YORKSHIRE INN BED AND BREAKFAST DEPLOYMENT PLAN

Team: Y Inn

Team members: Blendi Rrustemi, Sara Magas, Andrea Antunovic, Mario Stura, Lawrence Buljanovic, Orsat Mladosic & Denim Behn

Date: April 24th 2023

Year: 2023

Table of Content

Section 1: Requirements

This section details the requirements for a successful deployment.

1.1 Deployment Description

This project is a web based application, dependent on web technology such as PHP and MariaDB. As such, it has a very simple deployment process and will require little setup. The only dependencies to install are HTTP Apache Server and MariaDB Server. No dependencies, libraries, no installation, just migrate files to Apache server htdocs folder and populate the database. After installing the system, you need to follow to testing procedures

1.2 Installation Dependencies

1.2.1 Equipment

- Computer or modern digital device with capability of connection to internet
- Internet Access
- Server 24/7

1.2.2 Software

- Web Browser
- MariaDB Server
- Apache HTTP Server

Extra information:

- Apache HTTP Server: This is a web server software that allows you to host and serve web content. It is available for download from the Apache website.
- MariaDB Server: This is a relational database management system that is used to store and manage data. It is available for download from the MariaDB website.

1.3 Operational Dependencies

To ensure that the application runs smoothly in a production environment, you may need to consider the following operational dependencies:

- Adequate server resources: The server should have enough CPU, RAM, and storage to handle the expected traffic and data.
- Backup and recovery plan: In case of any data loss or system failure, there should be a backup and recovery plan in place to restore the system to its previous state. In the core folder of the project, there is a failover script for both site and database connection, which means if one fails, a second connection will be established to the backup server.
- Security measures: There should be considered to apply appropriate security measures, such as setting up firewalls and SSL certificates, to protect the system from unauthorized access and attacks.

Section 2: Installation Procedure

This section details the steps that one must follow for a successful deployment.

2.1 Installation Description

To install this application, you can follow these general steps:

1. Install and configure Apache HTTP Server, PHP 8.2 and MariaDB Server on your system.
2. Download the application files and copy them to the appropriate directory on the server.
3. Import the database in MariaDB.
4. If necessary, edit the configuration file to set up the database connection, ex. Folder path.
5. Test the application to ensure that it is functioning properly.

2.2 Rollout Timeline

We have 3 phases for rolling out the system entirely:

- Phase 1: System installation.
- Phase 2: Testing phase.
- Phase 3: Final checks and deploying to live usage.

2.3 Install tools

There are various tools available that can help with the installation and deployment of this application. Some examples include:

XAMPP: This is a pre-packaged software that includes Apache, MySQL, PHP, and other tools that can simplify the installation and configuration process.

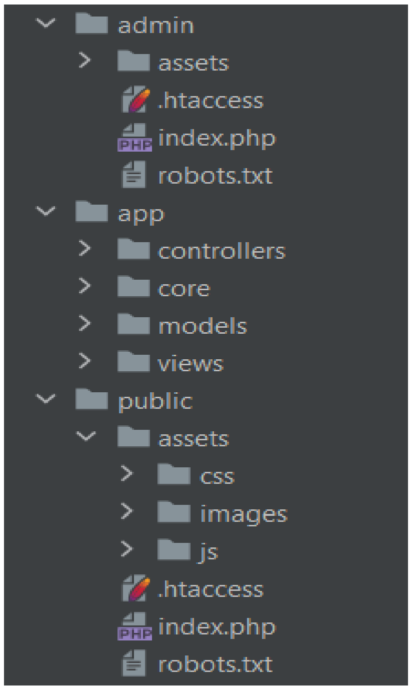
phpMyAdmin: This is a web-based tool that allows you to manage MySQL databases and tables through a web interface.

*Alternative option to phpMyAdmin: Any sort of database management application such as HeidiSQL or MySQL Workbench for editing and viewing the database.

Git: This is a version control system that can help with managing code changes and collaboration among team members.

2.4 Site configuration

We used an MVC architectural pattern which made the development process easier because of a well readable structure and therefore we made this project maintainable for future upgrades.



```
admin
    assets
    .htaccess
    index.php
    robots.txt
app
    controllers
    core
    models
    views
public
    assets
        css
        images
        js
    .htaccess
    index.php
    robots.txt
```

- admin folder - contains admin page and its components, which is accessible only with admin login

- App folder - contains controllers, models, views folders containing files that are part of MVC architecture, where the core folder is responsible for framework configuration files and its settings.
  - Models - deals with database CRUD
  - Views - view pages contain content, and it is displayed to the user. They also present data from the model.
  - Core - functionality scripts
  - Controller - updates model from view

- public folder - contains files that are hosted publicly in a root, therefore it is the only folder accessible to the web.
  - assets - contain javascript, css and images
  - index.php - main page where pages are routed and called upon
  - .htaccess - configuration file that allows modifying configuration without modifying main server configuration file
  - robots.txt - stops engine crawlers to accessing parts of project

Section 3: Post Installation

This section details the procedure post-installation for a successful deployment.

3.1 Testing

There are a few things that must be tested after the installation procedure is complete. They are listed as follows:

3.1.1 Links

Make sure the site navigates correctly by going to the following directories in the URL:

- Home: /home
- Login: /admin/login
- History: /history

3.1.2 Pages

Test each page to see that they work correctly while checking off the following:

- Spelling/grammar errors.
- Styling/graphical errors.
- Third-party website links
- Entity (such as the Google Map on the About page) errors.
- Session errors.
- Database query errors.

3.1.3 Login

Test the login functionality to ensure the following:

- Sessions work properly.
- User data is pulled and checked correctly from the database.

3.1.4 Admin

After logging in, test the Admin panel to ensure the following:

- The bookings and user data is being pulled and displayed in tables correctly.
- The editing of bookings and user data is being done correctly.
- The deleting of bookings and user data is being done correctly.

3.1.5 Security

Test the security of the website by trying the following:

- SQL injections
- Use ZAP to ‘attack’ the site and see feedback
- Latest version of PHP
- MariaDB server secure installation via script
- Chmod read write access configuration

3.2 Training

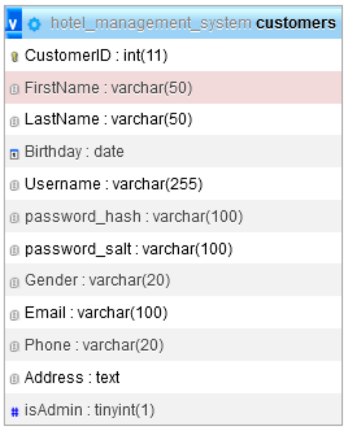
The team has prepared User and System documentation. With this it is possible to train any new employees to perform any role while using the website. There are detailed procedures and instructions on how to perform anything with the website and complete any task that it is capable of. Refer to those.

3.3 System Description

The system is a custom and pure PHP website that integrates with a MariaDB MySQL server to store data.

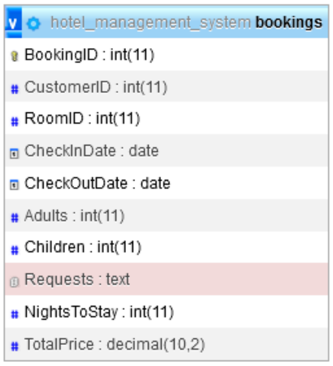
Here are some UML diagrams and tables to further describe the system and how it works:

UML diagram of the database tables:



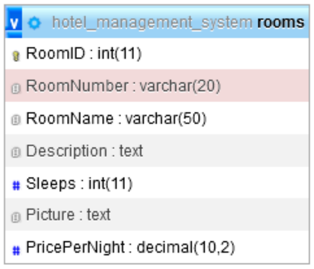
hotel_management_system **customers**

- CustomerID : int(11)
- FirstName : varchar(50)
- LastName : varchar(50)
- Birthday : date
- Username : varchar(255)
- password_hash : varchar(100)
- password_salt : varchar(100)
- Gender : varchar(20)
- Email : varchar(100)
- Phone : varchar(20)
- Address : text
- isAdmin : tinyint(1)



hotel_management_system **bookings**

- BookingID : int(11)
- CustomerID : int(11)
- RoomID : int(11)
- CheckInDate : date
- CheckOutDate : date
- Adults : int(11)
- Children : int(11)
- Requests : text
- NightsToStay : int(11)
- TotalPrice : decimal(10,2)



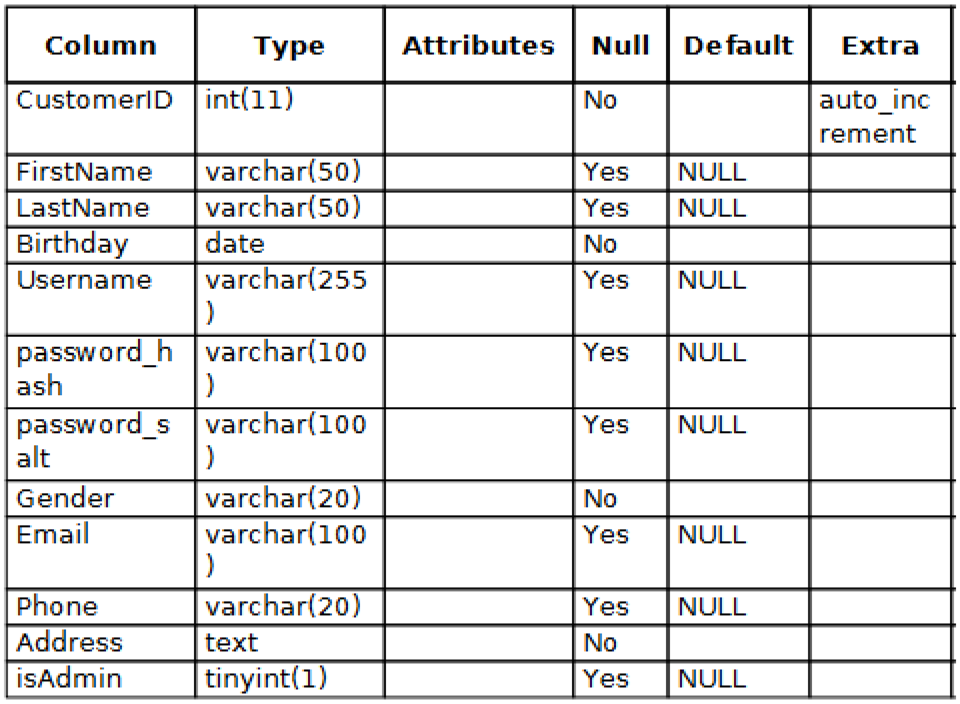
hotel_management_system **rooms**

- RoomID : int(11)
- RoomNumber : varchar(20)
- RoomName : varchar(50)
- Description : text
- Sleeps : int(11)
- Picture : text
- PricePerNight : decimal(10,2)

Database table: 'bookings'

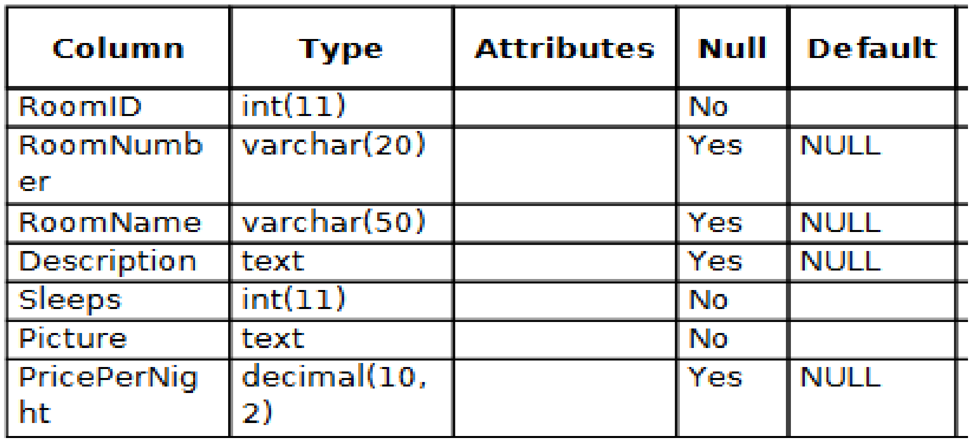
Column	Type	Attributes	Null	Default	Extra
BookingID	int(11)		No		auto_increment
CustomerID	int(11)		Yes	NULL	
RoomID	int(11)		Yes	NULL	
CheckInDate	date		Yes	NULL	
CheckOutDate	date		Yes	NULL	
Adults	int(11)		Yes	NULL	
Children	int(11)		Yes	NULL	
Requests	text		Yes	NULL	
NightsToStay	int(11)		Yes	NULL	
TotalPrice	decimal(10,2)		Yes	NULL	

Database table: ‘customers’



Column	Type	Attributes	Null	Default	Extra
CustomerID	int(11)		No		auto_increment
FirstName	varchar(50)		Yes	NULL	
LastName	varchar(50)		Yes	NULL	
Birthday	date		No		
Username	varchar(255)		Yes	NULL	
password_hash	varchar(100)		Yes	NULL	
password_salt	varchar(100)		Yes	NULL	
Gender	varchar(20)		No		
Email	varchar(100)		Yes	NULL	
Phone	varchar(20)		Yes	NULL	
Address	text		No		
isAdmin	tinyint(1)		Yes	NULL	

Database table: ‘rooms’



Column	Type	Attributes	Null	Default
RoomID	int(11)		No	
RoomNumber	varchar(20)		Yes	NULL
RoomName	varchar(50)		Yes	NULL
Description	text		Yes	NULL
Sleeps	int(11)		No	
Picture	text		No	
PricePerNight	decimal(10,2)		Yes	NULL

3.4 Transitioning from an Older System

When transitioning the Yorkshire Inn hotel to the new application, it is crucial to address compatibility concerns due to PHP 8.0+ with PDO usage. As PDO might be unsupported in older PHP versions, modify the code to utilize alternative database access methods or fallbacks, and test the application across different systems and PHP versions to resolve any compatibility issues.

Migrate relevant data, train staff on the new application, verify compatibility with existing hardware, software, and data formats, integrate the application with ongoing systems or processes, and strategically plan the rollout to minimize operational disruption.

3.5 Future Releases

We are not planning to release any future updates or versions of this application. Instead, we would like to focus on maintaining and supporting the existing codebase to ensure that it continues to function correctly and meets the needs of our users. This includes addressing any bugs or issues that arise, updating the application's security measures, optimizing its performance, providing user support, and maintaining up-to-date documentation. By doing so, we can ensure that the application remains valuable to the company and users.

3.6 Support Team and Support Procedures

As the project for the Yorkshire Inn hotel is now complete, we do not require a support team or support procedures for this application. We have provided detailed documentation with instructions on various operations, which can be used as a reference in case any issues arise. However, as we will not be providing ongoing support, any issues beyond the scope of the provided documentation may need to be resolved by the hotel staff or their hosting provider's technical support team.