Science Labs Inventory System – Requirements and Manual

# Server Requirements

* Apache 2.4.39 Web server
* MariaDB 10.3.16 Database server

PHP 7.3.7

## Packages used

All these packages are included in th project

* PHPSpreadsheet
* PHPMailer
* Bootstrap 4
* Gijgo
* JQuery
* Popper

## Setting up PHPMailer

* PHPMailer requires an internet connection to connect to a Google SMTP server.
* The username, password, and account details for the email ID need to be set in maildetails.php

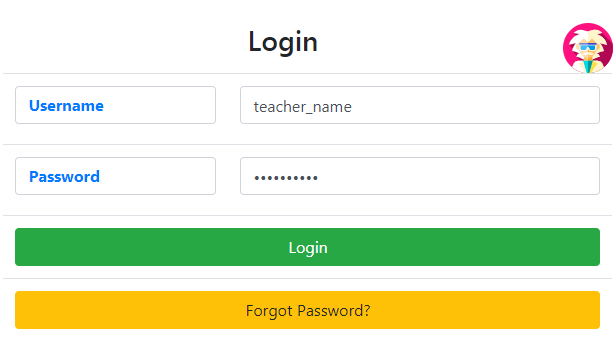
## Maintenance

* The server name, username, password, and database name must be set up in db.php to allow the project to connect the database server
* The projects needs 13 tables to be created. The SQL to create these has been given at the end of the document
* Information for the class, student, and teacher tables needs to be entered
  + Teacher name, email ID and access level
    - Access Level 0 – Admin
    - Access Level 1 – Head of Department
    - Access Level 2 – Lab In-Charge
    - Access Level 3 – Regular Teacher
  + Class name and the id of the class’ class teacher
  + Student Roll. No (as id), student name, and the id of their class
* The student table will have to be updated every year to account for new students and class changes

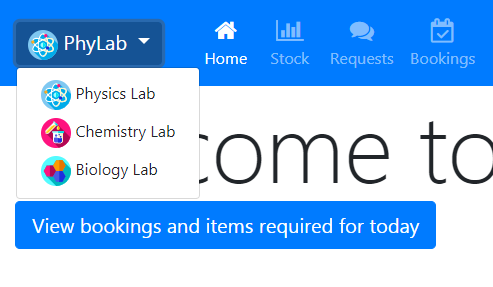
# Client Requirements

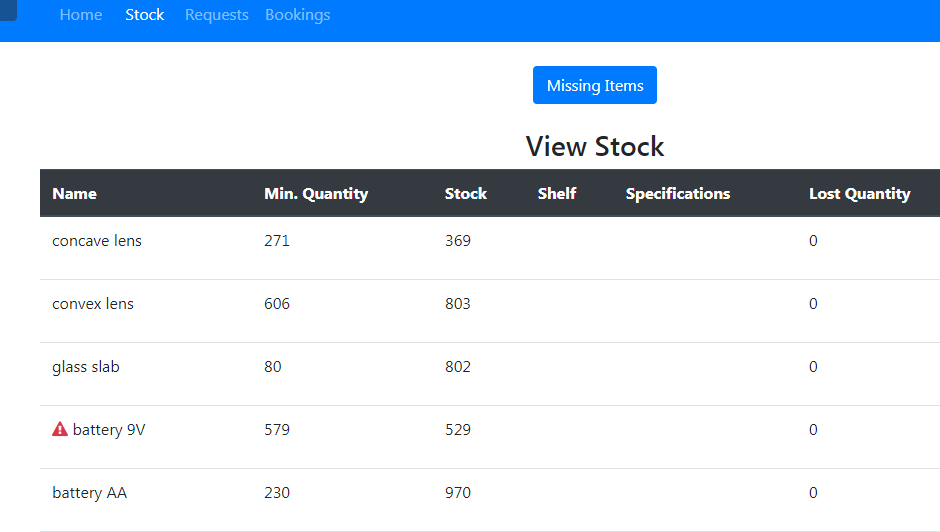
* Mozilla Firefox of Google Chrome
* The project may not work on Internet Explorer

# User Manual

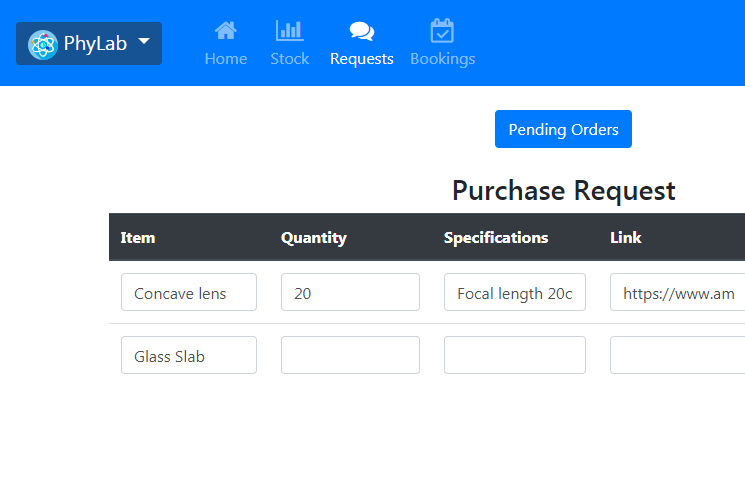
The starting page is a login page. You cannot do anything without logging in. So log in first.

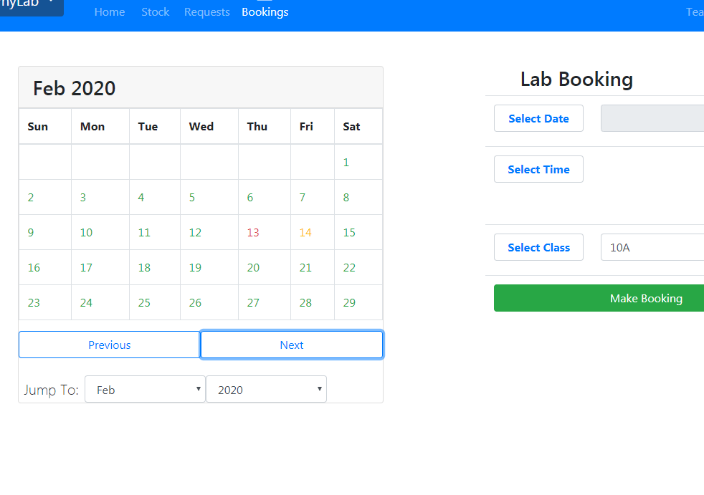
The first time you log in, your password will be set randomly. You will have to click Forgot Password and change it.

The navigation bar has a button called PhyLab. This stands for physics lab. It is important that before you start you MUST choose the right lab. The color of the navigation bar changes to remind you which lab you are in (blue – physics, red – chemistry, green – biology). The lab you are using can be changed at any time using the same dropdown.

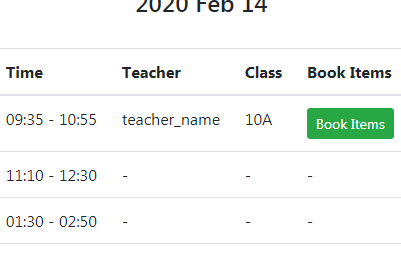


The stock tab lets you see all the available items in the lab you are in. This helps you plan ahead to make sure that you have what you need for your classes.

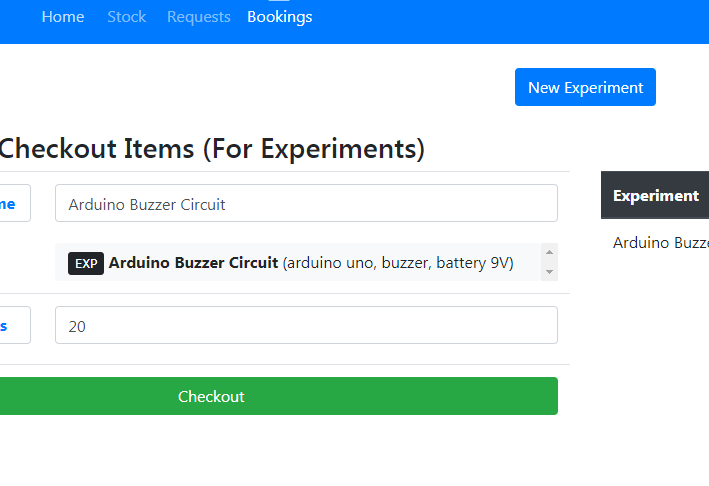
The request table allows you to put in a purchase request for an item. Click on ‘Add Order’ to add another row to the table, and ‘Place order’ to submit it. You MUST also provide a link and the approximate cost of your request. These requests are compiled periodically, usually around once a month (though this may change depending on the lab/lab in-charge). You will be notified by email when your shipment arrives.



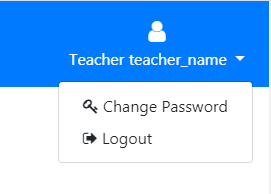
The bookings tab allows you to book the lab for a period of time, and also book the items that you will require so that these can be set aside. You can also use it to see when the lab has been booked and by who. The yellow days on the calendar mean that some slots are booked, while red days indicate a completely booked lab for the entire day.



Once you book the lab, click on the day you have booked it and go to ‘book items’ book an experiment. This is mandatory so that the lab in-charge can set aside these items for you, and it also helps the school keep a record of all the experiments that were conducted in the lab.



You can only book the items for a single experiment. If you are unable to find the experiment you are looking for, you can always create it yourself using the ‘New Experiment’ button.



You can click on your username on the top right corner to logout or change your password.

That’s all, folks!

# SQL Code

create database labs;

use labs;

create table teacher(

    id int not null unique,

    teacher\_name varchar(50) not null unique,

    teacher\_pass varchar(60) not null,

    levels int(1) not null,

    email varchar(40) default "<-- no email provided -->",

    primary key(id)

);

create table class(

    id int not null auto\_increment unique,

    class\_name varchar(6) not null unique,

    class\_teacher int not null unique,

    primary key(id),

    foreign key (class\_teacher) references teacher(id)

);

create table student(

    id int not null unique,

    student\_name varchar(50) not null unique,

    class\_id int not null,

    primary key(id),

    foreign key (class\_id) references class(id)

);

create table item(

    id int not null auto\_increment unique,

    item\_name varchar(50) not null,

    lab\_location varchar(10),

    specs varchar(100) default '',

    min\_quantity int default 0,

    lost\_quantity int default 0,

    quantity int default 0,

    price decimal(10,2) not null default 0,

    lab varchar(1) not null default "p",

    recon date,

    primary key(id)

);

create table student\_checkout(

    id int auto\_increment not null unique,

    student\_id int not null,

    item\_id int not null,

    quantity int not null,

    returned varchar(1) not null default "N",

    checkout\_date date default now(),

    lab varchar(1) not null default "p",

    returned\_date date,

    lost varchar(1) not null default "N",

    primary key(id),

    foreign key (student\_id) references student(id),

    foreign key (item\_id) references item(id)

);

create table purchase\_request(

    id int not null auto\_increment unique,

    teacher\_id int not null,

    item\_name varchar(50) not null,

    quantity\_ordered int not null,

    quantity\_received int,

    specs varchar(100) not null,

    link varchar(150) not null,

    cost decimal(10,2),

    date\_ordered datetime not null default now(),

    arrived bit not null,

    date\_arrived datetime not null default now(),

    comments varchar(100),

    bill\_code varchar(10) default "<No Bill>",

    lab varchar(1) not null default "p",

    primary key(id),

    foreign key (teacher\_id) references teacher(id)

);

create table dept\_transaction(

    id int not null auto\_increment,

    from\_lab varchar(1) not null,

    item\_id int not null,

    quantity int not null,

    to\_lab varchar(1) not null,

    transfer\_date datetime not null default now(),

    comments varchar(100),

    primary key(id),

    foreign key (item\_id) references item(id)

);

create table lab\_borrow(

    id int not null auto\_increment,

    from\_lab varchar(1) not null,

    item\_id int not null,

    quantity int not null,

    to\_lab varchar(30) not null,

    transfer\_date datetime not null default now(),

    return\_date datetime,

    comments varchar(100),

    item\_status varchar(10) not null default 'PENDING',

    primary key(id),

    foreign key (item\_id) references item(id)

);

create table lab\_booking(

    id int not null auto\_increment,

    booked\_date date,

    booked\_time varchar(15),

    teacher\_id int,

    class\_id int,

    lab varchar(1) default "p",

    primary key (id),

    foreign key (teacher\_id) references teacher(id),

    foreign key (class\_id) references class(id)

);

create table experiment(

    id int not null auto\_increment,

    exp\_name varchar(50) not null,

    lab varchar(1) not null,

    primary key (id)

);

create table experiment\_item(

    id int not null auto\_increment,

    exp\_id int not null,

    item\_id int not null,

    quantity int not null,

    primary key (id),

    foreign key (exp\_id) references experiment (id)

);

create table item\_booking(

    id int not null auto\_increment,

    labbooking\_id int not null,

    exp\_id int not null,

    quantity int not null default 1,

    returned varchar(1) default "N",

    primary key (id),

    foreign key (exp\_id) references experiment (id),

    foreign key (labbooking\_id) references lab\_booking (id)

);

create table missing(

    id int not null auto\_increment,

    item\_id int not null,

    quantity int not null,

    entry\_date date default now(),

    comments varchar(100) not null,

    accounted varchar(1) default "N",

    checkout\_id int,

    primary key (id),

    foreign key (item\_id) references item (id)

);