

DUBLIN PARTY HIRE WEBSITE



Technical Documentation

Url: https://stu33001.scss.tcd.ie/group_3/

Web Developers (SQLive)

Hugo Bolger
Rose Connolly
Ranjana Singh

When testing our website, we recommend using your real email address, as the multiple files make use of the php mail function.

The following documentation includes details to help a system developer understand the technical workings of the system for Dublin Party Hire's website. This website was created from February - March 2021. The website runs on the macneill server. Data for DPH is stored in tables in a mysql database. PHP is used to interact with this database. The website was developed to meet the requirements specified by the client. The main functionalities of the website include;

- An ordering system for customers to create new events and order equipment.
- A system for workers to be assigned shifts, clock in and out.
- The option for admins to change VAT rates, prices, employee details etc.
- The option for admins to view certain reports, such as sales per equipment.

The following logins can be used to navigate the website on staff and admin pages:

Staff:

Staff id: 101

Password: Duncan1D

Admin:

Staff id: 106

Password: f12TsD

Contents

Assumptions

Further Recommended Development

Diagrams

Fig1: Navigation Logic for Customer Registering a New Event

Fig 2: Customer Landing Page (customeraccount.php) Options

Fig 3: Index to staff side login

Fig 4: Staff (driver) Side Navigation

Fig 5: Admin Side Navigation

Fig: 6 staff hours (admin) Navigation

File References

SQL Tables

List of all Filenames

Assumptions

- **Sales/Discounts** - Sales on equipment are applied to the rental price and the setup price
- **Costs** - The customer pays a rental and setup cost for equipment based on the quantity ordered. The cost is not based on the number of days they keep the equipment.
- **VAT** - VAT is charged on the equipment rental and setup, not on delivery. Based on accounting standards, VAT is not included when determining the revenue DPH receives from orders. We have, however, included VAT in reports determining exactly how much customers have spent.
- **Returning Equipment** - The customer can keep equipment for up to five days after the collection/delivery date of the equipment. If needed to be changed, change relevant code in `newevent.php`.
- **Setup is all or nothing:** Customers cannot choose to have 2 tables setup, and 2 not setup, it is all or nothing. If they order 2 tables, no setup. -and then afterwards they select two tables with setup. All tables will have to be set up.
- **Staff Positions** - Only two positions available:
 - Driver - For all warehouse workers who would carry out delivery and pickups
 - Manager - Managerial Position (Aidan MacKeaney) giving only him admin access

As there is only one manager, manager and admin are used interchangeably in this documentation.

Further Recommended Development

- **Option to remove specified quantity of equipment:** In `editorder.php`, the user only has the option to remove all quantities of a product, introducing the option to remove some of a quantity is a feature that could be added.
- **Merge `loginghost.php` and `registerghost.php` with `login.php` and `registration.php`:** The code for these are very similar, but they redirect to different pages. The issue would be determining the criteria for where to send the user. You could check whether there exists a newly inputted event with a null email, and if true send to `invoice.php`, but this risks sending a customer who is just logging in to the invoice of another customer who is ordering equipment at the same time and is yet to get to login. A better option would be to create a session variable 'hasevent'. If this is set, the customer should be redirected to their invoice, if not set, it means the customer is just registering and should be redirected to `customeraccount.php`
- **Redirects:** Most forms print out a confirmation message and clear the form entries to prevent duplicate data entry following page refreshes. Others which cannot follow that structure require a redirect to a confirmation page. Where more efficient, confirmation messages should be made more prominent.
- **Data Retention Following Submit:** Most forms, where possible, use PHP echo statements to ensure select statements retain the user information after an unsuccessful submit. However, due to some select statements taking in a unique primary key, rather than the name or text associated with it, the text could not be retained. The code may be adjusted to account for this, however primary keys were necessary and their retrieval was made with efficient code.
- **Forgot Password Redirect:** After a user selects the option to change their password, they are sent to do so in a new tab. They currently receive a success message upon completion, and they can click back to the page they were on. This could potentially be changed. If desirable, the pages for changing the password could open in the current tab. They would then be redirected directly to their homepage (or to continue confirming their event depending on where they were) once they have changed their password. This could be done by setting a session variable in the source page, to let `changepassword.php` know where to send a user once their password is successfully changed.

Diagrams

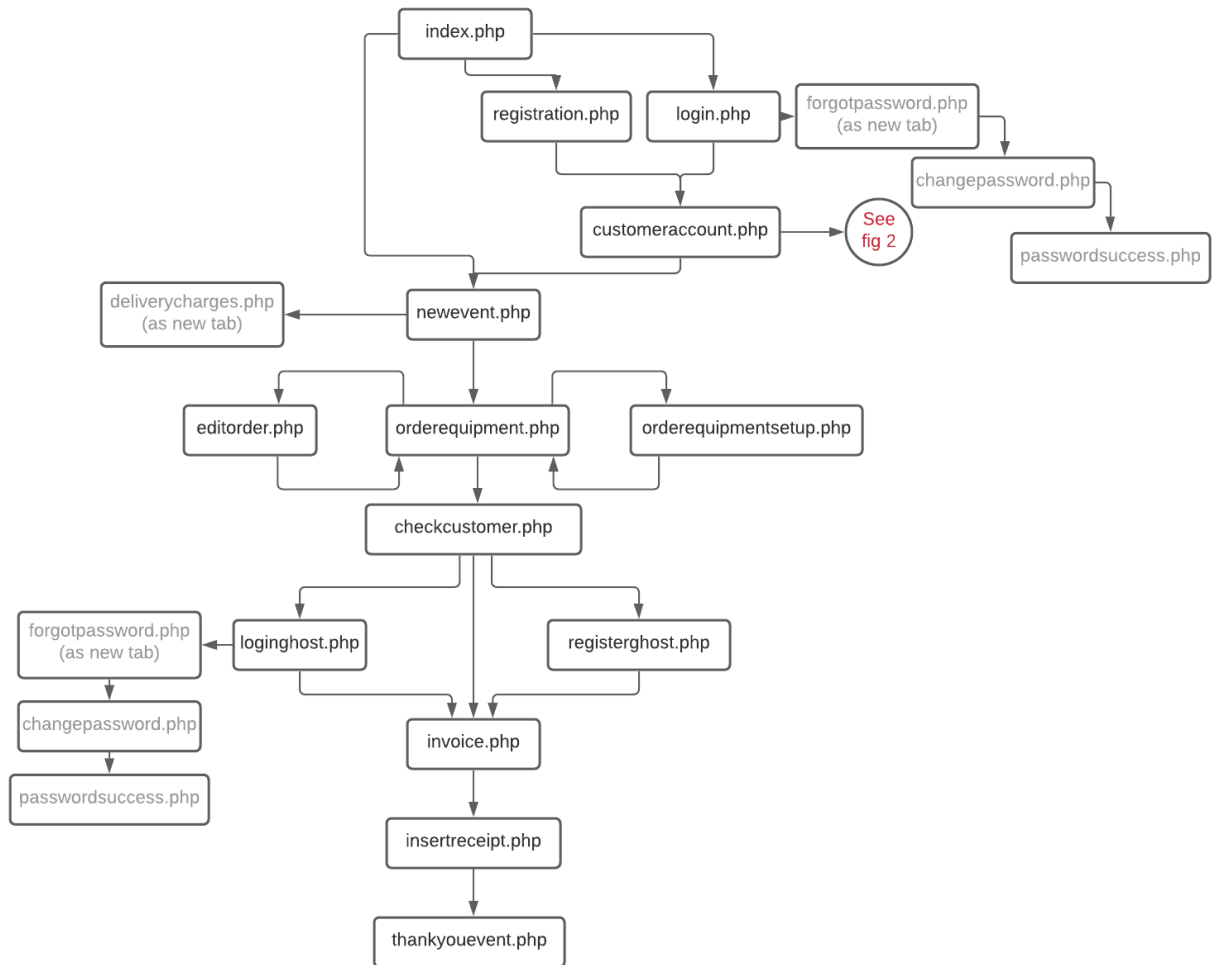


Fig1 - Navigation Logic for Customer Registering a New Event

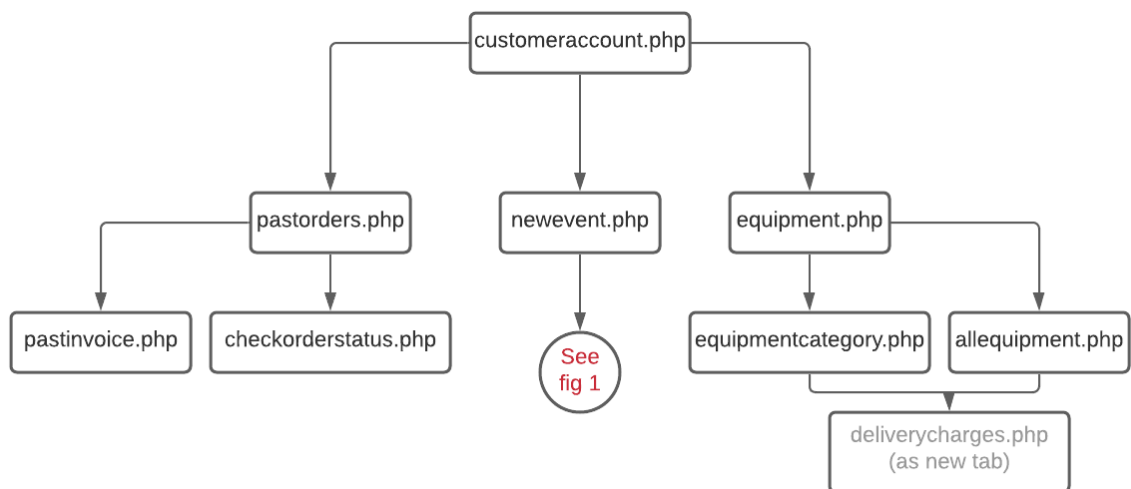


Fig 2: Customer Landing Page (customeraccount.php) Options

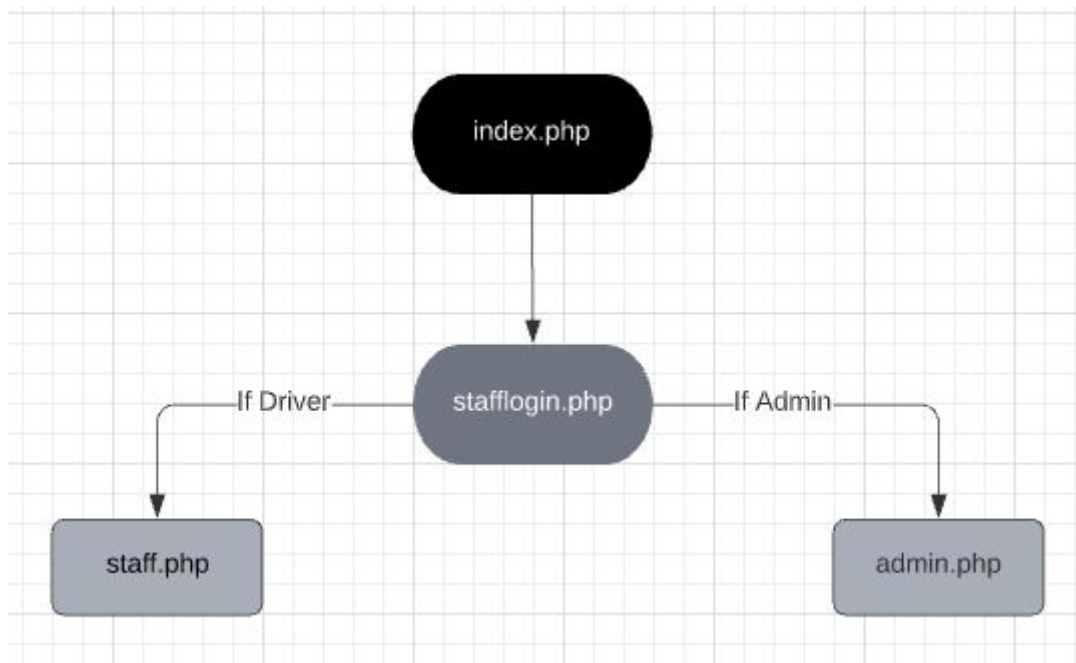


Fig 3: Index to staff side login

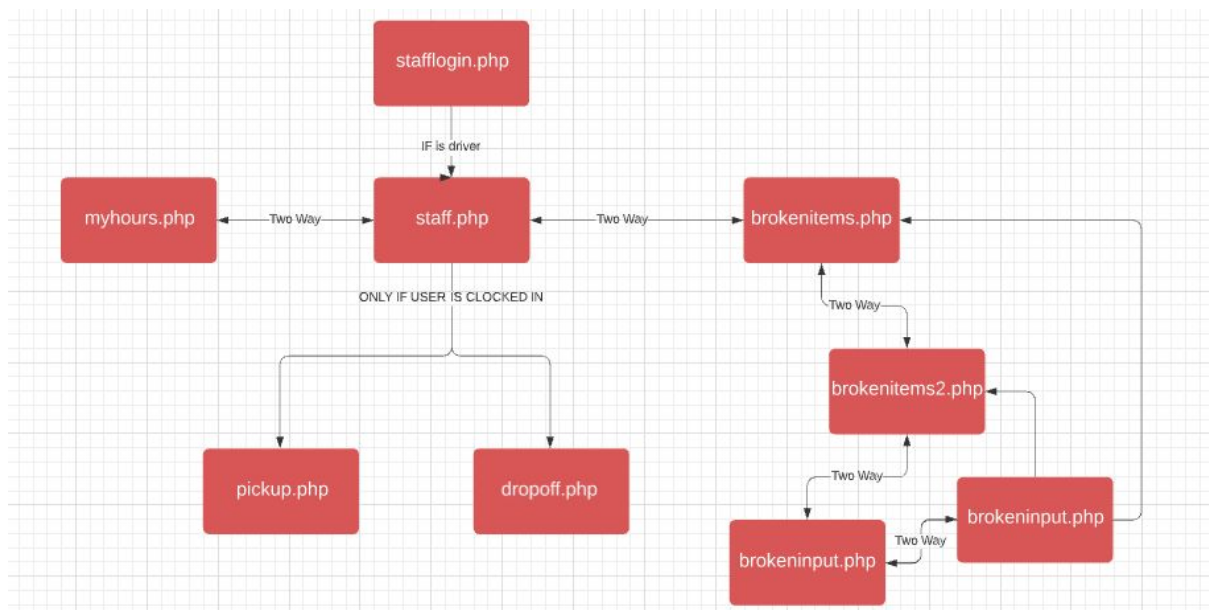


Fig 4: Staff (driver) Side Navigation

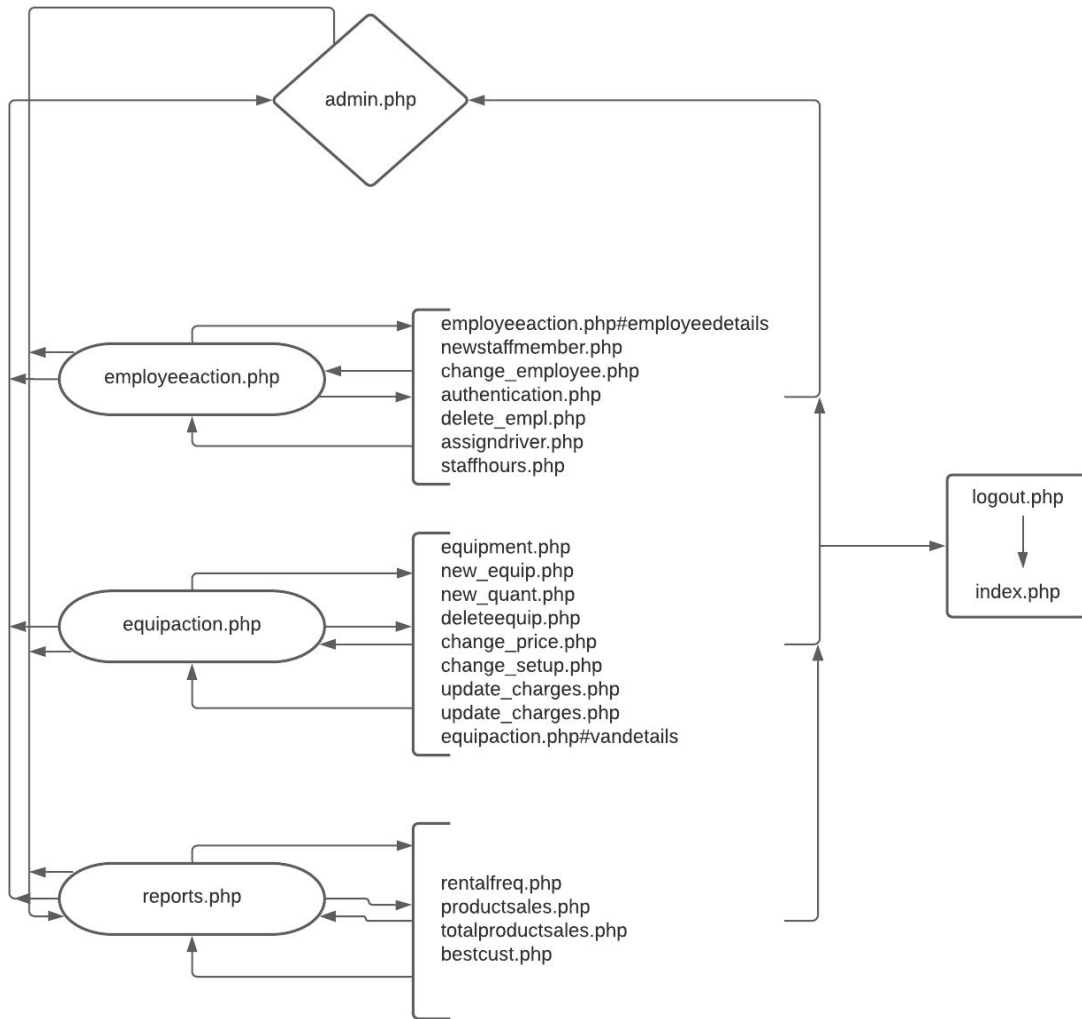


Fig 5: Admin Side Navigation

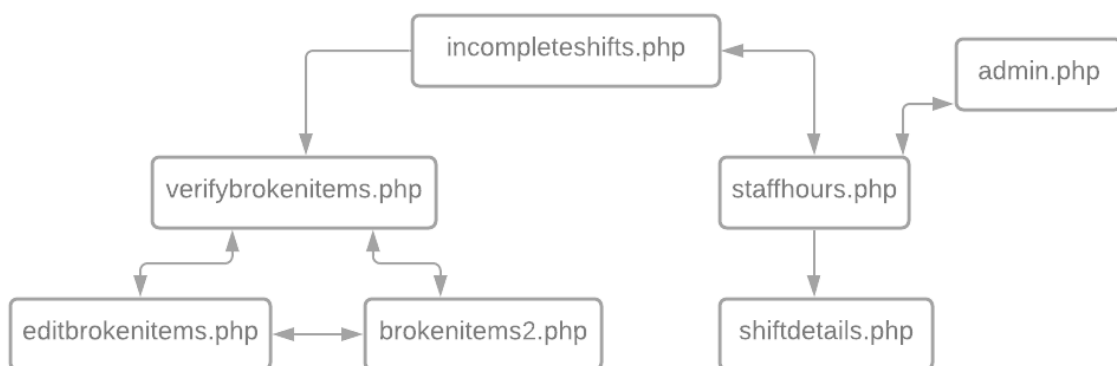


Fig 6 staff hours (admin) Navigation

File References

The pages for the website are all organised alphabetically for ease of reference. Some pages that are not technically complicated are not included, as we feel the file commenting is adequate.

admin.php

Purpose: allow administrators (managers in the organisation) access to operational functions relating to employees, equipment, scheduling work hours and viewing reports

Accessed from:

stafflogin.php if user is a manager (giving them admin access)

The navigation bar from all pages reached through admin.php

- employeeaction.php and related pages
- equipaction.php and related pages
- reports.php and related pages

Sends user to: The above three pages and logout.php which redirects to index.php

allequipment.php

Purpose: display all equipment in DPH's catalogue

Accessed from:

index.php/home
customeraccount.php

Sends user to: option to view deliverycharges.php (in a new tab)

- Displays all equipment details in a table
- If there is a sale, this is displayed in a column, otherwise nothing is displayed.

assigndriver_equipmentlist.php

Purpose: Present admin with all events and equipments ordered for a date set on assigndriver.php
Admin can scroll through events using next/prev

Accessed From:

assigndriver.php

Sends User To:

Pop-Up page with no links
nextpageassignevents.php & prevpageassignevents.php allow for scrolling function

assigndriver.php

Purpose: Admin can assign drivers to shifts and assign events to shifts

- The page stores a date selected or scrolled to by the admin (using next/prev links on top of the page)
- A table at the bottom of the page shows events for the day broken down by delivery / collection and drop-off (outgoing) / pick-up (incoming) as well as the number of drivers assigned to each event
- The admin can then create a new shift by assigning a driver a clock in and clock out time, or the admin can update existing shifts already assigned on the chosen date
- Once a shift is selected, the admin can then assign events to shifts

Accessed From:

admin.php
employeeaction.php

Sends User To:

Nav Bar - admin.php, employeeaction.php, logout.php

Redirects to: deleteshifts.php - in new tab
assigndriver_equipmentlist.php - in new tab
finishinputshifts.php - for exiting the assign shift loop and unsetting session variables
nextassigndriver.php & prevassigndriver.php for scrolling function for date

authentication.php

Purpose: for extra security, the admin is first redirected to this page when they click "Change Passwords" in employeeaction.php. This page requires the admin to validate their password again before altering driver passwords. The session variable \$_SESSION['staff_id'] is used to determine the admin's current password.

Accessed from:

employeeaction.php when Change Passwords is clicked

Sends user to:

Nav bar: logout redirecting to index .php, employeeaction.php and admin.php

Redirects to: change_password.php when correct information is input

best_cust.php

Purpose: display a table of the total spent by each customer in descending order in a given timeframe using mysql and php queries. If no time frame is selected, then all data is shown between earliest and latest time. All inputs are validated.

Accessed from: reports.php when Best Customers button is clicked

Sends user to:

Nav bar: back to reports.php, admin.php, logout.php redirecting to index.php through

Note:

- product_sales.php is a report showing the revenue (excluding VAT) brought in by a specific product in a time period
- totalproductsales.php is a report showing the total revenue (excluding VAT) brought in by all products in a time period
- rentalfreq.php is a report that shows how often a product was rented in a time period

The above reports are generated using mysql queries and table joins between the receipts, equipment and equipment_rented tables.

All pages are accessed through reports.php and redirect to the same pages as best_cust.php through the navigation bar

brokenbackpage.php

Purpose: Relocate user from brokenitems2.php to brokenitems.php

Accessed From: brokenitems2.php (staff side only), brokeninput.php

Sends user to: brokenitems.php

brokeninput.php

Purpose: One of 3 key pages for logging broken returns or unreturned items after events
This page displays the inputs that are already stored

Accessed From: brokenitems2.php, editbrokenitems.php

Sends User To:

Nav Bar: staff.php, myhours.php, logout.php

Redirects to: brokenbackpage.php -> return user to brokenitems.php

brokenitems2.php -> add more items

editbrokenitems.php -> delete inputs

brokenitems.php

Purpose: First Page for logging broken returns or unreturned items after events
Page allows a staff member (driver) to select one of their complete 'Pick-Up' shifts

Accessed From:

Nav Bar: staff.php, myhours.php
pickuplist.php

Sends User To:

Nav Bar: staff.php, myhours.php, logout.php
Redirects to: brokenitems2.php -> continue the process

brokenitems2.php

Purpose: One of 3 key pages for logging broken returns or unreturned items after events

- Can be accessed from both staff and admin side of the website
- Both options offer different links but provide the same functionality and output
- The goal is to allow users to input broken / unreturned equipment from a stored event and ensure that the total broken input is less than the total ordered equipment

Accessed From:

Staff Side:

brokenitems.php, brokeninput.php, editbrokenitems.php

Admin Side:

verifybrokenitems.php, editbrokenitems.php

Sends User To:

Staff Side:

Nav Bar: staff.php, myhours.php, brokeninput.php, logout.php
Redirects to: brokenbackpage.php
brokeninput.php

Admin Side:

Nav Bar: staffhours.php, verifybrokenitems.php, editbrokenitems.php, logout.php .
Redirects to: verifybrokenitems.php

change_employee.php

Purpose: a form that allows the admin to change an employee's contact details and their position (two of driver and manager but more can be added as needed into the staff table). All inputs are validated. It is assumed that each employee has a unique phone number and id that is not shared between them. Passwords use client side validation and must be at least 8 characters, have one upper case letter and one number. All forms that update tables have an additional confirm pop up message.

Accessed from: employeeaction.php when Change Employee Details selected

Sends user to:

Nav bar: employeeaction.php, admin.php, logout.php redirecting to index.php

change_password.php

Purpose: the form that allows the admin to change a driver's password. The same password is not allowed to be entered, a new one must be created. Passwords use client side validation.

Accessed from: authentication.php when admin password successfully input

Sends user to:

Nav bar: employeeaction.php, admin.php, logout.php redirecting to index.php

Redirects to: confirm_pass.php (confirmation page)

change_price.php

Purpose: the form that allows the admin to change the rental price of a selected item. All fields are validated. Number fields use both client and server side validations. All prices are taken to be in euro.

Accessed from: equipaction.php when Update Rental Price is selected

Sends user to:

Nav bar: equipaction.php, admin.php, logout.php redirecting to index.php

Note: **change_setup.php** changes the setup price per equipment

update_charges.php allows the admin to update vat rate, delivery rate and delivery rate per kilometre

all redirect to the same pages as change_price through the navigation bar and all are accessed through equipaction.php by clicking the relevant buttons

checkcustomer.php

Purpose: checkcustomer.php redirects the user to the appropriate page, depending on whether they are logged in or not

Accessed from: orderequipment.php

Sends user to:

invoice.php - if user is logged in, they are sent straight to invoice.php

loginghost.php or registerghost.php - if user is not logged in, they can choose to login as an existing customer or register as a new one

- Customer is logged-in if sessionvariable['user_email'] is set
- If the session variable is set, user is redirected directly to invoice.php
- If the session variable is null, checkcustomer.php displays a link to loginghost.php or registerghost.php. The customer can choose to login as an existing customer or register as a new customer. The user email for events is set in these pages .

checkorderstatus.php

Purpose: form to allow customer to view logged times made by staff (when they have loaded the van, and left)

Accessed from: pastorder.php

- A sql query uses an inner join on staff and event_hours tables where the event id is equal to the session variable for user_email
- The session variable has been set when the customer logged in (login.php) or registered (register.php).
- If no results are found, the user is told that the event is not yet underway.
- If there are results, the user can view the staff member assigned to their event delivery, the time the van was loaded, and the time the van departed.

- A for loop is used, so if the event requires multiple vans and drivers, all of them will be displayed.

customeraccount.php

Purpose: a kind of home page for customers

Accessed from:

login.php
registration.php

Sends user to:

newevent.php - create new event
pastorders.php - view past orders, and check delivery status
equipment.php - view catalog

- Page contains a welcome message for customers (retrieving name from SQL database) as well as links for the three pages above

delete_empl.php

Purpose: the form that allows the admin to remove a selected employee (only those who are drivers). Once successfully deleted after form validations are satisfied, the user is redirected to a confirmation page. Both the employee's record in the staff table and their future shifts are deleted.

Accessed from: employeeaction.php when Remove Employee is selected

Sends user to:

Nav bar: employeeaction.php, admin.php, logout.php redirecting to index.php

Redirect: confirm_deleempl.php after successful completion

deleteequip.php

Purpose: the form that allows the admin to remove a selected equipment from the catalogue. The user is redirected to a confirmation page upon successful deletion to prevent other items deleting given the nature of select input types.

Accessed from: equipaction.php when Delete Equipment is selected

Sends user to:

Nav bar: equipaction.php, admin.php, logout.php redirecting to index.php

Redirect: confirmdelete.php after successful completion

deleteshifts.php

Purpose: Display all shifts already assigned on a given day

Allow the admin to remove shifts that have been created on assigndriver.php

Accessed From: assigndriver.php

Sends User To: opens to new tab - no links

deliverycharges.php

Purpose: allow customer to view delivery charges by count

Accessed from: index.php, newevent.php, allequipment.php,equipmentcategory.php

dropofflist.php

Purpose: Staff side page

- Only accessible if the signed-in employee is currently clocked in to a shift
- When a user is clocked in, staff.php will display all events to be serviced
- dropofflist.php displays all equipment requirements for each event which are outgoing (Delivery Drop-Off / Collection Out-Going) scrolling function using next/prev

Accessed From: staff.php (only if clocked into shift)

Sends User To: opens to new tab - no links

editbrokenitems.php

Purpose: One of 3 key pages for logging broken returns or unreturned items after events

- Can be accessed from both staff and admin side of the website
- Both options offer different links but provide the same functionality and output
- This page gives the user an option to fully delete individual broken equipment inputs

Accessed From:

Staff Side:

brokenitems2.php, brokeninput.php

Admin Side:

verifybrokenitems.php, brokenitems2.php

Sends User To:

Staff Side:

Nav Bar: staff.php, myhours.php, brokeninput.php, logout.php
brokeninput.php
brokenitems2.php

Admin Side:

Nav Bar: staffhours.php, verifybrokenitems.php, editbrokenitems.php, logout.php
verifybrokenitems.php
brokenitems2.php

editorder.php

Purpose: allows customer to remove a product from their cart (using a delete for the sql equipment_rented table)

Accessed from: orderequipment.php

Sends user to: orderequipment.php

- A select/where query displays all the products and quantities in a dropdown. A where clause is used using the session variable of event_id, which was set in newevent.php.
- The customer chooses a product and clicks remove to delete that row from equipment_rented table.
- The user only has the option to remove all quantities of a product, introducing the option to remove some of a quantity is a feature that could be added with more time.
- The page is refreshed, so that if a customer wants to remove another item, the item just deleted is not displayed.

employeeaction.php

Purpose: a page that displays links to various forms and reports relating to employee details and work

Accessed from: admin.php when Employee Actions is selected and the following pages

- assigndriver.php
- authentication.php
- change_employee.php
- delete_empl.php
- newstaffmember.php
- staffhours.php

Sends user to:

Nav bar: assigndriver.php staffhours.php, newstaffmember.php admin.php, logout.php redirecting to index.php

Redirect: The pages in accessed from when the corresponding link is selected

equipeaction.php

Purpose: a page that displays links to various forms and reports relating to the equipment and other relevant operations

Accessed from: admin.php when Operations and Equipment is selected and the following pages

- change_price.php
- change_setup.php
- confirmdelete.php
- deleteequip.php
- new_equip.php
- new_quant.php

- update_charges.php

Sends user to:

Nav bar: admin.php, equipment.php, logout.php redirecting to index.php

Redirect: The pages in accessed from when the corresponding link is selected and equipment.php

equipment.php

Purpose: allows user to specify whether they want to view the catalogue by category or to view the whole thing

Accessed from: customeraccount.php

Sends user to:

equipmentquantity.php - if user wants to view by a specific quantity

allequipment.php - if user wants to view all equipment

- This page includes a form to select a category option within equipment. A select statement is used to fetch the unique categories from the equipment table. The customer can then choose one. The form sends the user to equipmentcategory.php, the category is posted as 'category'
- The second option on this page is a button which will bring the customer to allequipment.php, an option to view the entire catalog.

equipmentcategory.php

Purpose: allows user to specify how to want to view catalog by a specific quantity

Accessed from: equipment.php

Sends user to: deliverycharges.php - customers can view deliverycharges.php in a new tab

- Displays all equipment details from a specific category in a table
- The category has been posted by equipment.php, and a select/where statement is used to display all equipment in specified category
- If there is a sale, this is displayed in a column, otherwise nothing is displayed.

finishinputshifts.php

Purpose: For closing shift inputs (assigning events to drivers/shifts)

- Unsets all stored session variables and returns user to assigndriver.php
-

Accessed From: assigndriver.php

Sends User To: assigndriver.php

Forgotpassword.php

Purpose: If a user has forgotten a password, they can be sent a one time password that allows them to reset it

Accessed from: login.php and logginghost.php

Sends user to: change password.php

- The user inputs their email address in a form. They press “send link”. If the email is in the database a one time password (6 digit number) is generated and emailed to them.
- They can enter the OTP and if it is a match to the one generated in php, they are sent to changepassword.php. The email address to change is stored in a session variable.
- Once changed, an email is sent to the customer.

changepassword.php

(this is for customers, a staff version exists change_password.php accessible to admins only)

Purpose: Allows the customer to change their password. They have already been validated using an OTP in forgotpassword.php

Accessed from: forgotpassword.php

Sends user to: passwordsuccess.php

The user's password and confirm password contain the same validations as registration.php (see file). An update query updates the password in the database.

incompleteshifts.php

Purpose: Display all shifts for which the scheduled date has expired and no inputs exists for clock-in or clock-out (uses scrolling next/prev function to move between incomplete shifts)

Allows the user to 'alter' shift hours or 'delete' shift if unfulfilled

Accessed From: staffhours.php, admin.php

Sends User To:

Nav Bar: admin.php, staffhours.php, verifybrokenitems.php, logout.php

login.php

Purpose: This is the customer login page

Accessed from: index.php

Sends user to: customeraccount.php

- login.php checks that a customer is in the database by checking their email and password. They are sent to their homepage assuming they login in correctly.
- The session variable user_email is set.

loginhost.php

Purpose: allow customers who are registered already, but chose to create an event before logging in to log in

Accessed from: checkcustomer.php

Sends user to: invoice.php

- Please see notes for login.php. The code is similar with the following adjustments.
- A different message is displayed - "Login to complete your order"
- The user is sent to invoice.php to complete their order
- The purpose of this page is to allow customers who are registered already, but chose to create an event before logging in to log in
- They cannot confirm their event until they have logged in, as there is no email address to store in events.
- Once they login, the session variable for user_email is now set, the event table is updated with customer email (email previously null), and the customer is redirected to invoice.php.
- Note: could potentially merge this into login.php, and use if statements to determine where to send customer

logout.php

Purpose: Logs the user out across the site and destroys the session before returning user to the index page

Accessed From: All user non pop-up sites

Sends User To: index.php

myhours.php

Purpose: Display working history for logged in staff member

- There are 2 tables:
 1. Show date & hours and sum totals hours
 2. Show date, clock-in/clock-out, # dropoffs, # pickups

Accessed From:

Nav Bar: staff.php, brokenitems.php (all broken items staff side pages)

Sends User To:

Nav Bar: staff.php, brokenitems.php
stafflogin - if user has timed out / is not logged in

new_equip.php

Purpose: a form to allow admin to add more equipment to the catalogue. A MySQL insert statement is used once all the user inputs have been correctly validated. If there is no setup cost or no quantity on hand, 0 must be entered into those fields. Additionally, number fields have both client side and server side validation. Using php echo statements, a confirmation of the equipment added is displayed at the top of the page following successful submission.

Accessed from: equipaction.php when New Equipment is selected

Sends User to:

Nav Bar: admin.php, equipaction.php, logout.php redirecting to index.php

Note: new_quant.php updates the quantity of an item on hand

change_price.php updates the rental price of an item on hand

change_setup.php updates the setup price of an item on hand

update_charges.php change the vat, delivery rate or delivery rate per/km

All redirect to the same pages from the navigation bar and all are accessed through equipaction.php by selecting the relevant links. Similar code is also used among these pages.

newstaffmember.php

Purpose: allows admin to add a new staff member to the staff table. There are only two positions available: driver and manager, but more may be added as necessary. Elements of the form, relating to passwords make use of client side validations. A strong password that is at least 8 characters, has 1 upper case letter and 1 number is required. An on page confirmation message similar to new_equip.php is displayed following successful submission.

Accessed from: employeeaction.php when Add New Employee is selected

Sends User to:

Nav Bar: employeeaction.php, admin.php, logout.php redirecting to index.php

nextpage'...' .php

Purpose : Several nextpage'...' .php exist. They are used for scrolling through pages. They increment a \$count session variable which corresponds to a page number (SQL row)

Links: nextpage.php -> verifybrokenitems.php (increments count)

nextpageassigndriver.php -> assigndriver.php (increments date by 1 day)

nextpageassignevents.php -> assigndriver_equipmentsList.php (increments count)

nextpagedrop.php -> dropofflist.php (increments count)

nextpageincompleteshifts.php -> incompleteshifts.php (increments count)
nextpagepick.php -> pickuplist.php (increments count)

newevent.php

Purpose: form to create a new event

Accessed from:

index.php/home - in which case the email for the events table is set to null until they login (loginghost.php) or register (registerghost.php)
customeraccount.php - in which case the email is set already in a session variable, and can be inputted to the events table

Sends user to:

orderequipment.php - where customers select equipment for the event just created
deliverycharges.php - customers can view deliverycharges.php in a new tab

- The user selects fields for the events table such as the name, location, county, and whether they want delivery or collection.
- Please see the structure of events.php. Fields for the event table include start_date, start_time, end_date, end_time and delivery_status.
- A delivery_status of 1 indicates the customer wants equipment delivered. In this case, start_date, start_time, end_date, and end_time indicate the date the customer wants the equipment delivered and collected by DPH.
- A delivery_status of 0 indicates the customer wants to collect equipment themselves. In this case, start_date, start_time, end_date, and end_time indicate the date the customer wants to pick up and collect equipment.
- Certain validations are included such as: "all fields are required", "the end date cannot be before the start date", and "the end date cannot be after the due date".
- The due date is calculated by modifying the start date by adding 5 days. This means the customer has to return their equipment 5 days from the start date.
- If the start date and end date are the same, the end time must be before the start time.
- Assuming no errors, the event_id, delivery, start_date and end_date are all stored as session variables, for use in orderequipment.php. The event is inserted into SQL database and user is sent to orderequipment.php.

Orderequipment.php

Purpose: form to allow customer to order equipment for event just created

Accessed from: newevent.php

Sends user to:

editororder.php - page option to remove equipment
orderequipmentsetup.php - choose setup for equipment if offered, and delivery if the customer selects delivery
vieweventequipment.php - if the customer is happy with order, this page gives them a final view of their order, as well as the option to print their invoice

- The first display is the items on sale, selected from mysql using a `SELECT * from equipment where sale > 0`
- Next, the customer chooses a category in a form. When the customer submits the category a second dropdown appears.
- The customer chooses here a quantity and a product (only products in the specified category are shown, by using a select/where statement, using the product pushed from the first form)
- The dropdown for the product name also displays the rental price. An if statement displays the sale alongside the name and rental price if it is on sale. The customer can press a second submit
- PHP then calculates the quantity available. First, it looks at the equipment_rented table. It gets the total quantity of equipment rented already but at dates that don't crossover. Then it gets the total equipment already rented at any time. The total products rented minus the total products rented at dates that don't crossover is saved as \$unavailablequantity. Finally this is taken away from the total quantity on hand from the equipment table and saved as \$availablequantity.
- An error message is displayed if the customer tries to order a quantity over this amount.
- If there are no errors, the customer is then sent to orderequipmentsetup.php, if setup is offered, and they have selected delivery. The quantity and product id are sent in as session variables.
- Otherwise, the new order is added to equipment rented. If a user is selecting a product they already have in their basket (already a row in equipment_rented), the quantity is incremented by the desired amount.
- The rental price, setup price, and sale are all added to equipment_rented as a customer adds equipment. This is because these can all be changed, and when we want to display an invoice for a past event, we want to be able to take the rental price and sale paid at the time the customer made the order. Customers have the option to see past invoices from their homepage. If we just took these prices from the equipment table, they could have been changed, and the customer would see a different price to the one they 'paid' at the time they ordered.
- A shopping cart is displayed at the bottom. The code is modified from invoice.php.
- There is an option for the user to remove equipment in editorder.php

orderequipmentsetup.php

Purpose: form to allow customer to select setup for equipment if offered, and if the customer selects delivery

Accessed from: orderequipment.php

Sends user to: orderequipment.php

- The setup price of the product is displayed. The productid and quantity have been saved from orderequipment.php as session variables.
- The customer is sent a form which offers them setup or not.

- A row is added to equipment_rented table, with 1 (setup) or 0 (no setup) added for the setup column depending on their form answer.
- If a user is selecting a product they already have in their basket, the quantity is incremented by the desired amount. It is assumed that the customer can't choose to have some of a product setup. Eg: if they order 4 tables, they can't pay for two of them only to be set up. Setup is set as whatever the customer selected the second time.
- Customer is sent back to orderequipment.php

pastinvoice.php

Purpose: display invoice for an event created in the past

Accessed from: pastorders.php

- The code is modified from invoice.php.
- The events_id has been selected in pastorder.php and is pushed as 'rentals'.
- The VAT rate and delivery charge are not newly calculated from the delivery_charges table. They are taken from the receipts table, which contains the delivery charge and vat rate paid at the time of order.
- If we newly calculated them, the invoice might display a different price than the original one.
- Therefore it is important that we have saved the original vat rates and delivery_charges. (see insertreceipts.php).
- The same goes for the rental and setup prices, and sale. The original prices are saved in equipment_rented. We don't take the current prices from the equipment table, as these can be changed regularly.
- The customer is given the option to re print.

pastorders.php

Purpose: displays all events a customer has, allows them to view past invoices and check a delivery status

Accessed from: customeraccount.php

Sends user to:

pastinvoice.php - allow the customer to view a past invoice
checkorderstatus - allow the customer to see if their equipment is loaded etc.

- This page displays all the events a customer has booked at any time with DPH. This is displayed in a table. The select/where statement uses the session variable cust_email, which has been set when the customer logged in (login.php) or registered (register.php).
- There is also a form that allows you to choose an event in order to view a past invoice. All the events a customer has registered for are displayed in a dropdown. The customer picks one and the id of the selected event is posted to pastinvoice.php as the variable rentals.

- The final feature of this page is a form that allows you to choose an event in order to view the delivery status of an item. All the events a customer has registered for are displayed in a dropdown. The customer picks one and the id of the selected event is posted to checkorderstatus.php as the variable event.

pickuplist.php

Purpose: Staff side page

- Only accessible if the signed-in employee is currently clocked in to a shift
- When a user is clocked in, staff.php will display all events to be serviced
- pickuplist.php displays all equipment requirements for each event which are incoming (Delivery Pick-Up / Collection Returns) scrolling function using next/prev
-

Accessed From: staff.php (only if clocked into shift)

Sends User To: opens to new tab -

brokenitems.php - if pickup events exist for the shift

prevpage'...' .php

See nextpage'...' .php for details

Only accessible if count is greater than the lower limit

registerghost.php

Purpose: allow customers who are not registered already, but chose to create an event before registering to log in

Accessed from: checkcustomer.php

Sends user to: invoice.php

- Code modified from registration.php.
- A different message is displayed, and the customer is redirected to invoice.php instead of customeraccount.php
- Note: could potentially merge this into registration.php, and use if statements to determine where to send the customer.
- Once a user is registered with no errors, the event they have just created is updated. The email (previously null) is now set to the customer's email.

registration.php

Purpose: form for user to register as customer

Accessed from: index.php, login.php

Sends user to: customeraccount.php

- Contains multiple validations. The email must include certain elements, such as capital letters and numbers, javascript is used to display a message for all the password requirements, which turn green as they are included
- Confirm password cannot be pasted in
- A welcome message is sent to the new user, and the customer is added once there are no errors

shiftdetails.php

Purpose: Admin has selected a shift from staffhours.php to be further investigated

- This page displays a table with the shift date and shows the scheduled clock in / clock out time against the actual hours input by the employee
- Admin can then alter the hours
- A table is also shown below the events serviced on the shift with the tracker details of when vans were loaded and when they left etc

Accessed From: staffhours.php

Sends User To: opens new tab - no links

staff.php

Purpose: Staff (drivers) home page

- Displays all upcoming shifts that are set by admin on assigndriver.php
- If shift exists 'today', then option to clock in is displayed
- If Clock in is selected, display all events for the shift
If events are deliveries (ie off-site dropoff/pickup) an option to track delivery is shown (vans loading / enroute)
Users can also find links for equipment lists relating to events for the current shift
- User can then clock out and the display returns to all upcoming shifts

Accessed From: stafflogin.php, myhours.php & nav bar across all broken returns pages

Sends User To:

stafflogin.php (if user has timed out or if clocked into shift who's date has expired, ie forgot to clock out)
dropofflist.php - if user clocked-in to shift
pickuplist.php - if user clocked-in to shift

staffhours.php

Purpose: Admin Page for seeing employee history

- Admin first sees all staff with total hours, start date and number of shifts
- Admin can then narrow down results by using date range or employee selection
- If individual employee is selected, a breakdown of their shifts is shown (for the given date range), and offers a link for each shift to see further information and alter hours

Accessed From: admin.php

Sends User To:

Nav Bar - admin.php, incompleteshifts.php, verifybrokenitems.php, logout.php
shiftdetails.php (if employee selected)
incompleteshifts.php

stafflogin.php

Purpose: staff side login page

Accessed From: index.php

Sends User To:

admin.php (if user is an admin)
staff.php (if user is a staff member; one of our assumption is only 2 roles if more roles exist this can easily be changed)

thankyouevent.php

Purpose: thank our valuable customers for their custom

Accessed from: invoice.php

sends user to: customeraccount.php

verifybrokenitems.php

Purpose: Admin side verification for broken items

- Admin can see a list similar to brokeninput.php which displays all broken items stored by the staff member
- They can scroll (next/prev) through lists if more than one event is outstanding to be verified
- They have links to brokenitems2.php and editbrokenitems.php (admin side) to edit inputs
-

Accessed From: staffhours.php

Sends User To: brokenitems2.php, editbrokenitems.php, staffhours.php, logout.php

SQL Tables

*A brief display of the sql tables, their fields and explanations if required
Please see SQL data dump for more extensive information.*

Broken Returns	
equipment_id event_id quantity verified	An admin must verify all broken returns 1=verified 0 = not verified

Customers	
id name address eircode phone email is_business user_password	auto-generated in sql 1=business 0 = private customer

delivery_charges	
chargetype county dist_from_dub_km flat_rate rate_per_km	eg delivery, vat eg 75 euro rate per km outside of dub eg 0.75 cents

equipment	
id category product_name rental_price_excl_vat setup_price_excl_vat quantity setup sale	 1=setup offered 0 = setup not offered in decimal form. 0.30 = 30% off

equipment_rented	
equipment_id	
events_id	
quantity	
setup	1=yes 0=no
sale	sale at time of order
rental_price	rental price at time of order
setup_price	setup price at time of order

events	
event_id	
event_name	
cust_email	
location	
county	
start_date	collection or delivery date
end_date	drop off or pick up date
start_time	
end_time	
delivery_status	1=delivery 0 = collection

event_hours	
event_id	
staff_id	
vans_loading	time van loaded
vans_enroute	time van left dph
registration	of van
dropoff_pickup	shift related to
shift_id	

receipts	
event_id	
event_cost	cost at time of order
purchase_date	date of order
vat_charged	cvat rate at time of order
delivery_charged	delivery at time of order

staff	
employee_id	
full_name	
phone_number	
posistion	
password	
on_shift	
	1=yes 0 = no

staff_shifts	
shift_id	
staff_id	
shift_date	
clock_in_time	
clock_out_time	
set_clock_in	
set_clock_out	
	actual
	actual
	set
	set

vans	
registration_no	
capacity	

List of all filenames

Sources for images are included in the code comments

abstract-light-celebration-background-defocused-260nw-1115360159.webp
addsale.php
admin.php
allequipment.php
assigndriver.php
assigndriver_equipmentlist.php
authentication.php
bestcust.php
brokenbackpage.php
brokeninput.php
brokenitems.php
brokenitems2.php
cart-icon.png
changepassword.php
change_employee.php
change_password.php
change_price.php
change_setup.php
checkcustomer.php
checkorderstatus.php
confirmdelete.php
confirm_deleempl.php
confirm_pass.php
customeraccount.php
deleteequip.php
deleteshifts.php
delete_empl.php
Deliverycharges.php
dph.jpg
detail.php
dropofflist.php
editbrokenitems.php
editorder.php
employeeaction.php
equipaction.php
equipment.php
equipmentcategory.php
face.jpg
filenames.txt
filenamesdhl.txt
finishinputshifts.php
forgotpassword.php
incompleteshifts.php
index.php
insertreceipt.php
invoice.php

login.php
loginghost.php
logout.php
marquee.jpg
myhours.php
newevent.php
newstaffmember.php
new_equip.php
new_quant.php
nextpage.php
nextpageassigndriver.php
nextpageassignevents.php
nextpagedrop.php
nextpageincompleteshifts.php
nextpagepick.php
orderequipment.php
orderequipmentsetup.php
party-header.jpg
partypopper.jpg
passwordsuccess.php
pastinvoice.php
pastorders.php
pickuplist.php
prevpage.php
prevpageassigndriver.php
prevpageassignevents.php
prevpagedrop.php
prevpageincompleteshifts.php
prevpagepick.php
productsales.php
registerghost.php
registration.php
rentalfreq.php
reports.php
sale.png
shiftdetails.php
staff.php
staffhours.php
stafflogin.php
test.php
totalproductsales.php
thankyouevent.php
update_charges.php
verifybrokenitems.php