# CheckInOut: Database Documentation

## 1. Project Overview

CheckInOut is a database system designed for managing check-in and check-out activities for inter/intra hostel residents. It supports students, visitors, and staff, enabling efficient room allocation and mess tracking. The system ensures proper authentication, room availability status updates, and historical records of check-ins and check-outs.

## 2. Database Schema

## 2.1. Authentication Tables

### Users Table

|  |  |  |
| --- | --- | --- |
| Column | Type | Description |
| user\_id | INT, PRIMARY KEY, AUTO\_INCREMENT | Unique user identifier |
| username | VARCHAR(50), UNIQUE, NOT NULL | User's username |
| password\_hash | VARCHAR(255), NOT NULL | Hashed password |
| role | ENUM('Student', 'Visitor', 'Staff'), NOT NULL | User role for authentication |
| created\_at | TIMESTAMP, DEFAULT CURRENT\_TIMESTAMP | Account creation time |

## 2.2. Profile Tables

### Students Table

|  |  |  |
| --- | --- | --- |
| Column | Type | Description |
| roll\_no | VARCHAR(20), PRIMARY KEY, UNIQUE, NOT NULL | Student roll number |
| user\_id | INT, FOREIGN KEY REFERENCES Users(user\_id) | Link to authentication |
| name | VARCHAR(50), NOT NULL | Student's name |
| department | VARCHAR(50), NOT NULL | Department |
| gender | ENUM('Male', 'Female', 'Other'), NOT NULL | Gender |
| email | VARCHAR(20), UNIQUE, NOT NULL | Email address |
| hostel\_name | VARCHAR(30), NOT NULL  REFERENCES  Hostels Table(hostel\_name) | Allocated hostel |
| room\_no | VARCHAR(10), UNIQUE, NOT NULL  REFRENCES  HostelsRoom Table(room\_no) | Allocated room number |
| mess\_id | INT, FOREIGN KEY REFERENCES Mess(mess\_id) | Assigned mess |

### Visitors Table

|  |  |  |
| --- | --- | --- |
| Column | Type | Description |
| visitor\_id | INT, PRIMARY KEY, AUTO\_INCREMENT | Unique visitor identifier |
| user\_id | INT, FOREIGN KEY REFERENCES Users(user\_id) | Link to authentication |
| name | VARCHAR(50), NOT NULL | Visitor's name |
| contact\_no | INT(10), NOT NULL | Contact number |
| gender | ENUM('Male', 'Female', 'Other'), NOT NULL | Gender |
| room\_no | VARCHAR(10),NOT NULL | Allocated guest room |
| guesthouse\_name | VARCHAR(30), NOT NULL | Guesthouse assigned |
| duration\_of\_stay | INT, NOT NULL | Duration of stay in days |
| Purpose  Reference id | TEXT, NOT NULL  INT, FOREIGN KEY  REFERENCES  STUDENT, STAFF | Purpose of visit  References to someone in campus |

### Staff Table

|  |  |  |
| --- | --- | --- |
| Column | Type | Description |
| staff\_id | INT, PRIMARY KEY, AUTO\_INCREMENT | Unique staff identifier |
| user\_id | INT, FOREIGN KEY REFERENCES Users(user\_id) | Link to authentication |
| name | VARCHAR(50), NOT NULL | Staff name |
| department | VARCHAR(25), NOT NULL | Department |
| gender | ENUM('Male', 'Female', 'Other'), NOT NULL | Gender |
| contact\_no | VARCHAR(10), NOT NULL | Contact number |

## 2.3. Hostel and Guesthouse Information

### Hostels Table

|  |  |  |
| --- | --- | --- |
| Column | Type | Description |
| hostel\_name | VARCHAR(50), PRIMARY KEY | Unique hostel name |
| total\_rooms | INT, NOT NULL | Total number of rooms |
| location | VARCHAR(100), NOT NULL | Location of the hostel |

### Guesthouses Table

|  |  |  |
| --- | --- | --- |
| Column | Type | Description |
| guesthouse\_name | VARCHAR(50), PRIMARY KEY | Unique guesthouse name |
| total\_rooms | INT, NOT NULL | Total number of rooms |
| location | VARCHAR(100), NOT NULL | Location of the guesthouse |

## 2.4. Room Management

### HostelRooms Table

|  |  |  |
| --- | --- | --- |
| Column | Type | Description |
| room\_no | VARCHAR(20), PRIMARY KEY | Unique room number |
| hostel\_name | VARCHAR(50), NOT NULL, | Associated hostel |
| capacity | INT, NOT NULL | Maximum capacity |
| status | ENUM('Vacant', 'Occupied'), NOT NULL DEFAULT 'Vacant' | Room availability |

### GuestRooms Table

|  |  |  |
| --- | --- | --- |
| Column | Type | Description |
| room\_no | VARCHAR(10), PRIMARY KEY | Unique guestroom number |
| guesthouse\_name | VARCHAR(30), NOT NULL, | Associated guesthouse |
| capacity | INT, NOT NULL | Maximum capacity |
| type | VARCHAR(10), NOT NULL | Type of guesthouse (1BHK, 2BHK, etc.) |
| status | ENUM('Vacant', 'Occupied'), NOT NULL DEFAULT 'Vacant' | Room availability |

## 2.5. Mess and Check-in/Check-out Management

### Mess Table

|  |  |  |
| --- | --- | --- |
| Column | Type | Description |
| mess\_id | INT, PRIMARY KEY, AUTO\_INCREMENT | Unique mess identifier |
| mess\_name | VARCHAR(30), NOT NULL | Mess name |
| Capacity  location | INT, NOT NULL  VARCHAR(100) | Maximum capacity  Address of mess |

### HostelCheckInOut Table

|  |  |  |
| --- | --- | --- |
| Column | Type | Description |
| record\_id | INT, PRIMARY KEY, AUTO\_INCREMENT | Unique record identifier |
| user\_id | INT, FOREIGN KEY REFERENCES Student(roll\_no), Staff(staff\_id) | User checking in |
| room\_no | VARCHAR(20), NOT NULL  REFERENCES  HostelRooms (room\_no) | Room number |
| hostel\_name | VARCHAR(50), NOT NULL  REFERNCES  Hostels(hostel\_name) | Hostel name |
| checkin\_time | DATETIME, NOT NULL | Check-in time |
| checkout\_time | DATETIME, NOT NULL | Check-out time |

### GuestCheckInOut Table

|  |  |  |
| --- | --- | --- |
| Column | Type | Description |
| record\_id | INT, PRIMARY KEY, AUTO\_INCREMENT | Unique record identifier |
| user\_id | INT, FOREIGN KEY REFERENCES Users(user\_id) | User checking in |
| room\_no | VARCHAR(20), NOT NULL | Room number |
| Guesthouse\_name | VARCHAR(50), NOT NULL | Hostel name |
| checkin\_time | DATETIME, NOT NULL | Check-in time |
| checkout\_time | DATETIME, NULLABLE | Check-out time |

### MessCheckinCheckout Table

|  |  |  |
| --- | --- | --- |
| Column | Type | Description |
| record\_id | INT, PRIMARY KEY, AUTO\_INCREMENT | Unique record identifier |
| user\_id  user\_type | INT, NOT NULL  ENUM(‘Student’,’Staff’,  ‘Visitor’) NOT NULL | User ID  Role of mess visitor |
| mess\_id | INT, FOREIGN KEY REFERENCES Mess(mess\_id) | Mess checked into |
| checkin\_time | DATETIME, NOT NULL | Check-in time |
| Apply CHECK on user\_id = (‘roll\_no’,’Visitor\_id’,’Staff\_id’) |  |  |