Appointment Booking System Installation & Usage

By the LCJJ SE:PT Development Team

Table of Contents

Introduction **3**

System Prerequisites **3**

Operating System 3

Command Prompt 3

Web Browser 4

PHP 7.0.1 + 4

Important 5

Composer 5

Copy of the Application 5

Installation **6**

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

Usage **9**

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

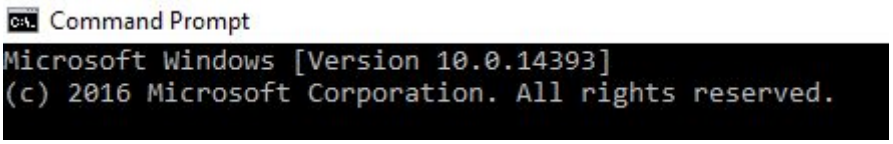
The follow application requires the proper environment to run the software.

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.



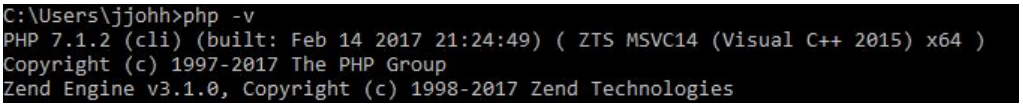
Web Browser

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.



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

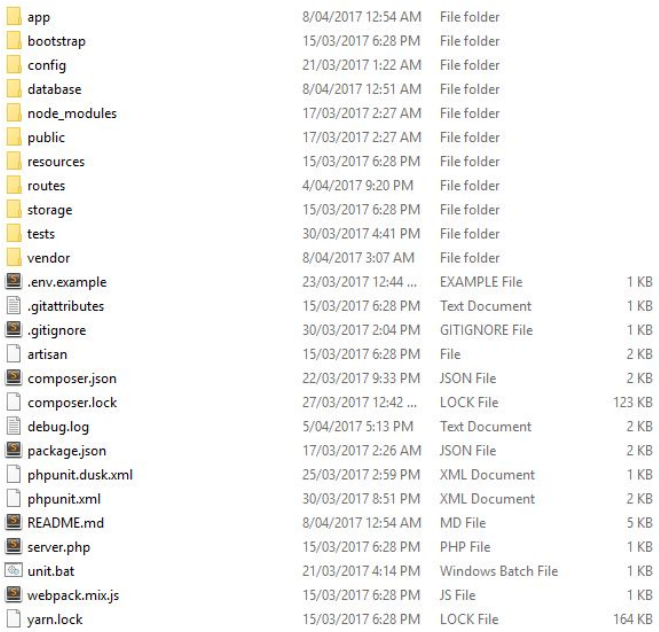
Laravel relies on a PHP dependency manager to install third-party software such as Bootstrap (a CSS framework).

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.

```
APP_ENV=production
APP_KEY=
APP_DEBUG=true
APP_LOG_LEVEL=debug
APP_URL=http://localhost:8000

DB_CONNECTION=sqlite
DB_HOST=127.0.0.1
DB_PORT=3306
DB_USERNAME=root
DB_PASSWORD=

BROADCAST_DRIVER=log
CACHE_DRIVER=file
SESSION_DRIVER=file
QUEUE_DRIVER=sync

REDIS_HOST=127.0.0.1
REDIS_PASSWORD=null
REDIS_PORT=6379

MAIL_DRIVER=smtp
MAIL_HOST=mailtrap.io
MAIL_PORT=2525
MAIL_USERNAME=null
MAIL_PASSWORD=null
MAIL_ENCRYPTION=null

PUSHER_APP_ID=
PUSHER_APP_KEY=
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 aritsan key:generate
```

Migrate the Database

Run the following command `***php artisan migrate:refresh***` to migrate database schemas (create tables in the database). Using the `**--seed**` flag is also useful to add predefined items to 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
```

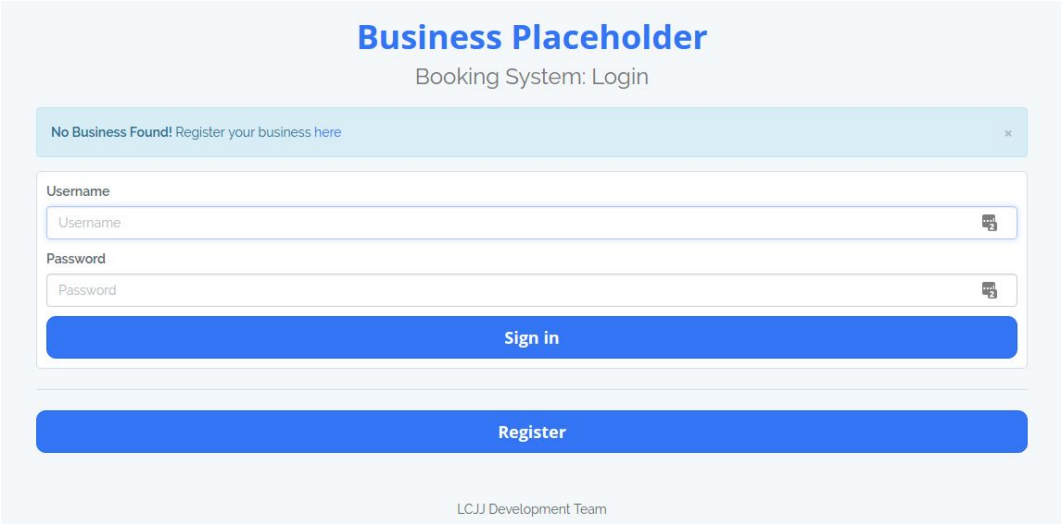
Ignore the timestamp prefix on the files, they do not have to be exactly the same as output shown above.

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.



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://locahost: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

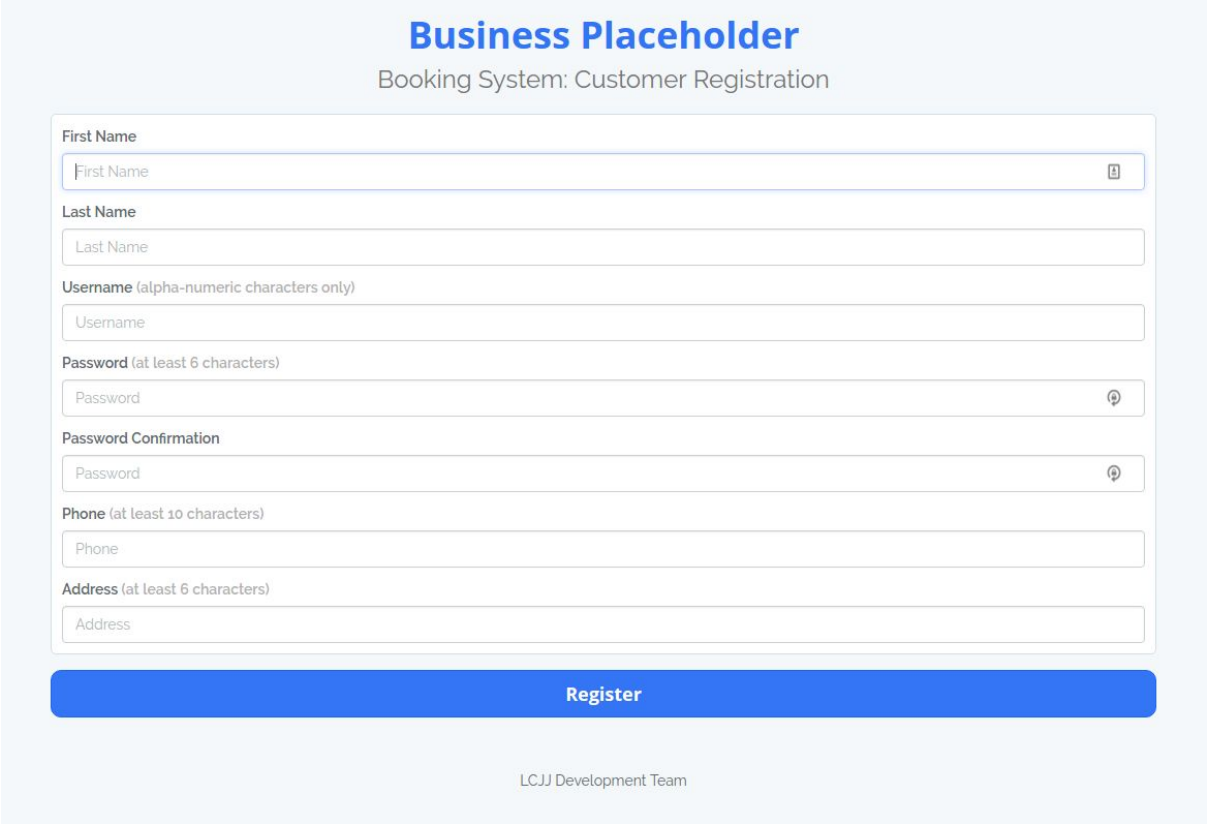
Requirements

The only software needed to use the application is a web browser as started in the software prerequisites.

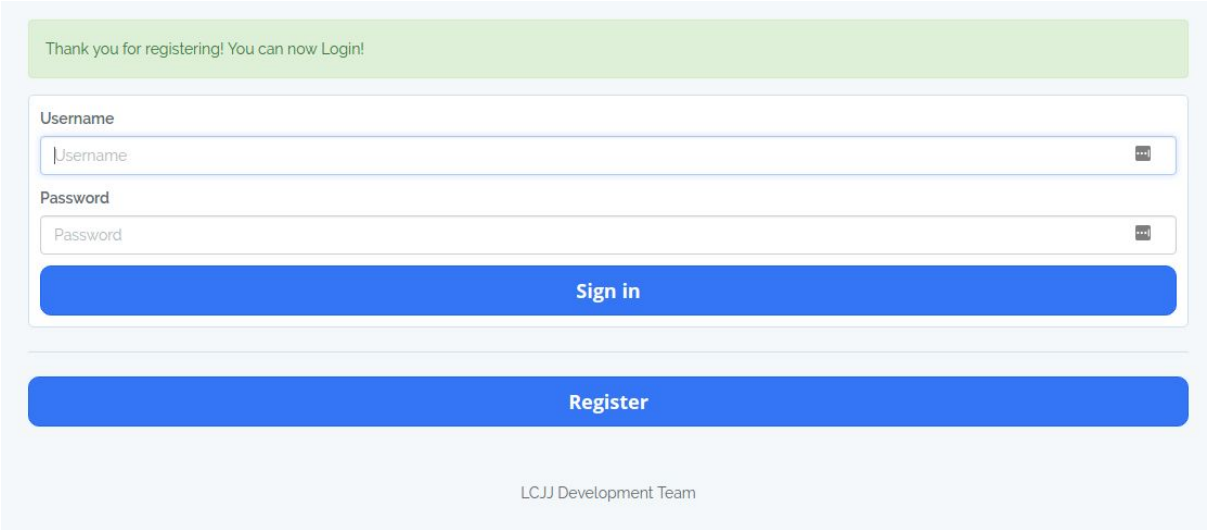
Customer

Register

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



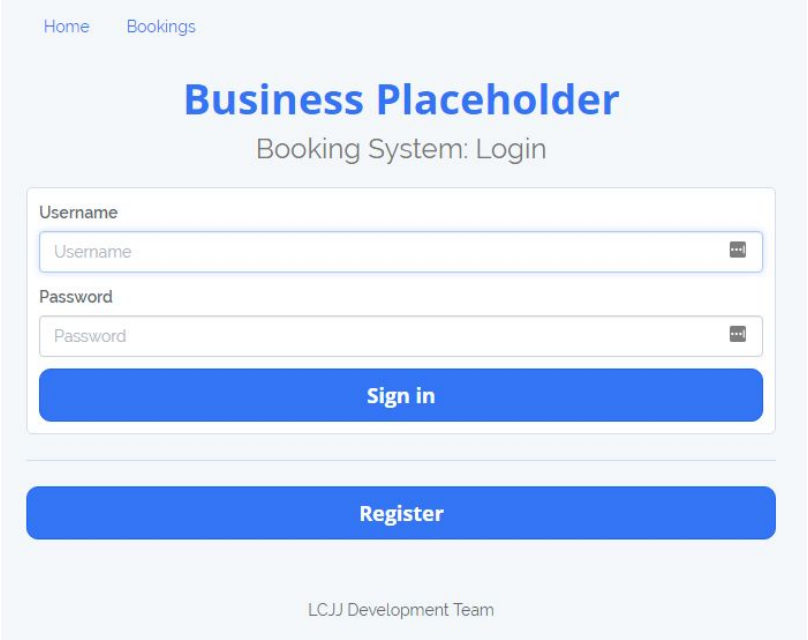
Click the 'Register' button to submit the customer registration form. Once the form has been successfully submitted and validated, a redirection to /login and a message shown stating a successful customer registration.



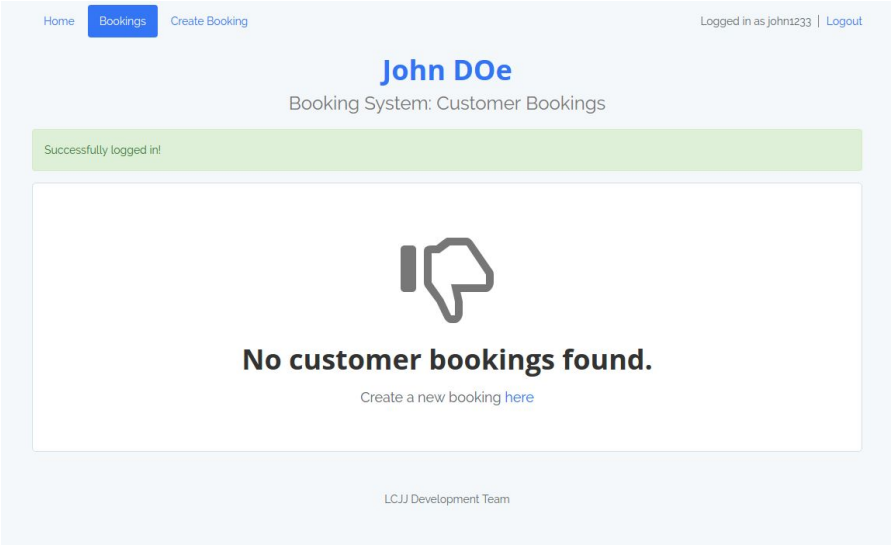
Login

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.



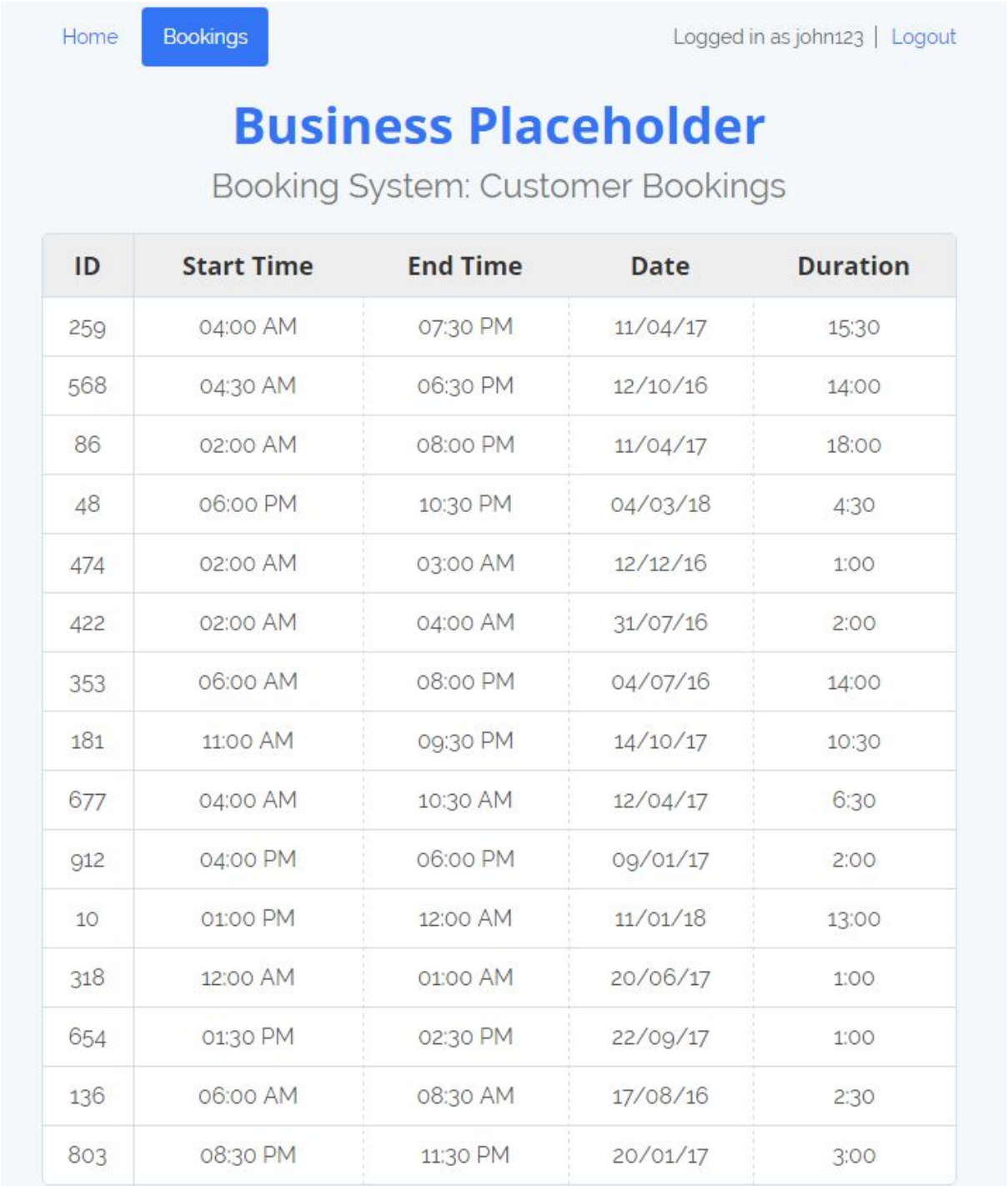
Once the login form is filled, click the 'Sign in' button and the page will redirect to /bookings if the login was successful.



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.

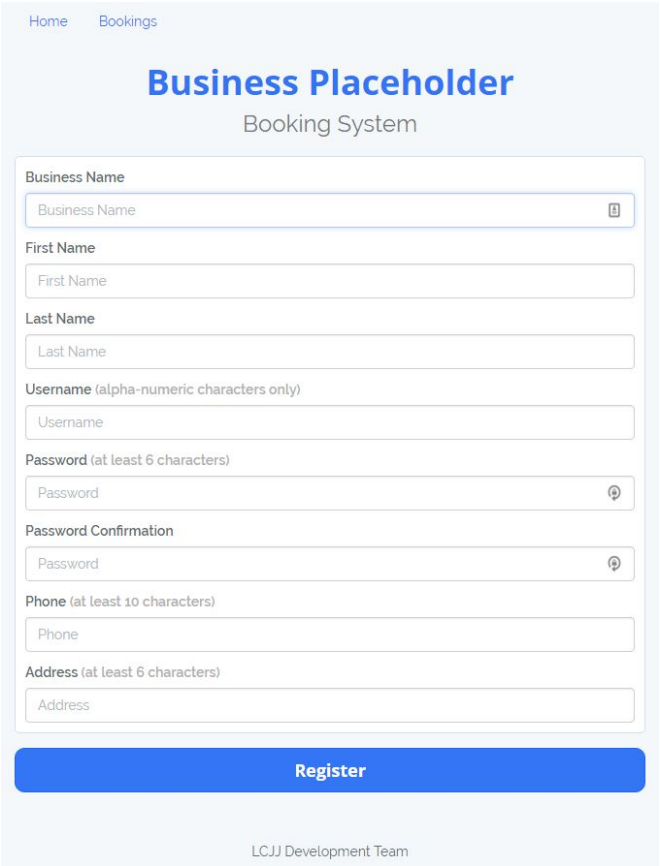


ID	Start Time	End Time	Date	Duration
259	04:00 AM	07:30 PM	11/04/17	15:30
568	04:30 AM	06:30 PM	12/10/16	14:00
86	02:00 AM	08:00 PM	11/04/17	18:00
48	06:00 PM	10:30 PM	04/03/18	4:30
474	02:00 AM	03:00 AM	12/12/16	1:00
422	02:00 AM	04:00 AM	31/07/16	2:00
353	06:00 AM	08:00 PM	04/07/16	14:00
181	11:00 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

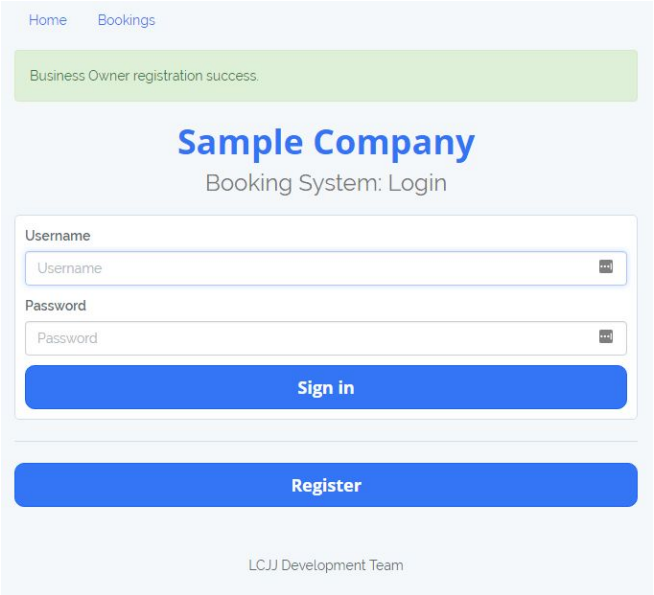
Business Owner (Administrator)

Register

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



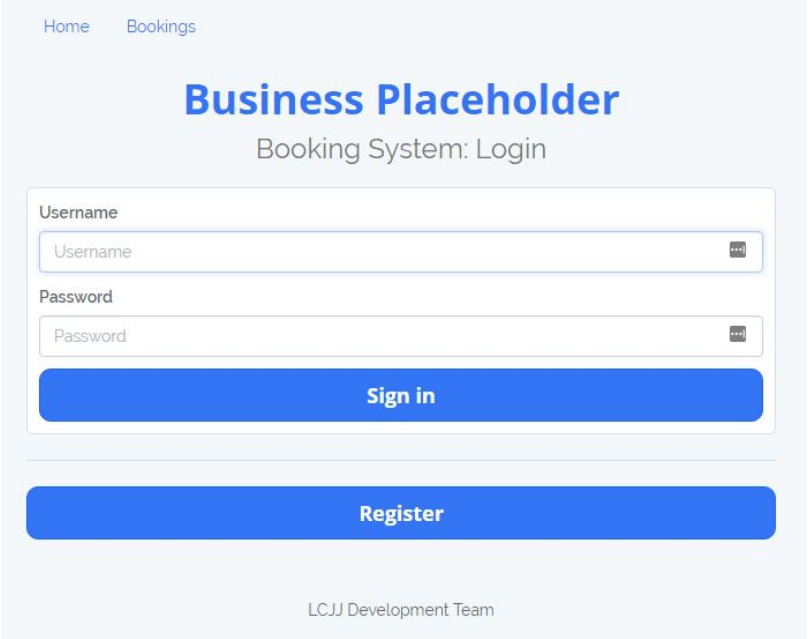
Click the 'Register' button to submit the business owner registration form. Once the form has been successfully submitted and validated, a redirection to /login and a message shown stating a successful business owner registration including the business itself.



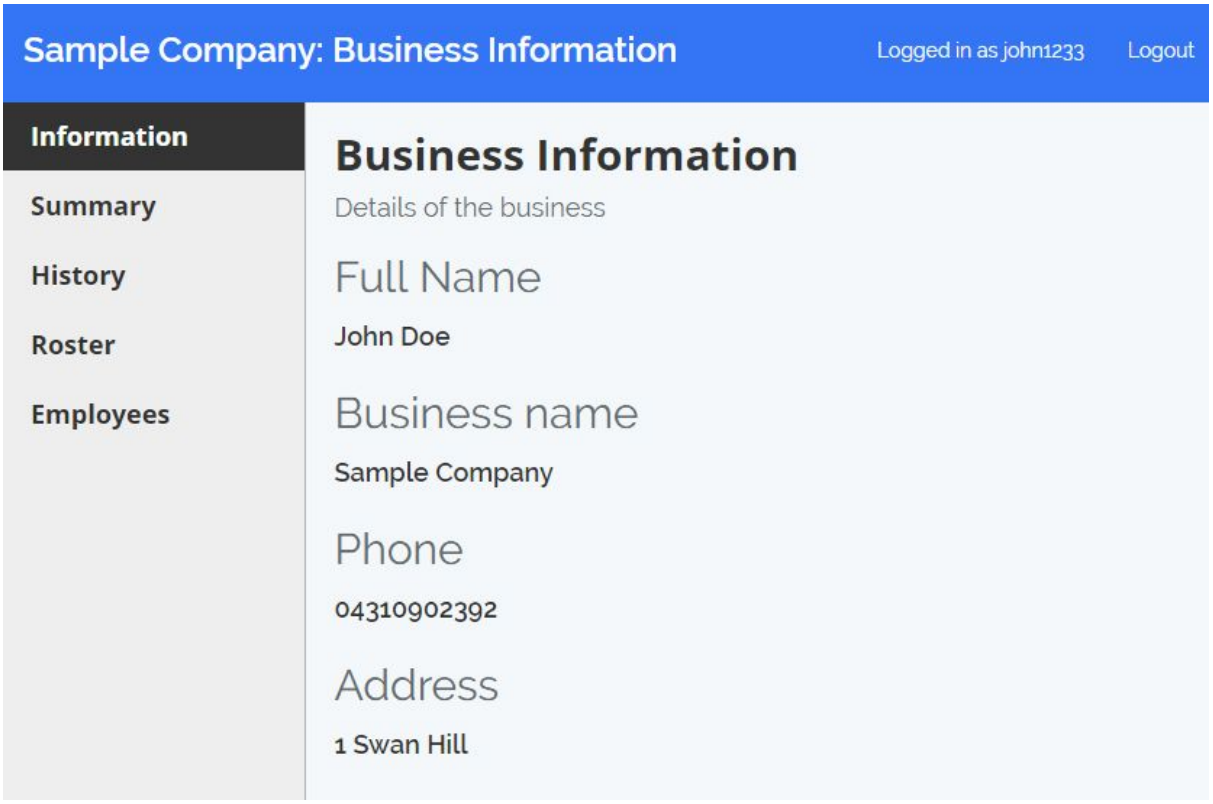
Login

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.



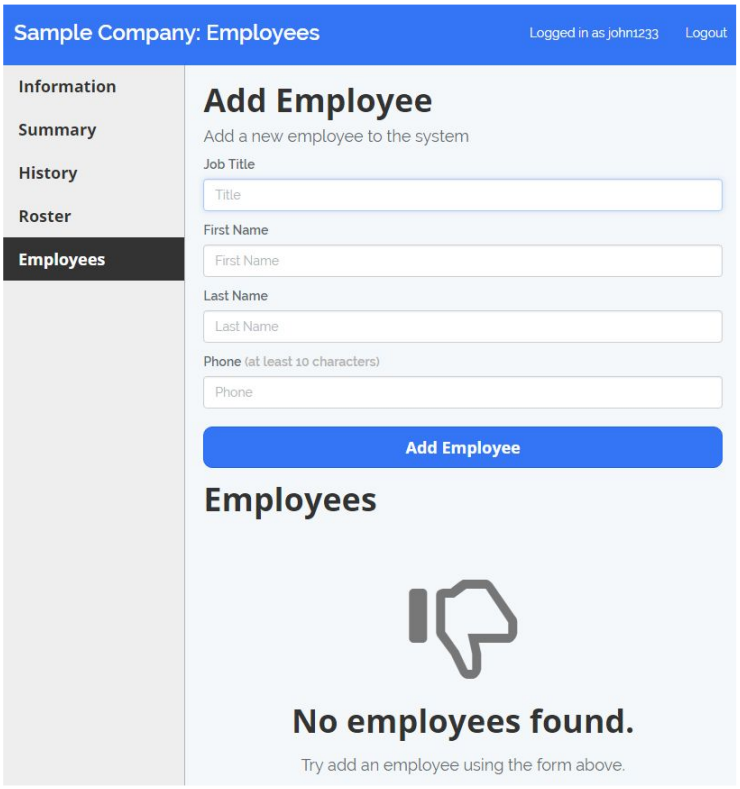
Once the login form is filled, click the 'Sign in' button and the page will redirect to /admin if the login was successful.



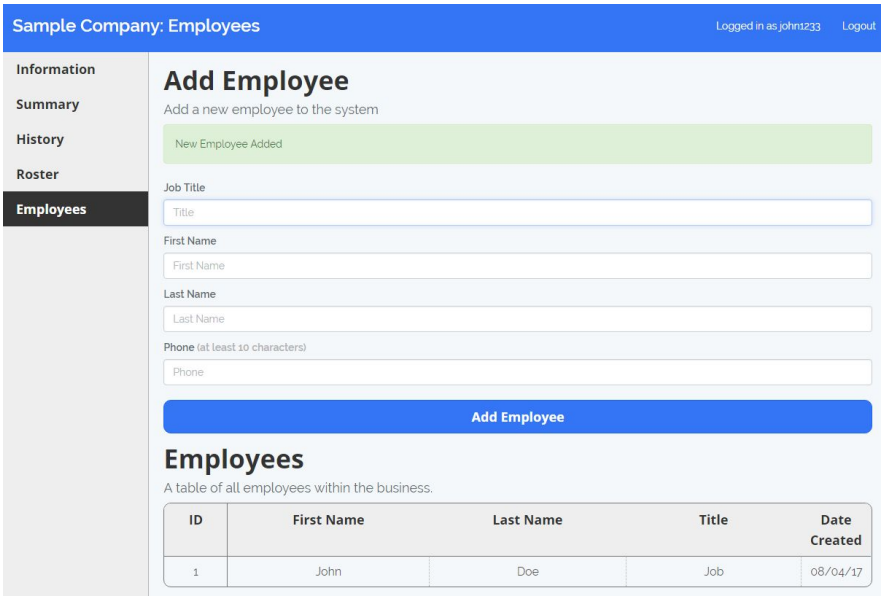
Add Employee

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.



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

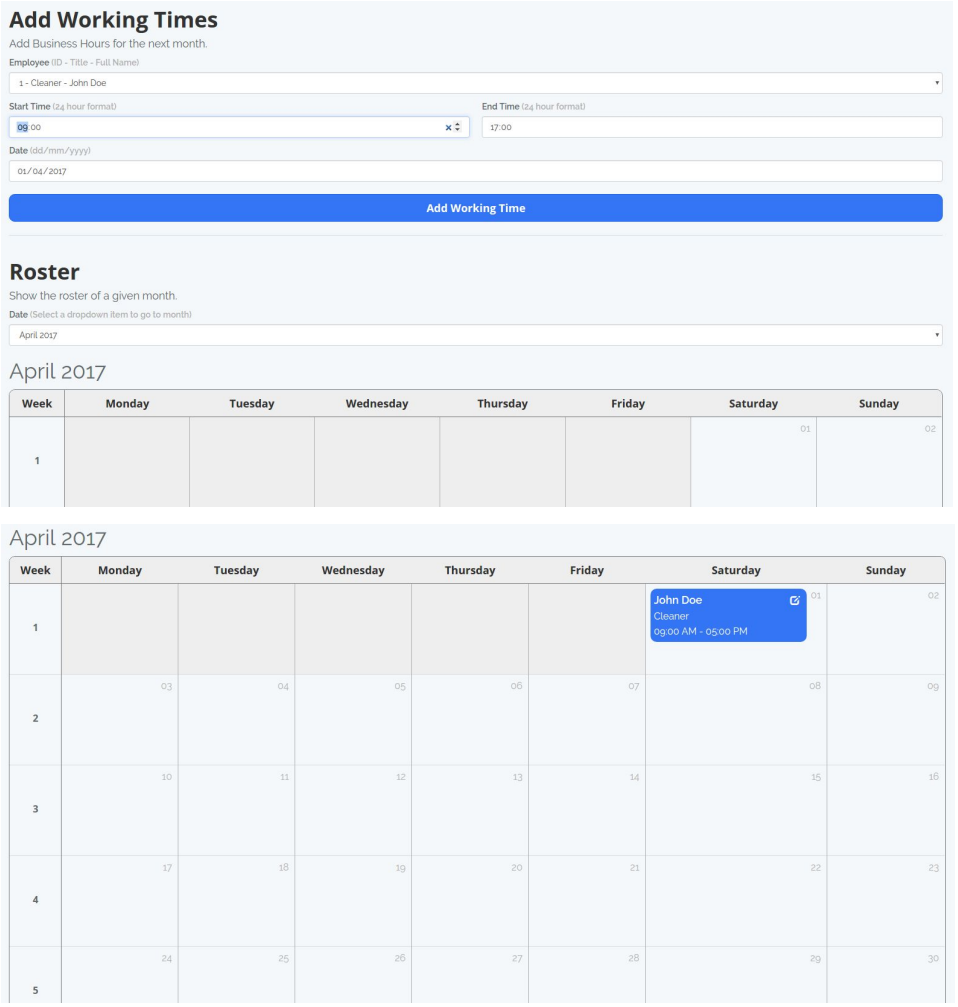


Add Working Time

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.

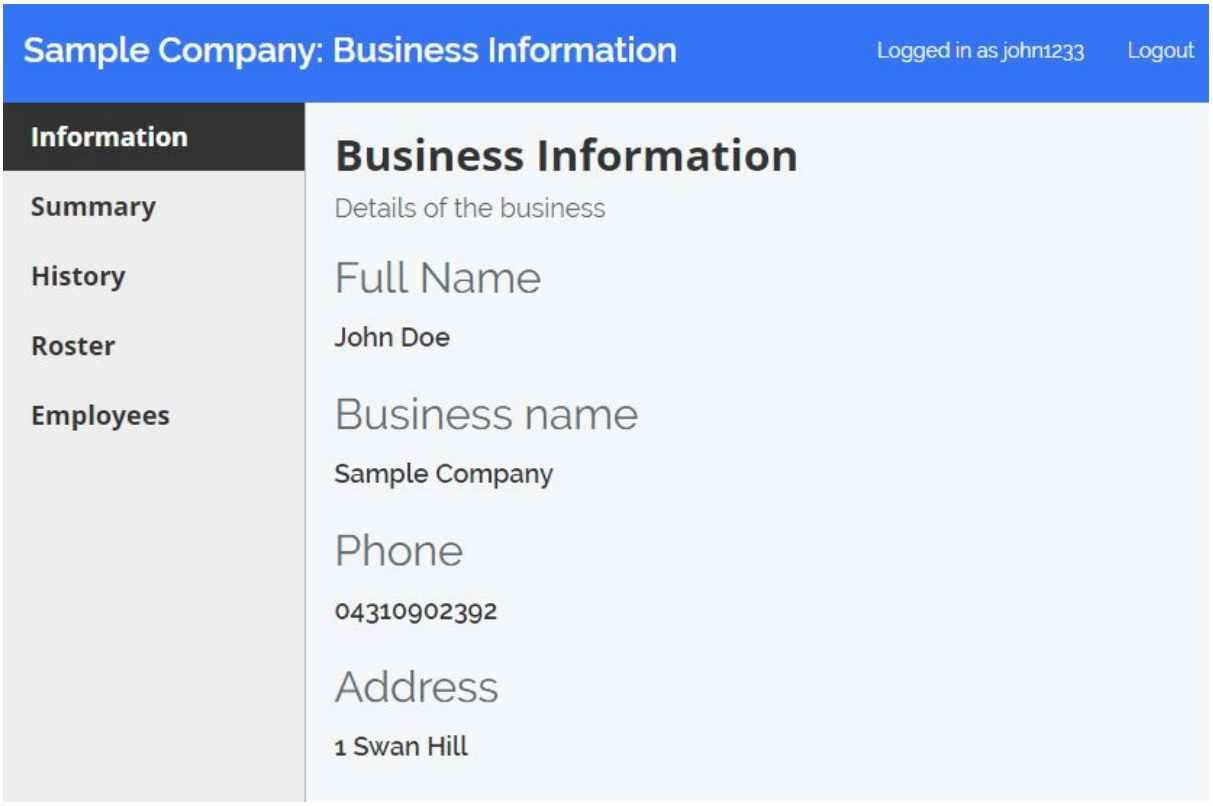
Fill out the form provided on the page and click the 'Add Working Time' button to submit the form.

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



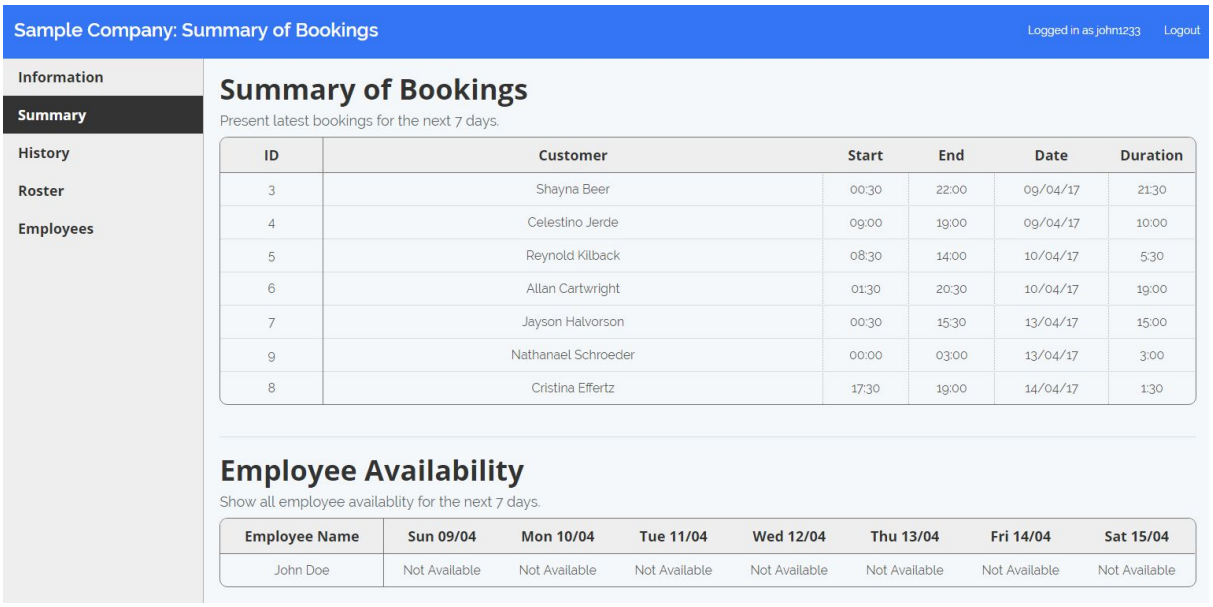
View Business Information

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



View Employee Availability

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



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

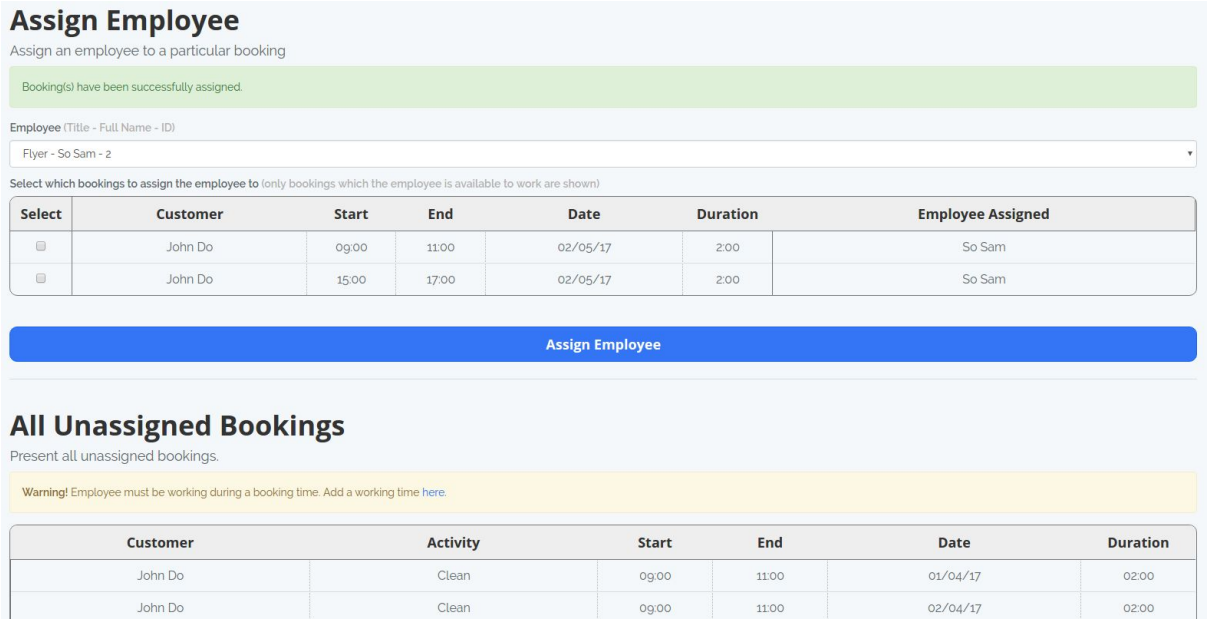
Present older bookings.

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

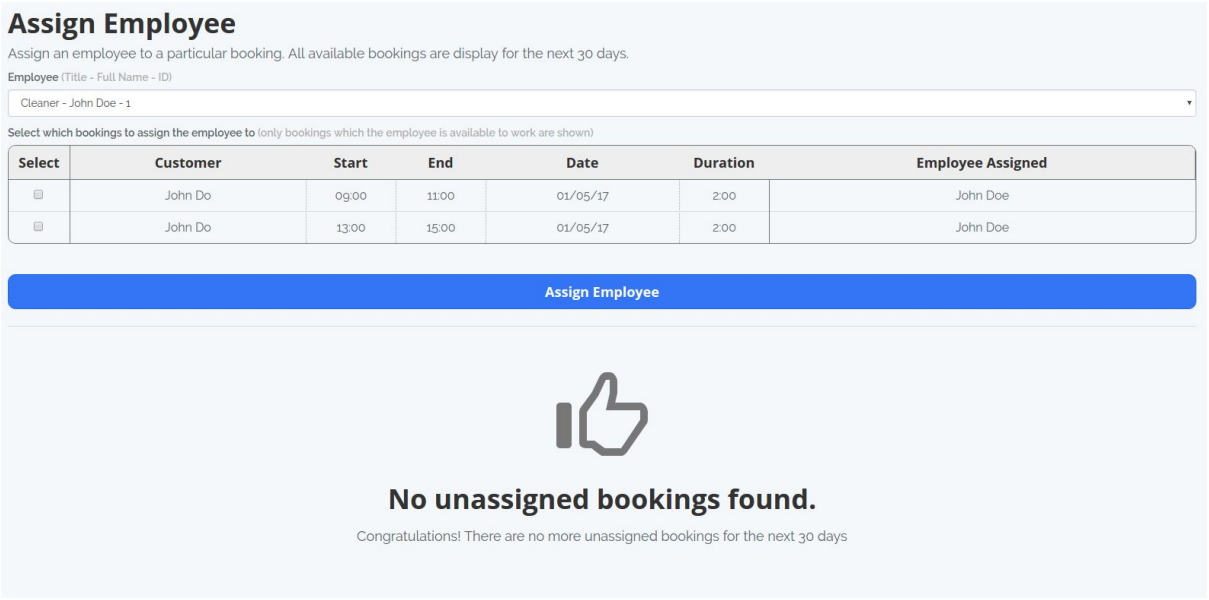
Assign Employee to Booking

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.



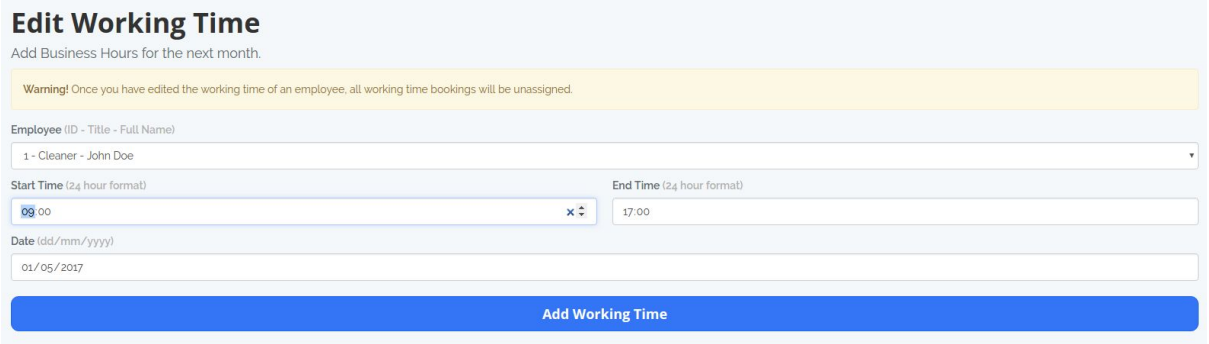
A table of unassigned bookings will assist you in missing bookings that a customer/admin submits without an employee.



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.



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.