

# **Appointment Booking System Installation & Usage**

By the LCJJ SE:PT Development Team



# Table of Contents

<b>Introduction</b>	<b>3</b>
<b>System Prerequisites</b>	<b>3</b>
Operating System	3
Command Prompt	3
Web Browser	4
PHP 7.0.1 +	4
Important	5
Composer	5
Copy of the Application	5
<b>Installation</b>	<b>6</b>
Moving Application	6
Installing Composer Dependencies	6
Creating an .env (environment) file	6
Generating a key	7
Migrate the Database	7
Starting a Server	8
<b>Usage</b>	<b>9</b>
Functionality	9
Customer	9
Business Owner (Administrator)	9
Requirements	9
Customer	10
Register	10
Login	11
View Bookings	12
Business Owner (Administrator)	13
Register	13
Login	14



Add Employee	15
Add Working Time	16
View Business Information	17
View Employee Availability	17
View Summary of Bookings	18
View History of Bookings	18
Assign Employee to Booking	19
Edit Working Time	19



# Introduction

The Appointment Booking System is a Software Engineering: Process & Tools project. The system is developed using PHP as the programming language with [Laravel](#) (a PHP framework).

This document will assume the reader has basic knowledge in operating a computer and basic web usage on the system.

Deploying this software for production use is acquired through running web hosting software ([Apache](#)) and a relational database management system ([MySQL](#), [PostgreSQL](#)).

For functionality sake, the document will go through the setup to use PHP (running a local server command) and SQLite (local database) to host on URL <http://localhost:8000>.

## System Prerequisites

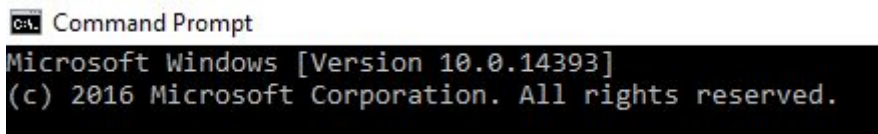
The follow application requires the proper environment to run the software.

### Operating System

It is recommended to use **Windows 10 Pro** as it has been tested and developed accordingly. Other operating systems may work, though configuration will be out reach for this documentation.

### Command Prompt

To use PHP or run installation commands, a terminal must be used that can run binary programs. Windows comes with command prompt by default and terminal for Mac OS.

A screenshot of the Windows Command Prompt window. The title bar reads "C:\> Command Prompt". The window content shows "Microsoft Windows [Version 10.0.14393]" and "(c) 2016 Microsoft Corporation. All rights reserved." on separate lines.

```
C:\> Command Prompt
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.
```

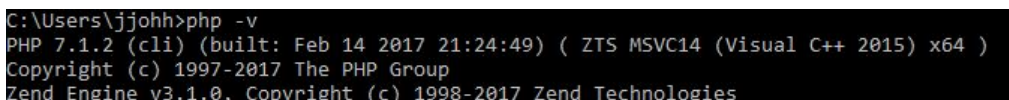
### Web Browser

Almost any latest common web browser can be used to visit the application webpages. Though the [Google Chrome browser](#) is recommended since it has been tested and developed with no functionality issues.

### PHP 7.0.1 +

System requires a [PHP binary package](#) with version 7.0.1 or later. Go through the [installation](#) provided by php.net.

To verify if PHP is running on your system, run the following command '**php -v**' on a command prompt.

A screenshot of a Command Prompt window showing the output of the 'php -v' command. The output displays the PHP version as 7.1.2 (cli) and includes build information, copyright details for The PHP Group and Zend Technologies, and the Zend Engine version v3.1.0.

```
C:\Users\jjohh>php -v
PHP 7.1.2 (cli) (built: Feb 14 2017 21:24:49) ( ZTS MSVC14 (Visual C++ 2015) x64 )
Copyright (c) 1997-2017 The PHP Group
Zend Engine v3.1.0, Copyright (c) 1998-2017 Zend Technologies
```





## Important

Once PHP is installed on the computer, the following PHP extensions must be enabled to allow the application to run properly.

- OpenSSL
- PDO
- Mbstring
- Tokenizer
- Sqlite3
- XML

Instructions to enable extensions for Windows is [here](#). Otherwise, a simple Google search on how to enable PHP extensions on other operating systems is relevant.

## Composer

Laravel relies on a PHP dependency manager to install third-party software such as Bootstrap (a CSS framework).

To verify that Composer is installed, run the following command **`composer -V`**.

```
C:\Users\jjohh>composer -V
Composer version 1.4.0 2017-03-08 17:51:24
```

## Copy of the Application

Make sure all files exist as shown on the image.

app	8/04/2017 12:54 AM	File folder	
bootstrap	15/03/2017 6:28 PM	File folder	
config	21/03/2017 1:22 AM	File folder	
database	8/04/2017 12:51 AM	File folder	
node_modules	17/03/2017 2:27 AM	File folder	
public	17/03/2017 2:27 AM	File folder	
resources	15/03/2017 6:28 PM	File folder	
routes	4/04/2017 9:20 PM	File folder	
storage	15/03/2017 6:28 PM	File folder	
tests	30/03/2017 4:41 PM	File folder	
vendor	8/04/2017 3:07 AM	File folder	
.env.example	23/03/2017 12:44 ...	EXAMPLE File	1 KB
.gitattributes	15/03/2017 6:28 PM	Text Document	1 KB
.gitignore	30/03/2017 2:04 PM	GITIGNORE File	1 KB
artisan	15/03/2017 6:28 PM	File	2 KB
composer.json	22/03/2017 9:33 PM	JSON File	2 KB
composer.lock	27/03/2017 12:42 ...	LOCK File	123 KB
debug.log	5/04/2017 5:13 PM	Text Document	2 KB
package.json	17/03/2017 2:26 AM	JSON File	2 KB
phpunit.dusk.xml	25/03/2017 2:59 PM	XML Document	1 KB
phpunit.xml	30/03/2017 8:51 PM	XML Document	2 KB
README.md	8/04/2017 12:54 AM	MD File	5 KB
server.php	15/03/2017 6:28 PM	PHP File	1 KB
unit.bat	21/03/2017 4:14 PM	Windows Batch File	1 KB
webpack.mix.js	15/03/2017 6:28 PM	JS File	1 KB
yarn.lock	15/03/2017 6:28 PM	LOCK File	164 KB



# Installation

## Moving Application

Simply place the application files in an accessible place on the local computer with proper read, write and execute permissions.

## Installing Composer Dependencies

Open a command prompt and change directory to the application folder.

Run the following command ***composer install*** to install required dependencies.

```
C:\Users\jjohh>composer install
```

## Creating an .env (environment) file

Laravel has predefined configurations that can be modified to suit different working environments. Whether that is a development, testing or production environment with the framework.

The required configuration needed is a production environment. To do so, there is a ***'env.example'*** file located at the root of the application folder.

Rename the file from ***'env.example'*** to ***'env'***. This let Laravel know that there is a user defined configuration for the environment.

```
1 APP_ENV=production
2 APP_KEY=
3 APP_DEBUG=true
4 APP_LOG_LEVEL=debug
5 APP_URL=http://localhost:8000
6
7 DB_CONNECTION=sqlite
8 DB_HOST=127.0.0.1
9 DB_PORT=3306
10 DB_USERNAME=root
11 DB_PASSWORD=
12
13 BROADCAST_DRIVER=log
14 CACHE_DRIVER=file
15 SESSION_DRIVER=file
16 QUEUE_DRIVER=sync
17
18 REDIS_HOST=127.0.0.1
19 REDIS_PASSWORD=null
20 REDIS_PORT=6379
21
22 MAIL_DRIVER=smtp
23 MAIL_HOST=mailtrap.io
24 MAIL_PORT=2525
25 MAIL_USERNAME=null
26 MAIL_PASSWORD=null
27 MAIL_ENCRYPTION=null
28
29 PUSHER_APP_ID=
30 PUSHER_APP_KEY=
31 PUSHER_APP_SECRET=
```

## Generating a key

The Laravel PHP framework needs a key to be generated before usage. Run the following command ***'php artisan key:generate'*** at the root of the application.



```
C:\Users\Johnny>php artisan key:generate
```

## Migrate the Database

Run the following command '**php artisan migrate:refresh**' to migrate database schemas (create tables in the database).

```
C:\Users\jjohh\projects\git\SE-PT-Assignment>php artisan migrate
Migration table created successfully.
Migrating: 2017_03_19_081558_create_business_owners_table
Migrated: 2017_03_19_081558_create_business_owners_table
Migrating: 2017_03_19_081613_create_bookings_table
Migrated: 2017_03_19_081613_create_bookings_table
Migrating: 2017_03_19_081620_create_customers_table
Migrated: 2017_03_19_081620_create_customers_table
Migrating: 2017_03_19_083138_create_employees_table
Migrated: 2017_03_19_083138_create_employees_table
Migrating: 2017_03_19_084917_create_availabilities_table
Migrated: 2017_03_19_084917_create_availabilities_table
Migrating: 2017_04_05_114720_create_working_times_table
Migrated: 2017_04_05_114720_create_working_times_table
```

Ignore the timestamp prefix on the files, they do not have to be exactly the same as output shown above.

## Starting a Server

Run the following command '**php artisan serve**' to start a local host server through command prompt.

```
C:\Users\jjohh\projects\git\SE-PT-Assignment>php artisan serve
Laravel development server started: <http://127.0.0.1:8000>
```

The application is now ready to be used through <http://127.0.0.1:8000> or <http://localhost:8000>.

The screenshot shows a web application interface for a 'Business Placeholder' booking system. At the top, the title 'Business Placeholder' is displayed in blue, with the subtitle 'Booking System: Login' below it. A light blue banner at the top of the login area reads 'No Business Found! Register your business here' with a close button. The login form consists of two input fields: 'Username' and 'Password', each with a placeholder text and a small icon on the right. Below these fields is a blue 'Sign in' button. At the bottom of the form is a blue 'Register' button. The footer of the page reads 'LCJJ Development Team'.



# Usage

This section will go through all functionality of the Appointment Booking System application.

The following instructions will assume that the application URL is <http://localhost:8000> and when told to visit a certain page given its route (e.g. /login) represents the URL <http://localhost:8000/login>.

## Functionality

The application can do the following actions organised by the role of the user of either customer or business owner.

### Customer

User is to not do any administrator actions and only be able to do the following:

- Login / Register as customer
- Add Booking without employee
- View Customer Bookings
- Logout

### Business Owner (Administrator)

User is able to do all actions provided by the application (super user). This includes customer actions.

- Login / Register as business owner (includes registering the business)
- Add Employee
- Add Working Time
- Edit Working Time
- Add Booking
- Add Activity
- Assign Employee to Booking
- View Business Information
- View Employee Availability
- View a Summary of Bookings
- View a History of Bookings
- Logout

## Requirements

The only software needed to use the application is a [web browser](#) as started in the software prerequisites.





## Customer

### Register

Visit the /register page and fill customer details in the input form.

### Business Placeholder

Booking System: Customer Registration

First Name

First Name

Last Name

Last Name

Username (alpha-numeric characters only)

Username

Password (at least 6 characters)

Password

Password Confirmation

Password

Phone (at least 10 characters)

Phone

Address (at least 6 characters)

Address

Register

LCJJ Development Team

Click the 'Register' button to submit the customer registration form. Once the form has been successfully submitted and validated, a redirection to /login and a message shown stating a successful customer registration.

Thank you for registering! You can now Login!

Username

Username

Password

Password

Sign in

Register

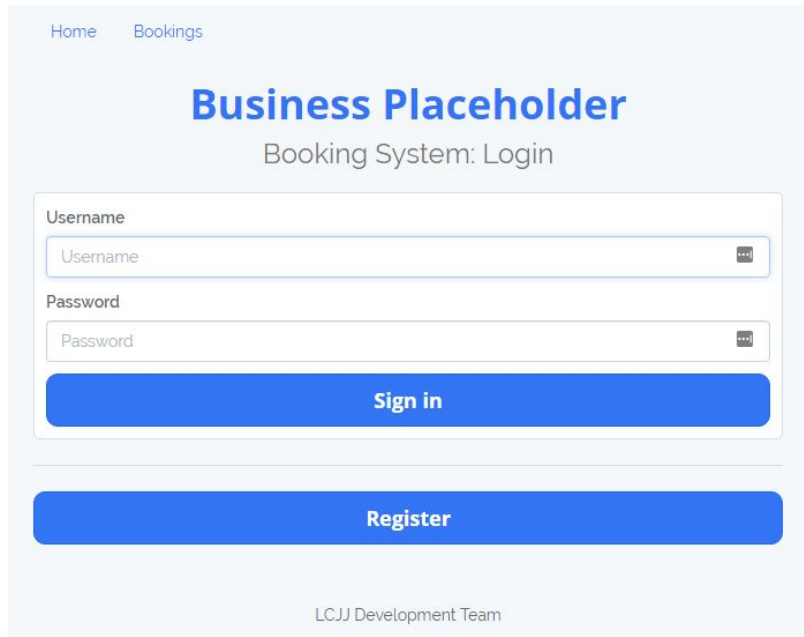
LCJJ Development Team



## Login

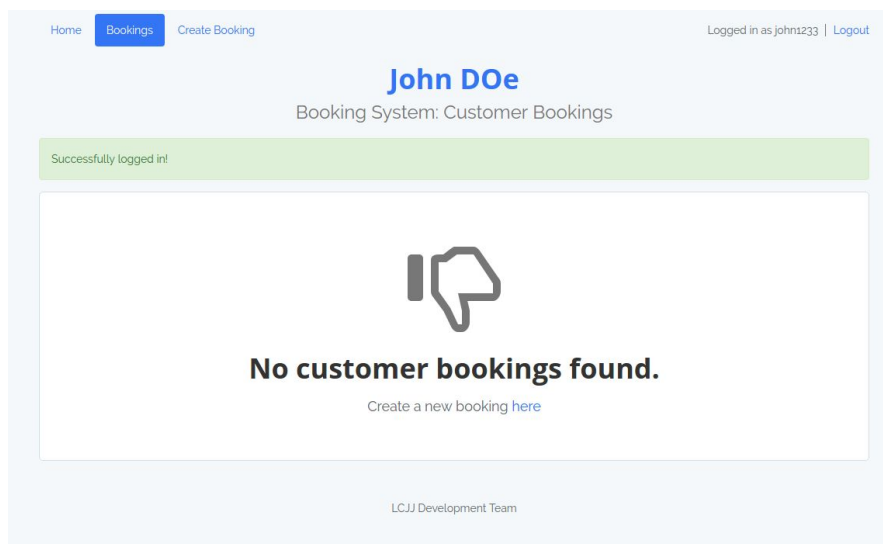
Visit the /login page or root of the application (<http://localhost:8000/> will redirect to /login). There will be an input form displayed with fields username and password.

It is important that a customer username and password must exist before logging in. Follow the customer registration section to create a new customer.



The screenshot shows the login page of the 'Business Placeholder' Booking System. At the top, there are navigation links for 'Home' and 'Bookings'. The main heading is 'Business Placeholder' in blue, followed by the subtitle 'Booking System: Login'. Below this is a login form with two input fields: 'Username' and 'Password', each with a placeholder text of the same name and a small eye icon to toggle visibility. A blue 'Sign in' button is positioned below the password field. Below the form is a blue 'Register' button. At the bottom of the page, it says 'LCJJ Development Team'.

Once the login form is filled, click the 'Sign in' button and the page will redirect to /bookings if the login was successful.



The screenshot shows the 'Customer Bookings' page after a successful login. The user is logged in as 'John DOe', with a 'Logout' link next to the name. The page title is 'John DOe' and the subtitle is 'Booking System: Customer Bookings'. A green notification bar at the top says 'Successfully logged in!'. The main content area features a large thumbs-down icon and the text 'No customer bookings found.' Below this, there is a link to 'Create a new booking here'. At the bottom, it says 'LCJJ Development Team'.



## View Bookings

Visit /bookings to view all customer bookings.

For part A of the assignment, dummy data is provided to show that bookings exist. This will not be the case once the assignment is complete.

Home

Bookings

Logged in as john123 | Logout

Business Placeholder

Booking System: Customer Bookings

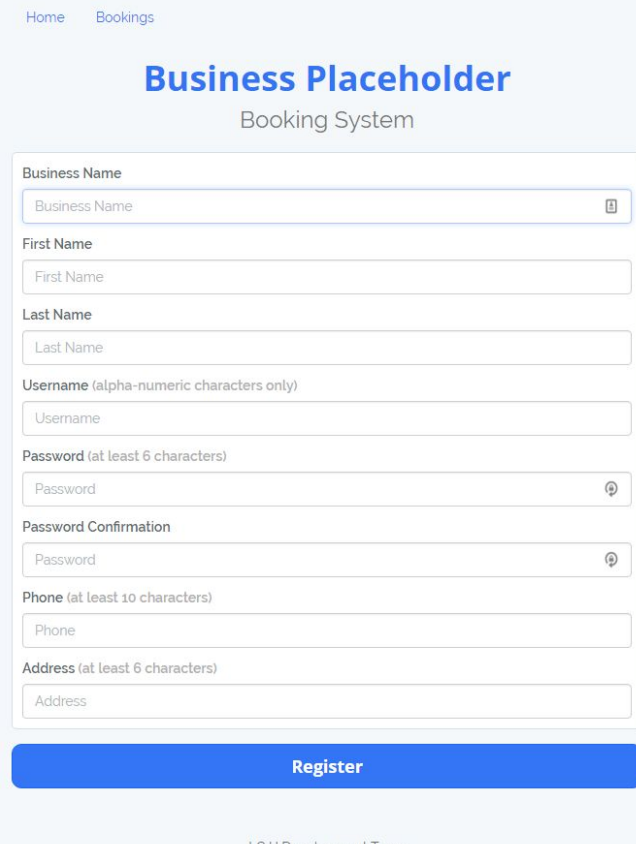
ID	Start Time	End Time	Date	Duration
259	04:00 AM	07:30 PM	11/04/17	15:30
568	04:30 AM	06:30 PM	12/10/16	14:00
86	02:00 AM	08:00 PM	11/04/17	18:00
48	06:00 PM	10:30 PM	04/03/18	4:30
474	02:00 AM	03:00 AM	12/12/16	1:00
422	02:00 AM	04:00 AM	31/07/16	2:00
353	06:00 AM	08:00 PM	04/07/16	14:00
181	11:00 AM	09:30 PM	14/10/17	10:30
677	04:00 AM	10:30 AM	12/04/17	6:30
912	04:00 PM	06:00 PM	09/01/17	2:00
10	01:00 PM	12:00 AM	11/01/18	13:00
318	12:00 AM	01:00 AM	20/06/17	1:00
654	01:30 PM	02:30 PM	22/09/17	1:00
136	06:00 AM	08:30 AM	17/08/16	2:30
803	08:30 PM	11:30 PM	20/01/17	3:00



## Business Owner (Administrator)

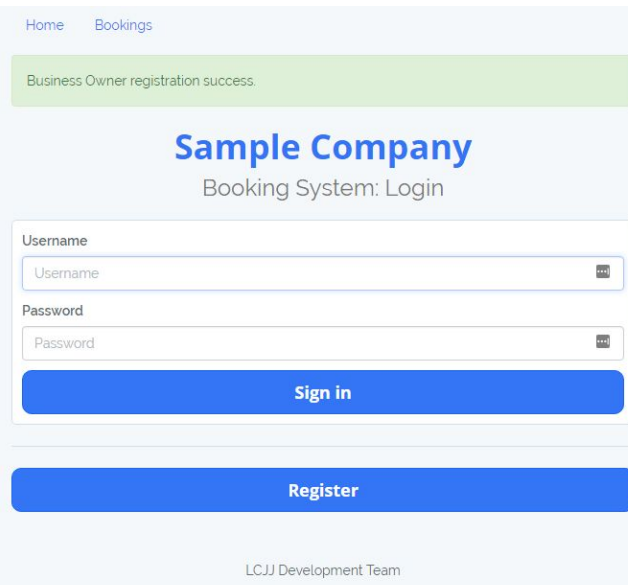
### Register

Visit the /admin/register page and fill business owner details in the input form.



The registration form is titled "Business Placeholder" with the subtitle "Booking System". It contains several input fields: "Business Name" (with a small icon), "First Name", "Last Name", "Username (alpha-numeric characters only)", "Password (at least 6 characters)", "Password Confirmation", "Phone (at least 10 characters)", and "Address (at least 6 characters)". Each field has a placeholder text. Below the fields is a large blue "Register" button. At the bottom, it says "LCJJ Development Team".

Click the 'Register' button to submit the business owner registration form. Once the form has been successfully submitted and validated, a redirection to /login and a message shown stating a successful business owner registration including the business itself.



The login and register form is titled "Sample Company" with the subtitle "Booking System: Login". It contains two input fields: "Username" and "Password", both with placeholder text and small icons. Below these fields is a large blue "Sign in" button. Below the "Sign in" button is a large blue "Register" button. At the bottom, it says "LCJJ Development Team". A green message box at the top says "Business Owner registration success."

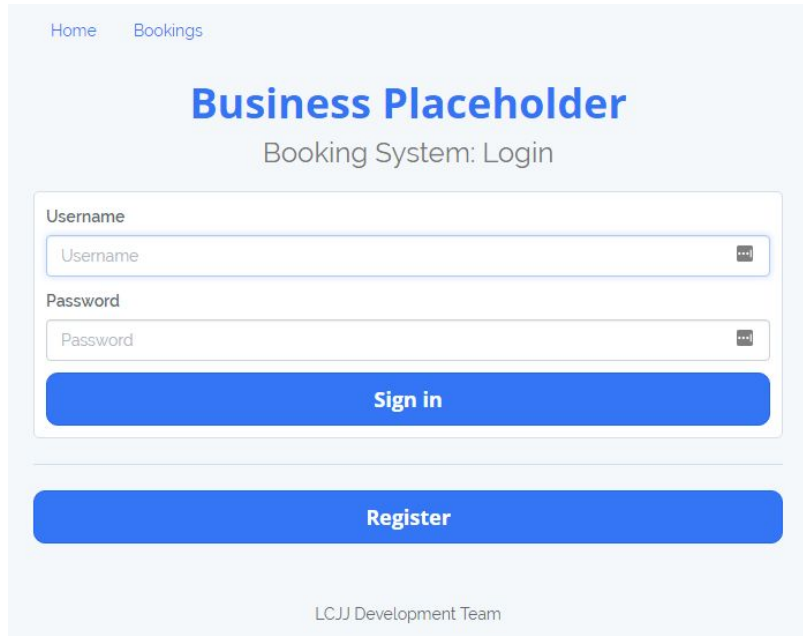




## Login

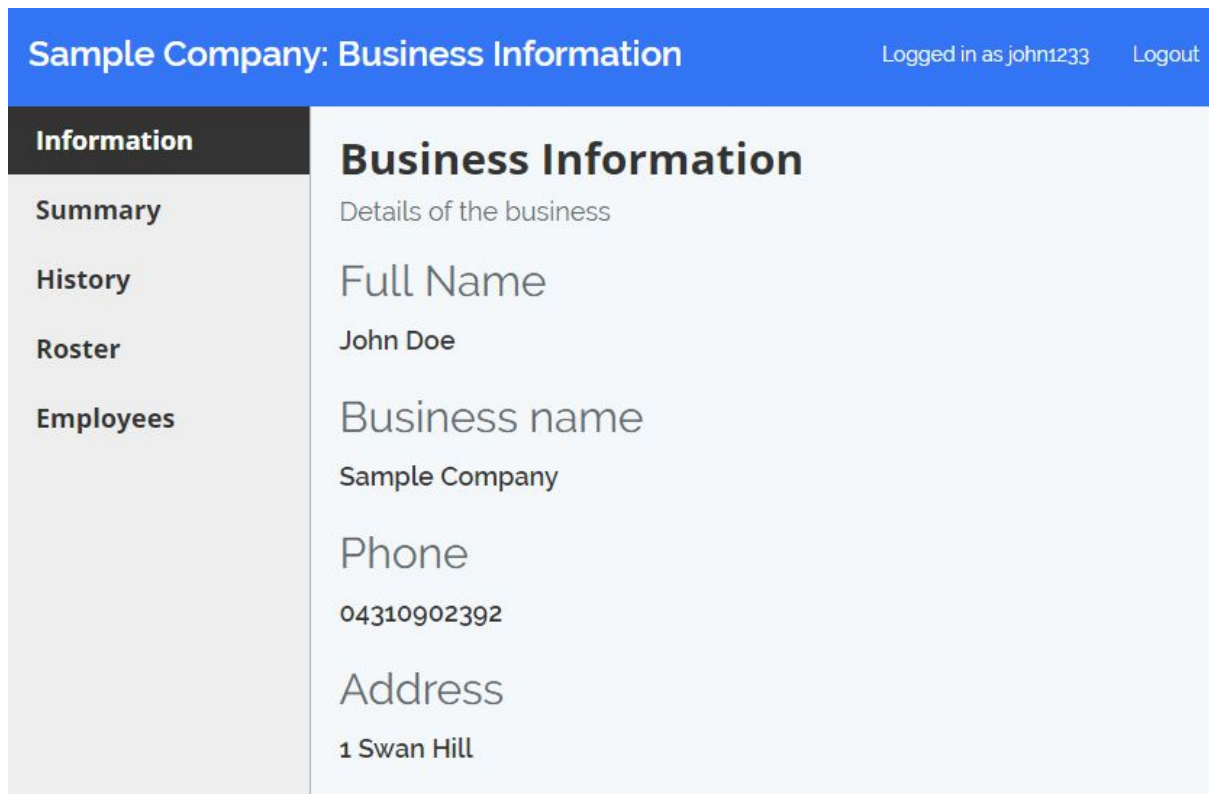
Visit the /login page or root of the application (<http://localhost:8000/> will redirect to /login). There will be an input form displayed with fields username and password.

It is important that a business owner username and password must exist before logging in. Follow the business owner registration section to create a new business owner.



The screenshot shows a web application interface for a 'Business Placeholder' booking system. At the top, there are navigation links for 'Home' and 'Bookings'. The main heading is 'Business Placeholder' in blue, followed by the subtitle 'Booking System: Login'. Below this is a login form with two input fields: 'Username' and 'Password', each with a placeholder text and a toggle icon. A blue 'Sign in' button is positioned below the password field. Below the login form is a blue 'Register' button. At the bottom of the page, it says 'LCJJ Development Team'.

Once the login form is filled, click the 'Sign in' button and the page will redirect to /admin if the login was successful.



The screenshot shows a web application interface for 'Sample Company: Business Information'. The top navigation bar is blue and contains the text 'Sample Company: Business Information' on the left, 'Logged in as john1233' in the center, and 'Logout' on the right. Below the navigation bar is a sidebar with a dark blue header 'Information' and four menu items: 'Summary', 'History', 'Roster', and 'Employees'. The main content area has a light blue background and a dark blue header 'Business Information' with the subtitle 'Details of the business'. The main content area displays the following information: 'Full Name' (John Doe), 'Business name' (Sample Company), 'Phone' (04310902392), and 'Address' (1 Swan Hill).



## Add Employee

Visit the /admin/employees page to add employees. Business owner must be registered and logged in to do the following actions.

Fill out the form provided on the page and click the 'Add Employee' button to submit the form.

Sample Company: Employees

Logged in as john1233Logout

Information

Summary

History

Roster

Employees

## Add Employee

Add a new employee to the system

Job Title


First Name

Last Name

Phone (at least 10 characters)

Add Employee

## Employees



### No employees found.

Try add an employee using the form above.

Once the form has been submitted and the input is valid, a new employee is created.

Sample Company: Employees

Logged in as john1233Logout

Information

Summary

History

Roster

Employees

## Add Employee

Add a new employee to the system

New Employee Added

Job Title

First Name

Last Name

Phone (at least 10 characters)

Add Employee

## Employees

A table of all employees within the business.

ID	First Name	Last Name	Title	Date Created
1	John	Doe	Job	08/04/17



## Add Working Time

Visit the /admin/roster page to add working time. Business owner must be registered and logged in to do the following actions. An employee must exist to create a working time.

Fill out the form provided on the page and click the 'Add Working Time' button to submit the form.

Once the form has been submitted and the input is valid, a new working time is created.

### Add Working Times

Add Business Hours for the next month.

Employee (ID - Title - Full Name)

1 - Cleaner - John Doe

Start Time (24 hour format)

09:00

End Time (24 hour format)

17:00

Date (dd/mm/yyyy)

01/04/2017

Add Working Time

### Roster

Show the roster of a given month.

Date (Select a dropdown item to go to month)

April 2017

April 2017

Week	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
1						01	02

April 2017

Week	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
1						John Doe Cleaner 09:00 AM - 05:00 PM	01 02
2	03	04	05	06	07	08	09
3	10	11	12	13	14	15	16
4	17	18	19	20	21	22	23
5	24	25	26	27	28	29	30



## View Business Information

Visit the /admin page to view business information. Business owner must be registered and logged in.

**Sample Company: Business Information**Logged in as john1233Logout

**Information**  
**Summary**  
**History**  
**Roster**  
**Employees**

## Business Information

Details of the business

**Full Name**  
John Doe

**Business name**  
Sample Company

**Phone**  
04310902392

**Address**  
1 Swan Hill

## View Employee Availability

Visit the /admin/summary page to view employee availability. Business owner must be registered and logged in.

**Sample Company: Summary of Bookings**Logged in as john1233Logout

**Information**  
**Summary**  
**History**  
**Roster**  
**Employees**

## Summary of Bookings

Present latest bookings for the next 7 days.

ID	Customer	Start	End	Date	Duration
3	Shayna Beer	00:30	22:00	09/04/17	21:30
4	Celestino Jerde	09:00	19:00	09/04/17	10:00
5	Reynold Kilback	08:30	14:00	10/04/17	5:30
6	Allan Cartwright	01:30	20:30	10/04/17	19:00
7	Jayson Halvorson	00:30	15:30	13/04/17	15:00
9	Nathanael Schroeder	00:00	03:00	13/04/17	3:00
8	Cristina Effertz	17:30	19:00	14/04/17	1:30

## Employee Availability

Show all employee availability for the next 7 days.

Employee Name	Sun 09/04	Mon 10/04	Tue 11/04	Wed 12/04	Thu 13/04	Fri 14/04	Sat 15/04
John Doe	Not Available	Not Available	Not Available	Not Available	Not Available	Not Available	Not Available





## View Summary of Bookings

Visit the /admin/summary page to view employee availability. Business owner must be registered and logged in. This action is similar to viewing employee availability since both functions are on the same page.

### Summary of Bookings

Present latest bookings for the next 7 days.

ID	Customer	Employee	Activity	Start	End	Duration	Date
6	John Do	John Doe	Clean	09:00	11:00	02:00	01/05/17
8	John Do	John Doe	Clean	13:00	15:00	02:00	01/05/17
9	John Do	So Sam	Clean	09:00	11:00	02:00	02/05/17
10	John Do	So Sam	Clean	15:00	17:00	02:00	02/05/17

## View History of Bookings

Visit the /admin/history page to view employee availability. Business owner must be registered and logged in.

### History of Bookings

Present older bookings.

ID	Customer	Employee	Activity	Start	End	Duration	Date
3	John Do	Unassigned	Clean	09:00	11:00	02:00	02/04/17
4	John Do	Unassigned	Clean	09:00	11:00	02:00	02/04/17
1	John Do	So Sam	Clean	09:00	11:00	02:00	01/04/17
2	John Do	Unassigned	Clean	09:00	11:00	02:00	01/04/17
5	John Do	Unassigned	Clean	09:00	11:00	02:00	01/04/17
7	John Do	John Doe	Clean	11:00	13:00	02:00	01/04/17



## Assign Employee to Booking

Visit the /admin/employees/assign page to assign an employee to a booking.

Use the checkboxes to select a booking to assign the selected employee above the table.

### Assign Employee

Assign an employee to a particular booking

Booking(s) have been successfully assigned.

Employee (Title - Full Name - ID)

Flyer - So Sam - 2

Select which bookings to assign the employee to (only bookings which the employee is available to work are shown)

Select	Customer	Start	End	Date	Duration	Employee Assigned
<input type="checkbox"/>	John Do	09:00	11:00	02/05/17	2:00	So Sam
<input type="checkbox"/>	John Do	15:00	17:00	02/05/17	2:00	So Sam

**Assign Employee**

### All Unassigned Bookings

Present all unassigned bookings.

Warning! Employee must be working during a booking time. Add a working time [here](#).

Customer	Activity	Start	End	Date	Duration
John Do	Clean	09:00	11:00	01/04/17	02:00
John Do	Clean	09:00	11:00	02/04/17	02:00

A table of unassigned bookings will assist you in missing bookings that a customer/admin submits without an employee.

### Assign Employee

Assign an employee to a particular booking. All available bookings are display for the next 30 days.


Employee (Title - Full Name - ID)

Cleaner - John Doe - 1

Select which bookings to assign the employee to (only bookings which the employee is available to work are shown)

Select	Customer	Start	End	Date	Duration	Employee Assigned
<input type="checkbox"/>	John Do	09:00	11:00	01/05/17	2:00	John Doe
<input type="checkbox"/>	John Do	13:00	15:00	01/05/17	2:00	John Doe

**Assign Employee**



**No unassigned bookings found.**

Congratulations! There are no more unassigned bookings for the next 30 days

## Edit Working Time

Visit the /admin/roster/{id}/edit page to edit an employee working time. The {id} in the URL



selects the working time ID from the database.

**Edit Working Time**  
Add Business Hours for the next month.

**Warning!** Once you have edited the working time of an employee, all working time bookings will be unassigned.

Employee (ID - Title - Full Name)  
1 - Cleaner - John Doe

Start Time (24 hour format) 09:00 x ↕

End Time (24 hour format) 17:00

Date (dd/mm/yyyy)  
01/05/2017

**Add Working Time**

Notice the warning show on the image above. Once the working time has been changed, all assigned bookings are unassigned. This will safely make all employees working on proper times.

