Name: &lt;/label&gt;
            &lt;input type="text" name="delFirstName" id="delFirstName" placeholder="First Name" disabled/&gt;
        &lt;/div&gt;
        &lt;!-- Surname --&gt;
        &lt;div class="inputbox"&gt;
            &lt;label for="delSurname"&gt;Surname:&lt;/label&gt;
            &lt;input type="text" name="delSurname" id="delSurname" placeholder="Surname" disabled/&gt;
        &lt;/div&gt;
        &lt;!-- Address --&gt;
        &lt;div class="inputbox"&gt;
            &lt;label for="delAddress"&gt;Address:&lt;/label&gt;
            &lt;input type="text" name="delAddress" id="delAddress" placeholder="Address" disabled/&gt;
        &lt;/div&gt;
        &lt;!-- Eircode --&gt;
        &lt;div class="inputbox"&gt;
            &lt;label for="delEircode"&gt;Eircode:&lt;/label&gt;
            &lt;input type="text" name="delEircode" id="delEircode" placeholder="Eircode" disabled /&gt;
        &lt;/div&gt;
    &lt;/div&gt;
&lt;div class="right-column"&gt;
</pre>

```

```

<!-- Phone Number -->
<div class="inputbox">
    <label for="delPhoneNumber">Phone Number:</label>
    <input type="text" name="delPhoneNumber" id="delPhoneNumber" placeholder="Phone Number" disabled/>
</div>
<!-- Job Title -->
<div class="inputbox">
    <label for="delJobTitle">Job Title:</label>
    <input type="text" name="delJobTitle" id="delJobTitle" placeholder="Job Title" disabled/>
</div>
<!-- Manager Status -->
<div class="inputbox">
    <label for="delManagerStatus">Manager Status:</label>
    <input type="text" name="delManagerStatus" id="delManagerStatus" placeholder="Manager Status" disabled/>
</div>
<!-- Login Name -->
<div class="inputbox">
    <label for="delLoginName">Login Name:</label>
    <input type="text" name="delLoginName" id="delLoginName" placeholder="Login Name" disabled/>
</div>
<!-- Date of Last Update -->
<div class="inputbox">
    <label for="delDateLastUpdated">Date Of Last Update:</label>
    <input type="text" name="delDateLastUpdated" id="delDateLastUpdated" placeholder="Date Last Updated" disabled/>
</div>
</div>
</div>
</div>
</fieldset>

<!-- Submit button for deleting the staff record -->
<div class="myButton">
    <input type="submit" value="Delete Staff" name="submit" onsubmit="return confirmCheck()"/>
</div>
</form>
</div>

<!-- External JavaScript file for additional functionality (if needed) -->
<script src="addStaff.js" defer></script>
<script src="../../sidebar.js"></script>
</body>
</html>

```

```

delete.php
<?php
// Name Of Screen: deleteStaff.php
// Purpose Of Screen: Process deletion of a staff member by marking the record as deleted (soft delete).
// Student ID: C00294564
// Name: Mantas Bilevicius
// Date Written: March 2025

session_start(); // Starts the session to use session variables

// Includes the database connection file to establish a connection
include '../db.inc.php';

// Validates the input
// Ensures that the 'delStaffID' POST variable exists.
// If not, output an error message and stop execution.
if (!isset($_POST['delStaffID'])) {
    die('Error: Staff ID not provided.');
}

// Creates the SQL query to mark the staff record as deleted.
// The DeletedFlag is set to true/1 instead of permanently deleting it.
$sql = "UPDATE Staff SET DeletedFlag = true WHERE staffID = " . $_POST['delStaffID'] . "";

// Executes the SQL query and check for errors.
// If the query fails, it outputs an error message and stop execution.
if (!mysqli_query($con, $sql)) {
    echo "Error: " . mysqli_error($con);
    exit();
}

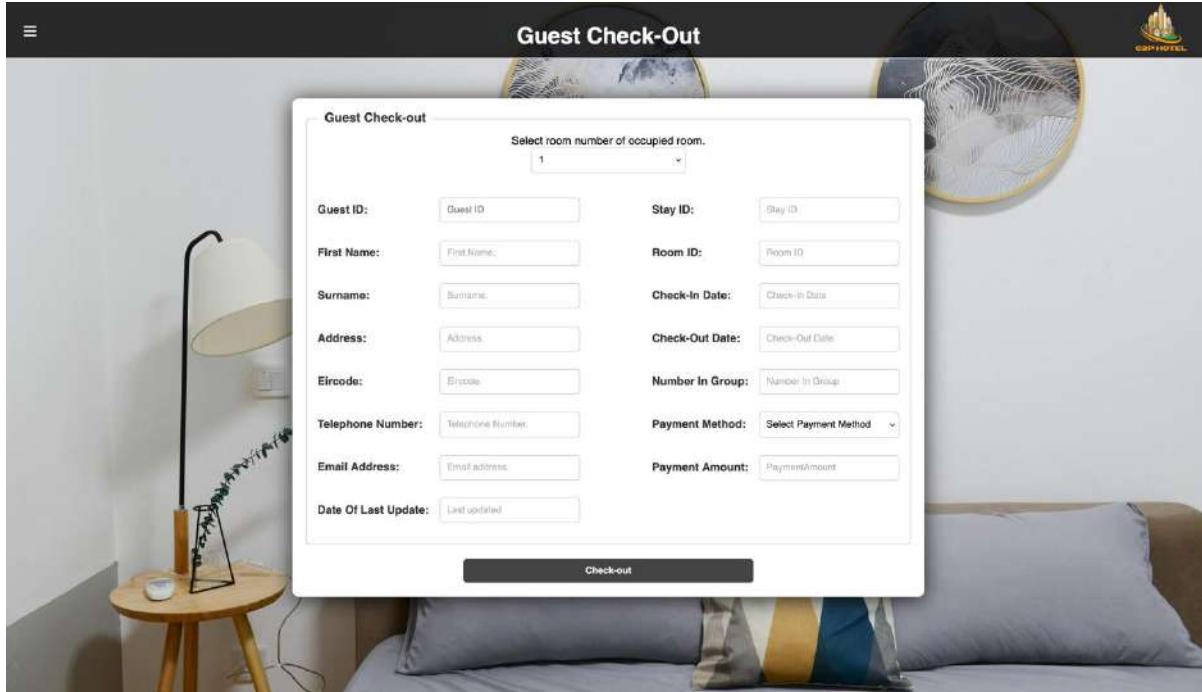
// Set session variables to store details of the deleted staff member.
// These variables can be used on the redirected page to display confirmation messages.
// Adjust variable names as necessary (e.g., use 'delSurname' if that's what your form sends).
$_SESSION["delStaffID"] = $_POST['delStaffID'];
$_SESSION["delFirstName"] = $_POST['delFirstName'];
$_SESSION["delLastName"] = $_POST['delLastName'];

// Closes the database connection to free resources.
mysqli_close($con);

// Redirects the user back to the delete staff page.
// The session variables set above can then be used to confirm the deletion.
header('Location: deleteStaff.html.php');
exit();
?>

```

## Guest Check-Out Screen



checkout.html.php

```
<!DOCTYPE html>
<!--
Name Of Screen: checkout.html.php
Purpose Of Screen: Display a guest check-out form pre-populated with guest and stay details from the selected occupied room.
Student ID: C00294564
Name: Mantas Bielevicius
Date Written: March 2025
-->
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Check-Out</title>
  <!-- Link to external CSS for styling -->
  <link rel="stylesheet" href="../styles.css">
  <link rel="webiste icon" value="png" href="../C2PLogoNew.png">
</head>
<body>
  <!-- Burger icon for toggling the sidebar -->
  <div class="burger-menu">&#9776;</div>

  <!-- Page header with logo and home page link -->
  <div class="page-header">
    <h1> Guest Check-Out </h1>
    <a href="../home.html">
      
    </a>
  </div>

  <!-- Include sidebar navigation from an external PHP file -->
  <?php include '../sidebar.php'; ?>
```

```

<!-- Main content area -->
<div class="content amend-view-page">
    <script>

        // Function: populate
        // Purpose: Populate the check-out form fields with details from the selected occupied room.
        // - Retrieves the value from the listbox (a comma-separated string).
        // - Splits the string into an array and assigns each value to its respective input field.

        function populate() {
            // Get the listbox element containing occupied room options
            var sel = document.getElementById("listbox");
            // Retrieve the value of the selected option
            var result = sel.options[sel.selectedIndex].value;
            // Split the string into an array using commas as delimiters
            var checkoutDetails = result.split(',');

            // Populate each field with the corresponding value from the array
            document.getElementById("GuestID").value = checkoutDetails[0];
            document.getElementById("FirstName").value = checkoutDetails[1];
            document.getElementById("Surname").value = checkoutDetails[2];
            document.getElementById("Address").value = checkoutDetails[3];
            document.getElementById("Eircode").value = checkoutDetails[4];
            document.getElementById("TelephoneNumber").value = checkoutDetails[5];
            document.getElementById("EmailAddress").value = checkoutDetails[6];
            document.getElementById("DateLastUpdate").value = checkoutDetails[7];
            document.getElementById("StayID").value = checkoutDetails[8];
            // Hidden field to ensure StayID is submitted even if the visible field is disabled
            document.getElementById("hiddenStayID").value = checkoutDetails[8];
            document.getElementById("RoomID").value = checkoutDetails[9];
            // Hidden field to ensure RoomID is submitted
            document.getElementById("hiddenRoomID").value = checkoutDetails[9];
            document.getElementById("CheckInDate").value = checkoutDetails[10];
            document.getElementById("CheckOutDate").value = checkoutDetails[11];
            // Hidden field for CheckOutDate submission
            document.getElementById("hiddenCheckOutDate").value = checkoutDetails[11];
            document.getElementById("NumberInGroup").value = checkoutDetails[12];
            document.getElementById("PaymentAmount").value = checkoutDetails[13];
        }

        // Function: confirmCheck
        // Purpose: Ask the user for confirmation before finalizing the check-out.
        // - If the user confirms, enable all fields (so their values are submitted).
        // - If the user cancels, repopulate the form with the original details.
        // - Returns true to allow form submission; false to prevent submission.

        function confirmCheck() {
            // Display a confirmation dialog
            var response = confirm('Are you sure you want to save these changes?');

            if (response) {
                // Enables all fields so their values are sent when the form is submitted
                document.getElementById("GuestID").disabled = false;
                document.getElementById("FirstName").disabled = false;
                document.getElementById("Surname").disabled = false;
                document.getElementById("Address").disabled = false;
                document.getElementById("Eircode").disabled = false;
                document.getElementById("TelephoneNumber").disabled = false;
                document.getElementById("EmailAddress").disabled = false;
                document.getElementById("DateLastUpdate").disabled = false;
                document.getElementById("StayID").disabled = false;
                document.getElementById("RoomID").disabled = false;
            }
        }
    </script>
</div>

```

```

document.getElementById("CheckInDate").disabled = false;
document.getElementById("CheckOutDate").disabled = false;
document.getElementById("NumberInGroup").disabled = false;
document.getElementById("PaymentMethod").disabled = false;
document.getElementById("PaymentAmount").disabled = false;

    return true; // Allow form submission
} else {
// If cancelled, restore the form to its original state by repopulating it
    populate();
    return false; // Prevent form submission
}
}

</script>

<!-- Guest check-out form -->
<!-- The form submits to checkout.php for processing the check-out -->
<form class="content.amend-view-page" id="checkout" action="checkout.php" method="POST">
<fieldset>
    <!-- Fieldset legend describing the purpose of this form section -->
    <legend>Guest Check-out</legend>
    <div class="centered-select">
        <!-- Container for the listbox -->
        <div class="inputbox" onchange="populate()">
            <div style="text-align: center;">
                <p style="color: black;">Select room number of occupied room.</p>
                <!-- Include a PHP file that generates a listbox of occupied rooms -->
                <?php include 'occupiedRoomsListbox.php'; ?>
            </div>
        </div>
    </div>
</div>

<!-- Two-column layout for guest and stay details -->
<div class="columns">
    <div class="left-column">
        <!-- Guest ID -->
        <div class="inputbox">
            <label for="GuestID">Guest ID: </label>
            <input type="text" name="GuestID" id="GuestID" placeholder="Guest ID" />
        </div>
        <!-- First Name (disabled for editing) -->
        <div class="inputbox">
            <label for="FirstName">First Name: </label>
            <input type="text" name="FirstName" id="FirstName" placeholder="First Name." disabled/>
        </div>
        <!-- Surname -->
        <div class="inputbox">
            <label for="Surname">Surname: </label>
            <input type="text" name="Surname" id="Surname" placeholder="Surname." disabled/>
        </div>
        <!-- Address -->
        <div class="inputbox">
            <label for="Address">Address: </label>
            <input type="text" name="Address" id="Address" placeholder="Address." disabled/>
        </div>
        <!-- Eircode -->
        <div class="inputbox">
            <label for="Eircode">Eircode: </label>
            <input type="text" name="Eircode" id="Eircode" placeholder="Eircode." disabled />
        </div>
        <!-- Telephone Number -->
        <div class="inputbox">

```

```

<label for="TelephoneNumber">Telephone Number:</label>
<input type="text" name="TelephoneNumber" id="TelephoneNumber" placeholder="Telephone Number." disabled />
</div>
<!-- Email Address -->
<div class="inputbox">
    <label for="EmailAddress">Email Address:</label>
    <input type="text" name="EmailAddress" id="EmailAddress" placeholder="Email address." disabled />
</div>
<!-- Date of Last Update -->
<div class="inputbox">
    <label for="DateLastUpdate">Date Of Last Update:</label>
    <input type="text" name="DateLastUpdate" id="DateLastUpdate" placeholder="Last updated" disabled/>
</div>
</div>

<div class="right-column">
    <!-- Stay ID (disabled, with hidden field for submission) -->
    <div class="inputbox">
        <label for="StayID">Stay ID:</label>
        <input type="text" name="StayID" id="StayID" placeholder="Stay ID." required disabled/>
        <input type="hidden" name="StayID" id="hiddenStayID" />
    </div>
    <!-- Room ID (disabled, with hidden field for submission) -->
    <div class="inputbox">
        <label for="RoomID">Room ID:</label>
        <input type="text" name="RoomID" id="RoomID" placeholder="Room ID" disabled/>
        <input type="hidden" name="RoomID" id="hiddenRoomID" />
    </div>
    <!-- Check-In Date -->
    <div class="inputbox">
        <label for="CheckInDate">Check-In Date:</label>
        <input type="text" name="CheckInDate" id="CheckInDate" placeholder="Check-In Date" disabled/>
    </div>
    <!-- Check-Out Date (disabled, with hidden field for submission) -->
    <div class="inputbox">
        <label for="CheckOutDate">Check-Out Date:</label>
        <input type="text" name="CheckOutDate" id="CheckOutDate" placeholder="Check-Out Date" disabled/>
        <input type="hidden" name="CheckOutDate" id="hiddenCheckOutDate" />
    </div>
    <!-- Number In Group -->
    <div class="inputbox">
        <label for="NumberInGroup">Number In Group:</label>
        <input type="text" name="NumberInGroup" id="NumberInGroup" placeholder="Number In Group" disabled/>
    </div>
    <!-- Payment Method (select box) -->
    <div class="inputbox">
        <label for="PaymentMethod">Payment Method:</label>
        <select id="PaymentMethod" name="PaymentMethod" required>
            <option value="" disabled selected>Select Payment Method</option>
            <option value="Cash">Cash</option>
            <option value="Cheque">Cheque</option>
            <option value="Debit Card">Debit Card</option>
            <option value="Credit Card">Credit Card</option>
        </select>
    </div>
    <!-- Payment Amount -->
    <div class="inputbox">
        <label for="PaymentAmount">Payment Amount:</label>
        <input type="text" name="PaymentAmount" id="PaymentAmount" placeholder="PaymentAmount" />
    </div>
</div>

```

```
</div>
</fieldset>

<!-- Submit button -->
<div class="myButton">
    <!-- onsubmit attribute in the input is redundant; the form's onsubmit is best handled on the form tag -->
    <input type="submit" value="Check-out" name="submit" onsubmit="return confirmCheck()" />
</div>
</form>
</div>

<!-- Include external JavaScript for sidebar functionality -->
<script src="../sidebar.js"></script>
</body>
</html>
```

```

checkout.php
<?php
// Name Of Screen: checkout.php
// Purpose Of Screen: Process guest check-out by updating the GuestStays record with check-out details
// and marking the corresponding room as available.
// Student ID: C00294564
// Name: Mantas Bielevicius
// Date Written: March 2025

// Include the database connection details from the 'db.inc.php' file
include '../db.inc.php';

// Set the default timezone to UTC for correct date formatting
date_default_timezone_set('UTC');

// Set the current date as DateOfPayment (format: YYYY-MM-DD)
$dateOfPayment = date("Y-m-d");

// First Query: Update the GuestStays table with check-out details
// This query updates the record corresponding to the provided StayID.
// It sets the CheckOutDate, TotalCost, PaymentMethod, PaymentAmount, and DateOfPayment.
$sql1 = "UPDATE GuestStays SET
    CheckOutDate = '$_POST[CheckOutDate]',
    TotalCost = '$_POST[PaymentAmount]',
    PaymentMethod = '$_POST[PaymentMethod]',
    PaymentAmount = '$_POST[PaymentAmount]',
    DateOfPayment = '$dateOfPayment'
    WHERE StayID = '$_POST[StayID]' ";

// Execute the first SQL query and check for errors.
if (!mysqli_query($con, $sql1)) {
    // If the query execution fails, output an error message along with the MySQL error.
    echo "Error: " . mysqli_error($con);
} else {
    // If the query executes successfully, check if any rows were affected.
    if (mysqli_affected_rows($con) != 0) {
        // If rows were updated, output the number of records updated.
        echo mysqli_affected_rows($con) . " record(s) updated <br>";
    } else {
        // If no rows were affected, output a message indicating that no records were changed.
        echo "No records were changed";
    }
}

// Second Query: Update the Rooms table to mark the room as available
// This query sets the CurrentStatus of the room (identified by RoomID) to "Available".
$available = "Available";
$sql2 = "UPDATE Rooms SET
    CurrentStatus = '$available'
    WHERE RoomID = '$_POST[RoomID]' ";

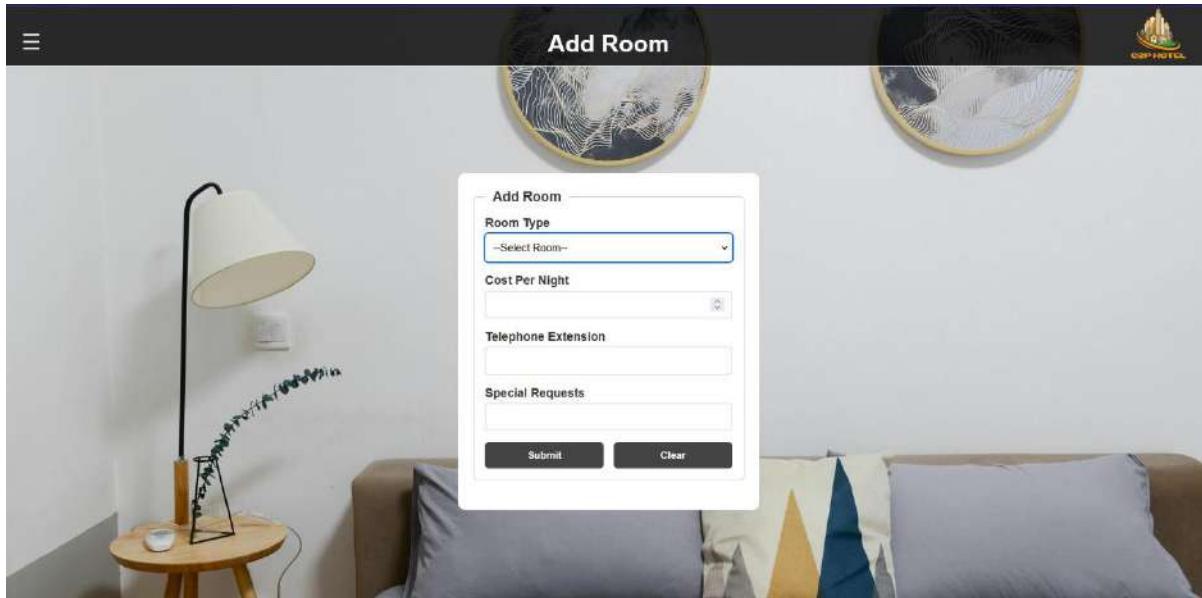
// Execute the second SQL query and check for errors.
if (!mysqli_query($con, $sql2)) {
    // If the query execution fails, output an error message along with the MySQL error.
    echo "Error: " . mysqli_error($con);
} else {
    // If the query executes successfully, check if any rows were affected.
    if (mysqli_affected_rows($con) != 0) {
        // If rows were updated, output the number of records updated.
        echo mysqli_affected_rows($con) . " record(s) updated <br>";
    } else {
        // If no rows were affected, output a message indicating that no records were changed.
        echo "No records were changed";
    }
}

```

```
    }  
}  
// Close the database connection to free resources.  
mysqli_close($con);  
// Redirect the user to the checkout page after processing.  
header('Location: checkout.html.php');  
exit();  
?>
```

Szymon Stepniak Written By: Szymon Stepniak

## Add Room Page



## Addroom.html.php

```

1  <!DOCTYPE html>
2  <!--
3      allows for rooms and it's details to be added into the database
4      Szymon Stepniak
5      C00300283
6  -->
7  <html lang="en">
8
9  <head>
10     <meta charset="UTF-8">
11     <title>Add Room</title>
12     <link rel="stylesheet" href="../../styles.css">
13     <link rel="webiste icon" value="png" href="../../C2PLogoNew.png">
14     <script src="validateRoom.js"></script>
15 </head>
16
17 <body>
18     <!-- Burger icon that will toggle the sidebar -->
19     <div class="burger-menu">=☰</div>
20     <div class="page-header">
21         <h1>Add Room</h1>
22         <a href="....home.html"></a>
23     </div>
24     <!-- Sidebar navigation -->
25     <?php include '....idebar.php'; ?>
26     <div class="content">
27         <form action="addRoom.php" method="post" onsubmit="return roomPrice()" onsubmit="return confirmAddRoom()">
28             <fieldset>
29                 <legend>Add Room</legend>
30                 <div class="inputbox">
31                     <label for="roomtype">Room Type</label>
32                     <select name="roomtype" id="roomtype" value="--Select Room--" autofocus>
33                         <option value="selectRoom">--Select Room--</option>
34                         <option value="single">Single</option>
35                         <option value="double">Double</option>
36                         <option value="family">Family</option>
37                         <option value="twin">Twin</option>
38                     </select>
39                 </div>
40                 <div class="inputbox">
41                     <label for="price">Cost Per Night</label>
42                     <input type="number" name="price" id="price" min="80" required />
43                 </div>
44                 <div class="inputbox">
45                     <label for="teleExtension">Telephone Extension</label>
46                     <input type="text" name="teleExtension" id="teleExtension" autofocus pattern="\d{3}" title="Extension can't be longer than 3 digits" required />
47                 </div>
48                 <div class="inputbox">
49                     <label for="specialNotes">Special Requests </label>
50                     <input type="text" name="specialNotes" id="specialNotes" pattern="A-Z,a-z" autofocus required />
51                 </div>
52                 <div class="myButton">
53                     <input type="submit" value="Submit" name="submit" />
54                     <input type="reset" value="Clear" name="reset" />
55                 </div>
56             </fieldset>
57         </form>
58     </div>
59 
```

```

1 <script>
2     document.addEventListener('DOMContentLoaded', function () {
3         // ===== SIDEBAR =====
4         const burger = document.querySelector('.burger-menu');
5         const sidebar = document.querySelector('.sidebar');
6
7         // Toggle sidebar on burger click
8         burger.addEventListener('click', (e) => {
9             e.stopPropagation();
10            sidebar.classList.toggle('open');
11        });
12
13         // Close sidebar when clicking outside
14         document.addEventListener('click', (e) => {
15             if (!sidebar.contains(e.target) && !e.target.classList.contains('burger-menu')) {
16                 sidebar.classList.remove('open');
17             }
18         });
19
20         // ===== DROPPDOWNS =====
21         const dropdowns = document.querySelectorAll('.dropdown');
22         let activeDropdown = null;
23
24         dropdowns.forEach(dropdown => {
25             const content = dropdown.querySelector('.dropdown-content');
26
27             // Show dropdown on hover
28             dropdown.addEventListener('mouseenter', () => {
29                 if (activeDropdown) {
30                     activeDropdown.classList.remove('visible');
31                 }
32                 content.classList.add('visible');
33                 activeDropdown = content;
34             });
35
36             // Hide dropdown with delay
37             dropdown.addEventListener('mouseleave', () => {
38                 setTimeout(() => {
39                     if (!content.matches(':hover')) {
40                         content.classList.remove('visible');
41                         activeDropdown = null;
42                     }
43                 }, 10);
44             });
45
46             // Keep dropdown open when hovering its content
47             content.addEventListener('mouseenter', () => {
48                 clearTimeout();
49                 content.classList.add('visible');
50             });
51
52             // Hide dropdown when leaving its content
53             content.addEventListener('mouseleave', () => {
54                 content.classList.remove('visible');
55                 activeDropdown = null;
56             });
57         });
58     });
59     </script>
60 </body>
61
62 </html>

```

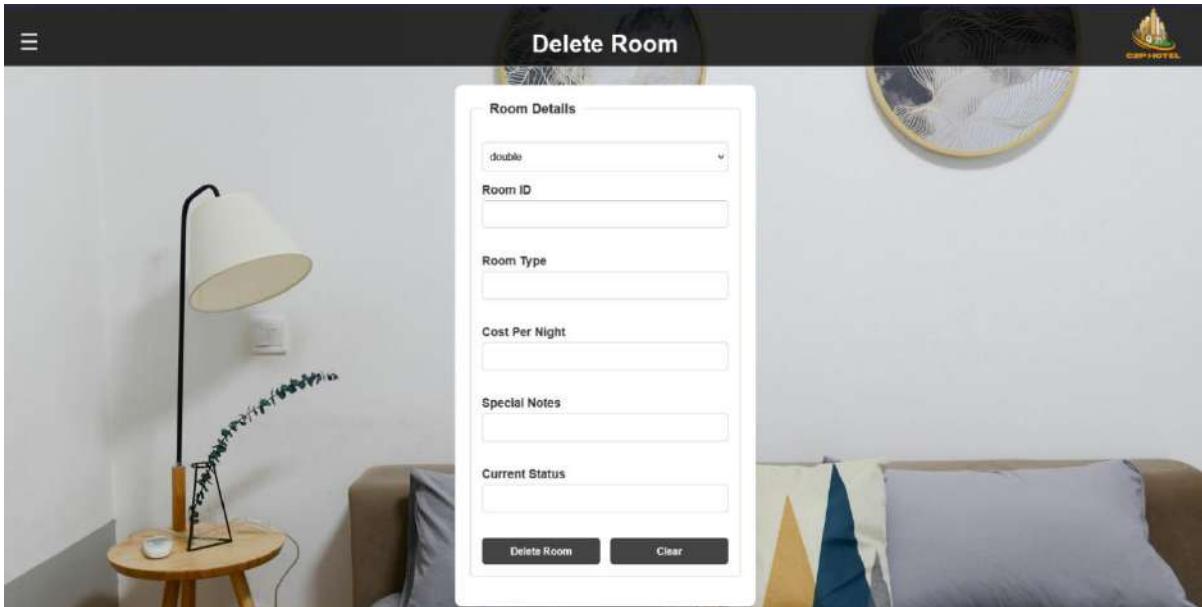
## addRoom.php

```
1 <?php
2     include 'db.inc.php';
3     date_default_timezone_set("UTC");
4     echo "The details sent down are: <br>";
5
6     echo "Room Type :" . $_POST['roomtype'] . "<br>";
7     echo "Price is :" . $_POST['price'] . "<br>";
8     echo "Email Address is :" . $_POST['teleExtension'] . "<br>";
9
10    $sql = "Insert into Rooms (Type,CostPerNight,TelephoneExtensionNumber,SpecialNotes) VALUES ('$_POST[roomtype]','$_POST[price]','$_POST[teleExtension]','$_POST[specialNotes]')";
11
12    if (!mysqli_query($con,$sql))
13    {
14        die ("An error in the SQL Query: " . mysqli_error($con));
15    }
16    echo "<br>A record has been added for " . $_POST['roomtype'] . ":" ;
17
18    mysqli_close($con);
19
20    ?>
21    <form action = "addrrom.html.php" method = "POST">
22        <br>
23        <input type="submit" value="Return to Insert Page"/>
24    </form>
25
```

## RoomValidation.js

```
1 // Validation for room type and the price of the room
2 // Szymon Stepniak
3 // c00300283
4 function roomPrice() {
5     let roomType = document.getElementById("roomtype").value;
6     let price = document.getElementById("price").value;
7
8     // Ensure a valid room type is selected
9     if (roomType === "selectRoom") {
10         alert("Please select a valid room type.");
11         return false;
12     }
13     // Ensure price is within the range of 100 to 220
14     price = parseFloat(price);
15     if (isNaN(price) || price < 100 || price > 220) {
16         alert("Room price must be between $100 and $220.");
17         return false;
18     }
19     if(!confirm("Are you sure all the details are correct?")){
20         return false;
21     }
22     return true; // Allow submission if everything is valid
23 }
24
```

## Delete Room



## deleteRoom.html.php

```

1  <!--
2   DeleteRoom.html.php
3   Used to delete Rooms from the database
4   Szymon Stepiak
5   C00300283
6 -->
7 <?php
8 include 'db.inc.php';
9 date_default_timezone_set('UTC');
10
11 $deleteMessage = '';// Variable to hold the deletion message
12
13 if ($_SERVER['REQUEST_METHOD'] == 'POST') {
14   $sql = "UPDATE Rooms set DeletedFlag = 1 WHERE RoomID = '$_POST[deleteRoomID]'";
15
16   if (!mysqli_query($con, $sql)) {
17     $deleteMessage = "Error: " . mysqli_error($con);
18   } else {
19     if (mysqli_affected_rows($con) > 0) {
20       $deleteMessage = mysqli_affected_rows($con) . " Record(s) Deleted <br>";
21       $deleteMessage .= "Guest ID " . $_POST['deleteRoomType'] .
22       " " . $_POST['deleteCostPerNight'] .
23       " " . $_POST['deleteSpecialNotes'] .
24       " " . $_POST['deleteCurrentStatus'] .
25       " Has been deleted";
26     } else {
27       $deleteMessage = "No records were changed";
28     }
29   }
30 }
31 mysqli_close($con);
32 ?
33
34
35 <!DOCTYPE html>
36
37 <html lang="en">
38
39 <head>
40   <meta charset="UTF-8">
41   <meta name="viewport" content="width=device-width, initial-scale=1.0">
42   <link rel="stylesheet" href="../../styles.css">
43   <link rel="webiste icon" value="png" href="../../C2PLogoNew.png">
44   <title>Delete Room</title>
45 </head>
46
47 <body>
48   <!-- Burger icon that will toggle the sidebar -->
49   <div class="burger-menu">&#9776;</div>
50   <div class="page-header">
51     <h1>Delete Room</h1>
52     <a href="../../home.html"></a>
53   </div>
54   <!-- Sidebar navigation -->
55   <?php include '../../sidebar.php'; ?>
56   <div class="content">
57     <script>
58       // Function to handle form submission
59       function confirmCheck() {
60         var response = confirm('Are You sure you want to delete this Room?');
61         if (response) {
62           document.getElementById("deleteRoomID").disabled = false;
63           document.getElementById("deleteRoomType").disabled = false;
64           document.getElementById("deleteCostPerNight").disabled = false;
65           document.getElementById("deleteSpecialNotes").disabled = false;
66           document.getElementById("deleteCurrentStatus").disabled = false;
67
68           return true; // Validate before submitting
69         } else {
70           populate();
71           toggleLock();
72           return false;
73         }
74       }
75
76       // Function to toggle the lock/unlock on the form fields
77       function toggleLock() {
78         if (document.getElementById("amendView").value == "Delete Guest") {
79           document.getElementById("deleteRoomType").disabled = false;
80           document.getElementById("deleteCostPerNight").disabled = false;
81           document.getElementById("deleteSpecialNotes").disabled = false;
82           document.getElementById("deleteCurrentStatus").disabled = false;
83           document.getElementById("delete").value = "View Details";
84         }
85       }
86
87       // Function to populate fields
88       function populate() {
89         var sel = document.getElementById("listbox");
90         var result = sel.options[sel.selectedIndex].value;
91         var personDetails = result.split(',');
92
93         document.getElementById("deleteRoomID").value = personDetails[0];
94         document.getElementById("deleteRoomType").value = personDetails[1];
95         document.getElementById("deleteCostPerNight").value = personDetails[2];
96         document.getElementById("deleteSpecialNotes").value = personDetails[3];
97         document.getElementById("deleteCurrentStatus").value = personDetails[4];
98       }
99     </script>

```

```

1 <form action="deleteRoom.html.php" method="POST" onsubmit="return confirmCheck()">
2     <fieldset>
3         <legend>Room Details</legend>
4         <div class="centered-select">
5             <div class="inputbox"><?php include 'RoomListBox.php'; ?></div>
6         </div>
7         <!-- Grouping all input fields in one container -->
8         <div class="input-fields">
9             <div class="inputbox">
10                <label for="deleteRoomID">Room ID</label>
11                <input type="text" name="deleteRoomID" id="deleteRoomID" disabled><br><br>
12            </div>
13            <div class="inputbox">
14                <label for="deleteRoomType">Room Type</label>
15                <input type="text" name="deleteRoomType" id="deleteRoomType" disabled><br><br>
16            </div>
17            <div class="inputbox">
18                <label for="deleteCostPerNight">Cost Per Night</label>
19                <input type="text" name="deleteCostPerNight" id="deleteCostPerNight" disabled><br><br>
20            </div>
21            <div class="inputbox">
22                <label for="deleteSpecialNotes">Special Notes</label>
23                <input type="text" name="deleteSpecialNotes" id="deleteSpecialNotes" disabled><br><br>
24            </div>
25            <div class="inputbox">
26                <label for="deleteCurrentStatus">Current Status</label>
27                <input type="text" name="deleteCurrentStatus" id="deleteCurrentStatus" disabled><br><br>
28            </div>
29        </div>
30        <div class="myButton">
31            <input type="submit" value="Delete Room" id="delete" onclick="toggleLock()">
32            <input type="reset" value="Clear" name="reset" />
33        </div>
34    </fieldset>
35    <!-- Display the deletion message above the dropdown -->
36    <?php if ($deleteMessage) : ?>
37    <div class="delete-message">
38        <?php echo $deleteMessage; ?>
39    </div>
40    <?php endif; ?>
41
42    </div>
43    <script>
44        document.addEventListener('DOMContentLoaded', function () {
45            // ===== SIDEBAR =====
46            const burger = document.querySelector('.burger-menu');
47            const sidebar = document.querySelector('.sidebar');
48
49            // Toggle sidebar on burger click
50            burger.addEventListener('click', (e) => {
51                e.stopPropagation();
52                sidebar.classList.toggle('open');
53            });
54
55            // Close sidebar when clicking outside
56            document.addEventListener('click', (e) => {
57                if (!sidebar.contains(e.target) && !e.target.classList.contains('burger-menu')) {
58                    sidebar.classList.remove('open');
59                }
60            });
61
62            // ===== DROPPDOWNS =====
63            const dropdowns = document.querySelectorAll('.dropdown');
64            let activeDropdown = null;
65
66            dropdowns.forEach(dropdown => {
67                const content = dropdown.querySelector('.dropdown-content');
68
69                // Show dropdown on hover
70                dropdown.addEventListener('mouseenter', () => {
71                    if (activeDropdown) {
72                        activeDropdown.classList.remove('visible');
73                    }
74                    content.classList.add('visible');
75                    activeDropdown = content;
76                });
77
78                // Hide dropdown with delay
79                dropdown.addEventListener('mouseleave', () => {
80                    setTimeout(() => {
81                        if (!content.matches(':hover')) {
82                            content.classList.remove('visible');
83                            activeDropdown = null;
84                        }
85                    }, 10);
86                });
87
88                // Keep dropdown open when hovering its content
89                content.addEventListener('mouseenter', () => {
90                    clearTimeout();
91                    content.classList.add('visible');
92                });
93
94                // Hide dropdown when leaving its content
95                content.addEventListener('mouseleave', () => {
96                    content.classList.remove('visible');
97                    activeDropdown = null;
98                });
99            });
100        });
101    </script>
102 </body>
103 </html>

```

## Amend / View Page

The screenshot displays a web-based application for managing hotel rooms. On the left, there is a photograph of a room featuring a large floor lamp with a wide, shallow shade, a small round wooden side table with a white bowl, and a brown sofa with several pillows. On the right, another photograph shows a bed with blue and yellow patterned bedding. The central area contains a form titled "Amend/View Room" under the heading "Room Details". The form includes fields for "Room Type" (set to "double"), "Room ID" (empty), "Room Type" (empty), "Cost Per Night" (empty), "Address" (empty), "Current Status" (empty), and "Special Notes" (empty). At the bottom of the form are two buttons: "Save Changes" and "Amend Details". In the top right corner of the page header, there is a logo for "C2P HOTEL".

## AmendViewRoom.html.php

```

1  <!--
2   AmendViewRoom.html.php
3   Amend and change details of the Rooms
4   Szymon Stepiak
5   C00300283
6  -->
7  <!DOCTYPE html>
8  <html lang="en">
9
10 <head>
11   <meta charset="UTF-8">
12   <meta name="viewport" content="width=device-width, initial-scale=1.0">
13   <link rel="stylesheet" href="../../styles.css">
14   <link rel="website icon" value="png" href="../../C2PLogoNew.png">
15   <title>Amend/View Guest</title>
16 </head>
17
18 <body>
19   <!-- Burger icon that will toggle the sidebar -->
20   <div class="burger-menu">#9776;</div>
21   <div class="page-header">
22     <h1>Amend/View Room</h1>
23     <a href="../../home.html"></a>
24   </div>
25   <!-- Sidebar navigation -->
26   <?php include '../../sidebar.php'; ?>
27
28   <div class="content">
29     <script>
30
31       // Function to handle form submission
32       function confirmCheck() {
33         var response = confirm('Are these details correct?');
34         if (response) {
35           document.getElementById("amendRoomId").disabled = false;
36           document.getElementById("amendRoomType").disabled = false;
37           document.getElementById("amendCostPerNight").disabled = false;
38           document.getElementById("amendTeleExtension").disabled = false;
39           document.getElementById("amendSpecialNotes").disabled = false;
40           document.getElementById("amendCurrentStatus").disabled = false;
41           return true; // Validate before submitting
42         } else {
43           populate();
44           toggleLock();
45
46           return false;
47         }
48       }
49
50       // Function to toggle the lock/unlock on the form fields
51       function toggleLock() {
52         if (document.getElementById("amendView").value == "Amend Details") {
53           document.getElementById("amendRoomType").disabled = false;
54           document.getElementById("amendCostPerNight").disabled = false;
55           document.getElementById("amendTeleExtension").disabled = false;
56           document.getElementById("amendSpecialNotes").disabled = false;
57           document.getElementById("amendCurrentStatus").disabled = true;
58           document.getElementById("amendViewbutton").value = "View Details";
59         } else {
60
61           document.getElementById("amendRoomType").disabled = true;
62           document.getElementById("amendCostPerNight").disabled = true;
63           document.getElementById("amendTeleExtension").disabled = true;
64           document.getElementById("amendSpecialNotes").disabled = true;
65           document.getElementById("amendCurrentStatus").disabled = false;
66           document.getElementById("amendemail").disabled = true;
67           document.getElementById("amendViewbutton").value = "View Details";
68         }
69       }
70
71       // Function to populate fields
72       function populate() {
73         var sel = document.getElementById("listbox");
74         var result = sel.options[sel.selectedIndex].value;
75         var personDetails = result.split(',');
76
77         document.getElementById("amendRoomId").value = personDetails[0];
78         document.getElementById("amendRoomType").value = personDetails[1];
79         document.getElementById("amendCostPerNight").value = personDetails[2];
80         document.getElementById("amendTeleextension").value = personDetails[3];
81         document.getElementById("amendSpecialNotes").value = personDetails[4];
82         document.getElementById("amendCurrentStatus").value = personDetails[5];
83       }
84     </script>
85
86     <form action="amendViewRoom.php" onsubmit="confirmCheck()" method="post">
87       <fieldset>
88         <legend>Room Details</legend>
89         <div class="centered-select">
90           <div class="inputbox"><?php include 'RoomListBox.php'; ?></div>
91         </div>
92         <div class="columns">
93           <div class="inputbox">
94             <label for="amendRoomId">Room ID</label>
95             <input type="text" name="amendRoomId" id="amendRoomId" disabled><br><br>
96           </div>
97     </form>

```

```

1      <div class="inputbox">
2          <label for="amendRoomType">Room Type</label>
3          <input type="text" name="amendRoomType" id="amendRoomType" disabled><br><br>
4      </div>
5
6      <div class="inputbox">
7          <label for="amendCostPerNight">Cost Per Night</label>
8          <input type="text" name="amendCostPerNight" id="amendCostPerNight" disabled><br><br>
9      </div>
10     <div class="inputbox">
11         <label for="amendTeleExtension">Address</label>
12         <input type="text" name="amendTeleExtension" id="amendTeleExtension" disabled><br><br>
13     </div>
14
15     <div class="inputbox">
16         <label for="amendCurrentStatus">Current Status</label>
17         <input type="text" name="amendCurrentStatus" id="amendCurrentStatus" disabled><br><br>
18     </div>
19
20     <div class="inputbox">
21         <label for="amendSpecialNotes">Special Notes</label>
22         <input type="text" name="amendSpecialNotes" id="amendSpecialNotes" disabled><br><br>
23     </div>
24
25     <div class="myButton">
26         <input type="submit" value="Save Changes">
27         <input type="button" value="Amend Details" id="amendView" onclick="toggleLock()">
28     </div>
29
30     </fieldset>
31 </form>
32 </script>
33     document.addEventListener('DOMContentLoaded', function () {
34         // ===== SIDEBAR =====
35         const burger = document.querySelector('.burger-menu');
36         const sidebar = document.querySelector('.sidebar');
37
38         // Toggle sidebar on burger click
39         burger.addEventListener('click', (e) => {
40             e.stopPropagation();
41             sidebar.classList.toggle('open');
42         });
43
44         // Close sidebar when clicking outside
45         document.addEventListener('click', (e) => {
46             if (!sidebar.contains(e.target) && !e.target.classList.contains('burger-menu')) {
47                 sidebar.classList.remove('open');
48             }
49         });
50
51         // ===== DROPPDOWNS =====
52         const dropdowns = document.querySelectorAll('.dropdown');
53         let activeDropdown = null;
54
55         dropdowns.forEach(dropdown => {
56             const content = dropdown.querySelector('.dropdown-content');
57
58             // Show dropdown on hover
59             dropdown.addEventListener('mouseenter', () => {
60                 if (activeDropdown) {
61                     activeDropdown.classList.remove('visible');
62                 }
63                 content.classList.add('visible');
64                 activeDropdown = content;
65             });
66
67             // Hide dropdown with delay
68             dropdown.addEventListener('mouseleave', () => {
69                 setTimeout(() => {
70                     if (!content.matches(':hover')) {
71                         content.classList.remove('visible');
72                         activeDropdown = null;
73                     }
74                 }, 10);
75             });
76
77             // Keep dropdown open when hovering its content
78             content.addEventListener('mouseenter', () => {
79                 clearTimeout();
80                 content.classList.add('visible');
81             });
82
83             // Hide dropdown when leaving its content
84             content.addEventListener('mouseleave', () => {
85                 content.classList.remove('visible');
86                 activeDropdown = null;
87             });
88         });
89     });
90 </script>
91 </body>
92
93 </html>

```

## AmendViewRoom.php

```

1 <?php
2     include 'db.inc.php';
3     date_default_timezone_set('UTC');
4
5
6 $sql = "UPDATE Rooms SET Type ='$_POST[amendRoomType]',  

7 CostPerNight ='$_POST[amendCostPerNight]',  

8 TelephoneExtensionNumber ='$_POST[amendTeleExtension]',  

9 SpecialNotes ='$_POST[amendSpecialNotes]',  

10 CurrentStatus ='$_POST[amendCurrentStatus]'  

11 WHERE RoomID = '$_POST[amendRoomId]'";
12
13
14 if(!mysqli_query($con,$sql))
15 {
16     echo "Error".mysqli_error($con);
17 }
18 else
19 {
20     if(mysqli_affected_rows($con)>0)
21     {
22         echo mysqli_affected_rows($con)."Record(s) updated <br>";
23         echo "Room ID ". $_POST['amendRoomId'].  

24             " ".$_POST['amendRoomType'].  

25             " ".$_POST['amendCostPerNight'].  

26             " ".$_POST['amendTeleExtension'].  

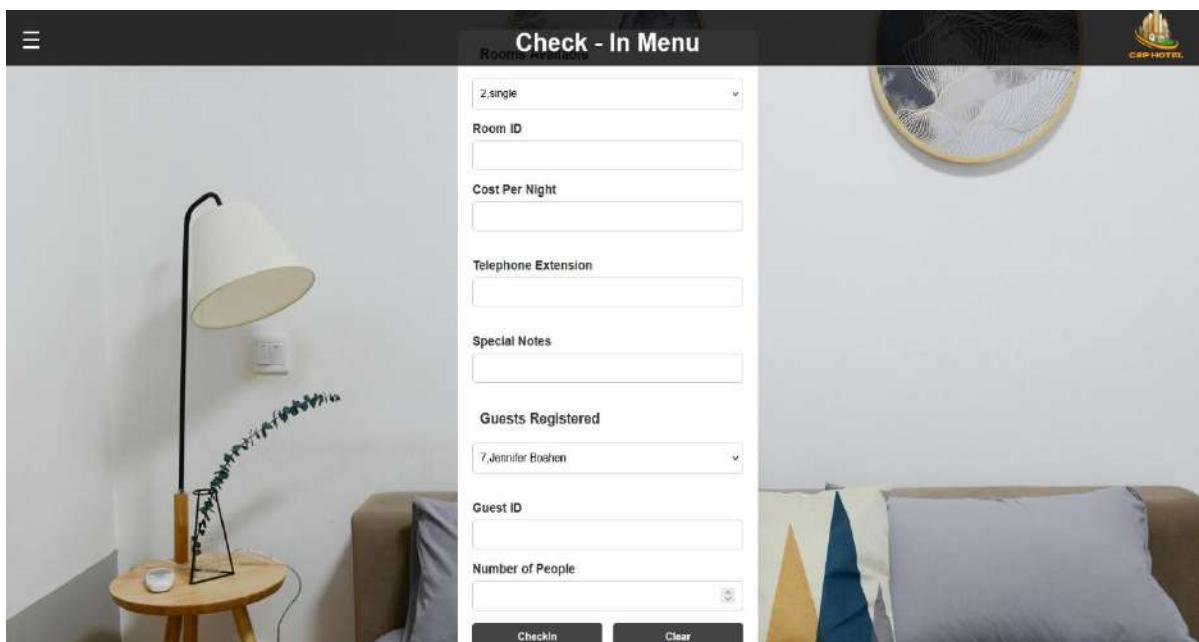
27             " ".$_POST['amendSpecialNotes'].  

28             " ".$_POST['amendCurrentStatus'].  

29             " ." Has been updated";
30     }
31     else
32     {
33         echo"No records were changed";
34     }
35 }
36 mysqli_close($con);
37 ?>
38 <form action ="AmendViewRoom.html.php" method="POST">
39 <input type="submit" value="Return to previous Screen">
40 </form>
41

```

## Check in menu



## Checkin.html.php

```

1  <!--
2   Checkin.html.php
3   This allows the user to select the name and the room they want to check into which is then stored in the GuestStays table
4   Szymon Stepiak
5   c00300283
6  -->
7  <!DOCTYPE html>
8  <html lang="en">
9  <head>
10    <meta charset="UTF-8">
11    <title>Check-in Menu</title>
12    <link rel="stylesheet" href="../styles.css">
13    <link rel="website icon" value="png" href="../C2PLogoNew.png">
14    <script src="checkinValidation.js"></script>
15  </head>
16  <body>
17    <!-- Burger icon that will toggle the sidebar -->
18    <div class="burger-menu">&#9776;</div>
19    <div class="page-header">
20      <h1>Check - In Menu</h1>
21      <a href="../home.html.php"></a>
22    </div>
23    <!-- inclusion of sidebar -->
24    <?php include '../sidebar.php'?>
25    <div class="content">
26      <form action="checkin.php" method="POST" onsubmit="return checkNumOfPPL();">
27        <legend>Rooms Available</legend>
28        <div class="columns">
29          <div class="left-column">
30            <div class="inputbox"><?php include 'RoomListBox.php';?></div>
31            <div class="inputbox">
32              <label for="amendRoomId">Room ID</label>
33              <input type="text" name="amendRoomId" id="amendRoomId" disabled>
34              <input type="hidden" name="amendRoomId" id="amendRoomIdHidden">
35            </div>
36            <div class="inputbox">
37              <label for="amendCostPerNight">Cost Per Night</label>
38              <input type="text" name="amendCostPerNight" id="amendCostPerNight" disabled><br><br>
39            </div>
40            <div class="inputbox">
41              <label for="amendTeleExtension">Telephone Extension</label>
42              <input type="text" name="amendTeleExtension" id="amendTeleExtension" disabled><br><br>
43            </div>
44            <div class="inputbox">
45              <label for="amendSpecialNotes">Special Notes</label>
46              <input type="text" name="amendSpecialNotes" id="amendSpecialNotes" disabled><br><br>
47            </div>
48          </div>
49          <div class="right-column">
50            <legend>Guests Registered</legend>
51            <div class="inputbox"><?php include 'checkinGuestBox.php';?></div><br>
52
53            <div class="inputbox">
54              <label for="guestID">Guest ID</label>
55              <input type="text" id="guestID" name="guestID" disabled>
56              <input type="hidden" id="guestIDHidden" name="guestID">
57            </div>
58            <div class="inputbox">
59              <label for="numberInGroup">Number of People</label>
60              <input type="number" name="numberInGroup"
61                  id="numberInGroup" min="1" max="5"
62                  value="minimum amount of people in a room is 1 and max amount of people in a room is 5"
63                  required/>
64            </div>
65          </div>
66        </div>
67        <div class="myButton">
68          <input type="submit" value="CheckIn" id="submit"/>
69          <input type="reset" value="Clear" name="reset" />
70        </div>
71      </form>
72      <script>
73        // Function to populate fields
74        function populate() {
75          var sel = document.getElementById("listbox");
76          var result = sel.options[sel.selectedIndex].value;
77          var personDetails = result.split(',');
78          document.getElementById("amendRoomId").value = personDetails[0];
79          document.getElementById("amendRoomIdHidden").value = personDetails[0];
80          document.getElementById("amendCostPerNight").value = personDetails[2];
81          document.getElementById("amendTeleExtension").value = personDetails[3];
82          document.getElementById("amendSpecialNotes").value = personDetails[4];
83        }
84
85        function populate1() {
86          var select = document.getElementById("listbox1");
87          var results = select.options[select.selectedIndex].value;
88          var guestDetails = results.split(',');
89          document.getElementById("guestID").value = guestDetails[0];
90          document.getElementById("guestIDHidden").value = guestDetails[0];
91        }
92      </script>

```

```

1 <script>
2     document.addEventListener('DOMContentLoaded', function () {
3         // ===== SIDEBAR =====
4         const burger = document.querySelector('.burger-menu');
5         const sidebar = document.querySelector('.sidebar');
6
7         // Toggle sidebar on burger click
8         burger.addEventListener('click', (e) => {
9             e.stopPropagation();
10            sidebar.classList.toggle('open');
11        });
12
13         // Close sidebar when clicking outside
14         document.addEventListener('click', (e) => {
15             if (!sidebar.contains(e.target) && !e.target.classList.contains('burger-menu')) {
16                 sidebar.classList.remove('open');
17             }
18         });
19
20         // ===== DROPSHEDS =====
21         const dropdowns = document.querySelectorAll('.dropdown');
22         let activeDropdown = null;
23
24         dropdowns.forEach(dropdown => {
25             const content = dropdown.querySelector('.dropdown-content');
26
27             // Show dropdown on hover
28             dropdown.addEventListener('mouseenter', () => {
29                 if (activeDropdown) {
30                     activeDropdown.classList.remove('visible');
31                 }
32                 content.classList.add('visible');
33                 activeDropdown = content;
34             });
35
36             // Hide dropdown with delay
37             dropdown.addEventListener('mouseleave', () => {
38                 setTimeout(() => {
39                     if (!content.matches(':hover')) {
40                         content.classList.remove('visible');
41                         activeDropdown = null;
42                     }
43                 }, 10);
44             });
45
46             // Keep dropdown open when hovering its content
47             content.addEventListener('mouseenter', () => {
48                 clearTimeout();
49                 content.classList.add('visible');
50             });
51
52             // Hide dropdown when leaving its content
53             content.addEventListener('mouseleave', () => {
54                 content.classList.remove('visible');
55                 activeDropdown = null;
56             });
57         });
58     });
59     </script>
60 </div>
61 </body>
62 </html>

```

## Checkin.php

```

1 <?php
2 //this inserts data necessary to checkin into the GuestStays table
3 //checks if the data in post are empty
4 //updates the room status to occupied with the room thats in post
5 //matches the GuestID of the guest in the guest table and the GuestStays table
6 //Szymon Stepnia
7 //c00300283
8 include 'db.inc.php';
9
10 // Ensure POST data is set before using it
11 $amendRoomId = $_POST['amendRoomId'] ?? null;
12 $guestID = $_POST['guestID'] ?? null;
13 $numberInGroup = $_POST['numberInGroup'] ?? null;
14
15 if ($amendRoomId === null) {
16     die("Error: Room ID is not set.");
17 }
18
19 if ($guestID === null) {
20     die("Error: Guest ID is not set.");
21 }
22
23 if ($numberInGroup === null) {
24     die("Error: Number of people is not set.");
25 }
26
27 // Get the current date for CheckInDate
28 $checkInDate = date("Y-m-d");
29
30 // Insert the guest stay into the GuestStays table
31 $sqlInsertGuestStay = "INSERT INTO GuestStays (GuestID, RoomID, CheckInDate, NumberInGroup)
32                         VALUES ('$guestID', '$amendRoomId', '$checkInDate', '$numberInGroup')";
33
34 if (!mysqli_query($con, $sqlInsertGuestStay)) {
35     die("Error inserting guest stay: " . mysqli_error($con));
36 }
37
38 // Update the room status to 'Occupied' where RoomID matches
39 $sqlroom = "UPDATE Rooms SET CurrentStatus='Occupied' WHERE RoomID='$amendRoomId'";
40
41 if (!mysqli_query($con, $sqlroom)) {
42     die("Error updating room status: " . mysqli_error($con));
43 }
44
45 // Get the guestID that matches the given guest from Guests table and ensure it's linked in GuestStays
46 $sqlguestStay = "SELECT Guests.guestID
47                  FROM Guests
48                  INNER JOIN GuestStays ON Guests.guestID = GuestStays.guestID
49                  WHERE Guests.guestID = '$guestID'";
50
51 $result = mysqli_query($con, $sqlguestStay);
52 if (!$result) {
53     die("Error fetching guest data: " . mysqli_error($con));
54 }
55
56 // Check if we got a valid guest ID
57 if (mysqli_num_rows($result) > 0) {
58     $row = mysqli_fetch_assoc($result);
59     $matchedGuestID = $row['guestID'];
60
61     echo "<br>Guest with ID $matchedGuestID has been checked into room $amendRoomId successfully!";
62 } else {
63     die("Error: No matching guest found in GuestStays.");
64 }
65
66 // Close the database connection
67 mysqli_close($con);
68 ?>
69
70 <!-- Check-in confirmation form -->
71 <form action="checkin.html.php" method="POST">
72     <input type="submit" value="Go Back">
73 </form>

```

## CheckinValidation.js

```

1 // Checks the input that is in the listbox based on the room type and validates for each type of room using switch statement
2 // Validates that a certain amount of persons can stay in a specific room using numberInGroup from POST
3 // Szymon Stepniak
4 // c00300283
5
6 // Function to validate the number of people based on the room type
7 function checkNumOfPPL() {
8     // Get the selected room details
9     var sel = document.getElementById("listbox");
10    var result = sel.options[sel.selectedIndex].value;
11    var roomDetails = result.split(',');
12
13    var roomType = roomDetails[1]; // Room type is at index 1
14    var numberInGroup = parseInt(document.getElementById("numberInGroup").value);
15
16    // Validation logic based on room type
17    switch (roomType) {
18        case "single":
19            if (numberInGroup > 1) {
20                alert("Error: A Single room can only accommodate 1 person.");
21                return false; // Prevent form submission
22            }
23            break;
24
25        case "double":
26        case "twin":
27            if (numberInGroup > 2) {
28                alert("Error: A " + roomType + " room can only accommodate 2 people.");
29                return false; // Prevent form submission
30            }
31            break;
32
33        case "family":
34            if (numberInGroup > 5) {
35                alert("Error: A Family room can only accommodate a maximum of 5 people.");
36                return false; // Prevent form submission
37            }
38            break;
39
40        default:
41            alert("Error: Invalid room type selected.");
42            return false; // Prevent form submission
43    }
44
45    if(!confirm("Are you sure all the details are correct?")){
46        return false;
47    }
48    return true; // Allow form submission
49 }
50
51 // Attach the validation function to the form's onsubmit event
52 document.querySelector("form").onsubmit = function () {
53     return checkNumOfPPL();
54 };
55

```