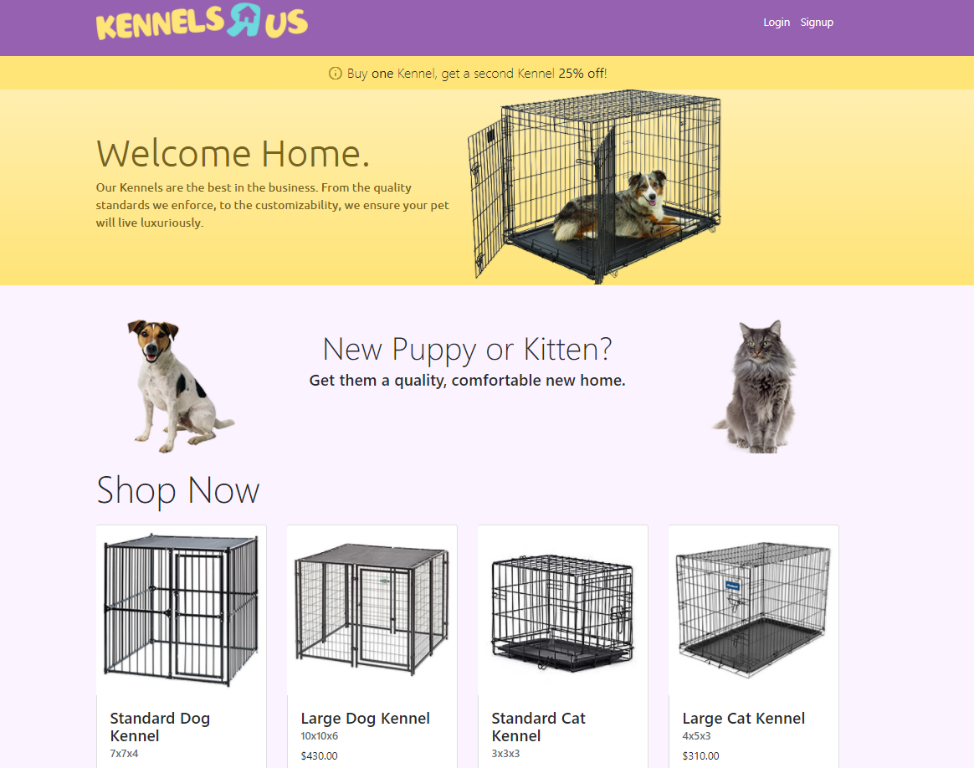


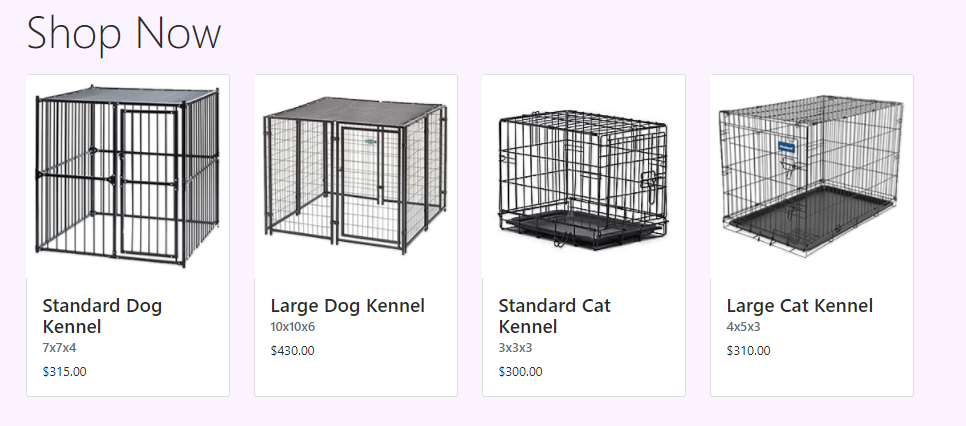
User Manual

**SYSTEM OVERVIEW:**

**The customer side** contains an index page which displays a summary of the cages offered by Kennels R US.



The kennels offered include: a standard dog kennel, large dog kennel, standard cat kennel, and a large cat kennel.



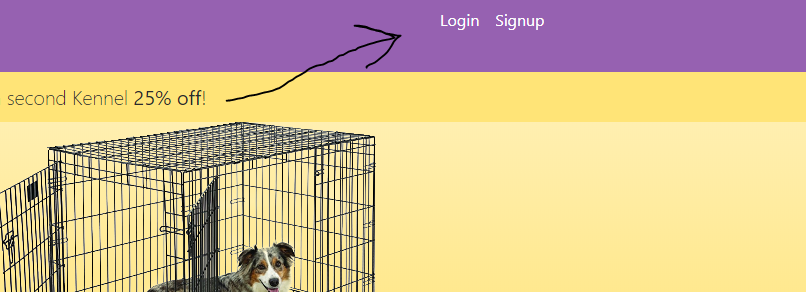
Upon clicking one, the user is redirected to an account creation and login page, unless the user is logged in already.



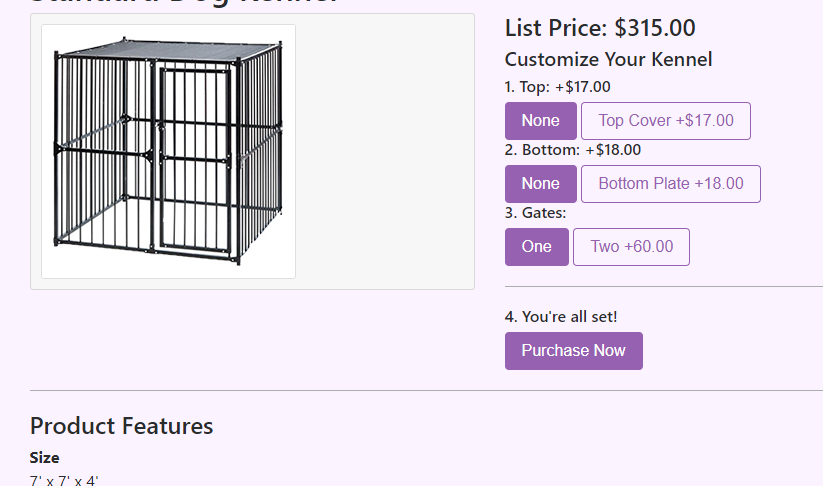
The account creation process collects customer information, generates an id, and stores the information in the database to be used later.



The login and registration pages can be clicked on directly on the top right:

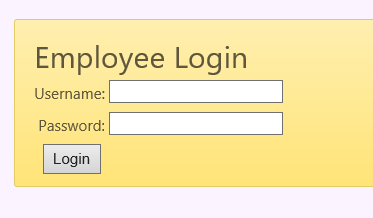


Once logged in, users can click on any of the cages from the index to be directed their corresponding customizations. Details for the cages are listed below the cages:



Once options for cage customization are met, just click on purchase now to be directed to the payment page to complete the order.

For **the employee side**, employees will need their own respective login information. They will also use a different address:



kennelsremployee.22web.org

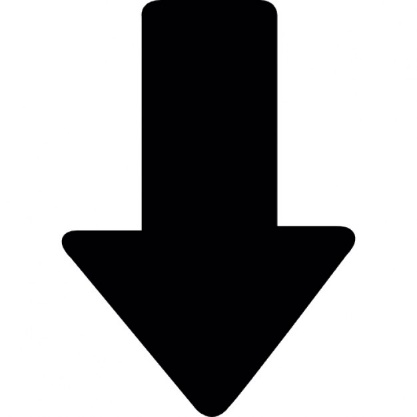
Once logged in, employees can manage their respective pages.

**Managers** can update order statuses, create purchase orders, pay vendors, generate sales reports, generate customer reports, generate inventory reports, and print shipping labels.

**Sales** can generate sales reports and customer reports.

**Production Supervisor** can update order statuses, print shipping labels tied to orders, and generate inventory reports within defined parameters.

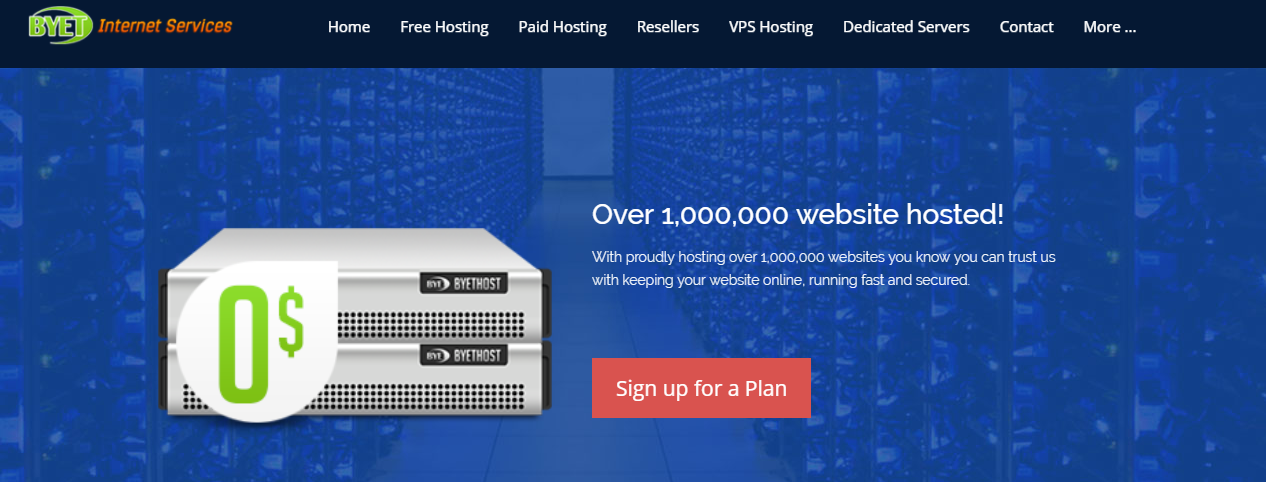
**Accounting** workers can create purchase orders, pay vendors, and process order statuses.

****

**SYSTEM REQUIREMENS AND SETUP INSTRUCTIONS:**

For these pages to operate correctly, a webhost is required. In these examples, we used Byethost’s server renting service, which includes servers for PHP and SQL too.

Proceed to <https://byet.host> to register an account and get started.



Once an account is created once can begin uploading the database and website files. We recommend purchasing premium account membership for benefits such as

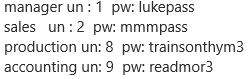
Since this information is stored outside of the company building, one does not need specific components other than a machine by which to access employee data.

We recommend purchasing premium hosting for better speeds and benefits.

**OPERATOR INSTRUCTIONS:**

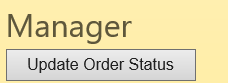
Use this webpage for the employee login:



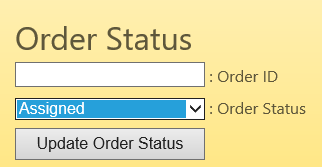


Follow this chart to login to the desired user.

**Manager:** *To**update an order status,* click “Update Order Status,”



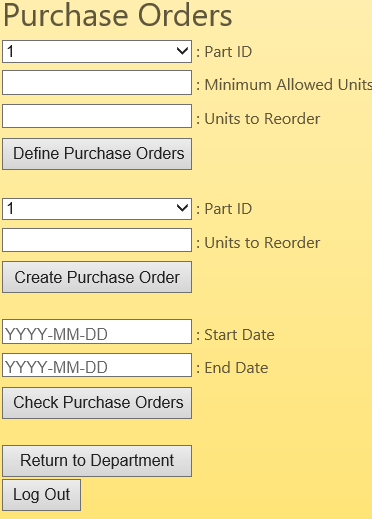
Then enter the order number and change the status to whatever is desired.



*To manage purchase orders*, click “Purchase Orders.”

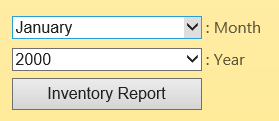


From here, one can define purchase orders for a specific part ID by entering data at the top.



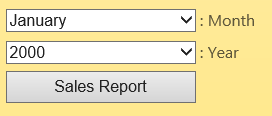
Below the define area, one can create purchase orders for part IDs and check existing purchase orders within a date range.

*For inventory reports*, click on the “Inventory Report” button.

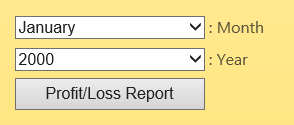


Reports are automatically displayed on the loaded page.

*For sales reports*, simply click on “Sales Report” button.



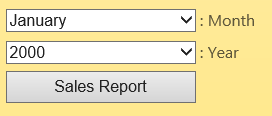
*To generate a profit/loss report*, click the “Profit/Loss Report” button.



*Finally, to print a shipping label*, click “Print Shipping Label”

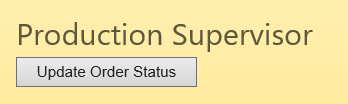


**Sales:** For sales reports, simply click on “Sales Report”



Reports are automatically displayed on the loaded page.

**Production:** *To**update an order status,* click “Update Order Status.”

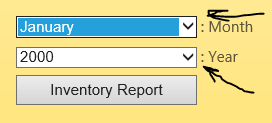


Then enter the order number and change the status to whatever is desired.

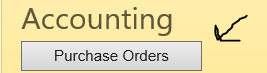
*To print a shipping label*, enter the order number click “Print Shipping Label”.



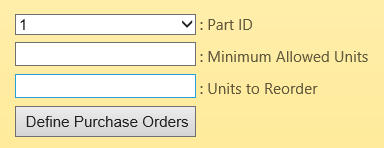
*To receive an inventory report*, click the dropdown menu for month and year and then click “Inventory Report”.



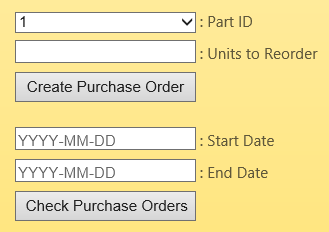
**Accounting:** *To manage purchase orders*, click “Purchase Orders.”



From here, one can define purchase orders for a specific part ID by entering data.



With this information one can create purchase orders for part IDs and check existing purchase orders within a date range.

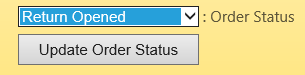


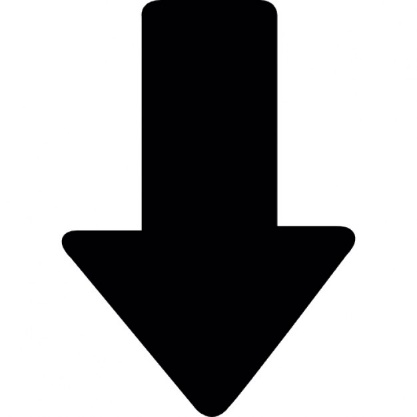
Observable here

*To**update an order status,* click “Process Orders”



Enter the order number and then change the status to whatever is desired.



****

**TABLE INFORMATION:**

**Customer Table:**

This table holds information on customers. cust\_id -int(11) is a column that tracks customers by a unique primary key that is generated on registration. cust\_last\_name - varchar(30) and cust\_first\_name - varchar(30) store information containing the customer’s name. cust\_phone – varchar(12) and cust\_email – varchar(50) are used for the customer’s contact information. cust\_street – varchar(50), cust\_city – varchar(30), cust\_state – char(2), and cust\_zip - varchar(10) contain the customer’s address information.

**Customization:** This table holds information on customizations options offered. *custom\_id – int(11)* is a column that uniquely identifies each customization. It also allows for the generation of bills of materials when used in conjunction with the *Customization\_Details* table. *custom\_name – varchar(60)* stores a description of the customization. *custom\_price – decimal(10,2)* lists the sale price of a customization. *custom\_cost – decimal(10,2)* lists the cost in parts of a customization to be produced.

**Customization\_Details:** This table contains details on what a given customization contains. *custom\_id – int(11)* is a foreign key from the *Customization* table which allows the user to create bills of materials. *part\_id – int(11)* is a foreign key from the *Inventory* table which identifies which parts are used in creating a customization. *custom\_quantity – decimal(5,2)* displays how many of a given part are used in the construction of a customization.

**Department:** This table contains information on the departments within the company. *dept\_id – int(11)* contains the unique identifier for a given department. *dept\_name – varchar(25)* describes the department.

**Employee:** This table contains information on the employees of the company. *emp\_id - int(11)* is a column that tracks employees by a unique primary key that is generated on employment. *emp\_last\_name - varchar(30)* and *emp\_first\_name - varchar(30)* store information containing the customer’s name. *dept\_id – int(11)* is a foreign key from the *Department* table that shows to which department an employee belongs. *emp\_phone – varchar(12)* and *emp\_email – varchar(50)* are used for the employee’s contact information. *emp\_street – varchar(50), emp\_city – varchar(30), emp\_state – char(2), and emp\_zip - varchar(10)* contain the employee’s address information. *hire\_date – date* records the date on which the employee was hired by the company. *currently\_employed - varchar(3)* shows whether an employee is currently employed with the company. *retire\_date – date* records the date at which the employee retired from the company. *emp\_password – varchar(20)* stores the password set for the employee used to log in to the employee site.

**Inventory:** *part\_id – int(11)* is a unique key that identifies parts used for the construction of both kennels and their customizations by the company. *part\_name – varchar(40)* describes the parts. *part\_price – decimal(10,2)* records the price paid by the company for a given part. *units – decimal(10,2)* keeps track of the number of units on hand for a given part. *unit\_of\_measure – varchar(10)* shows what units of measure for parts in the table. *supplier\_id – int(11)* is a foreign key that shows which supplier the company orders a part from. *minimum – decimal(10,2)*  stores the level of inventory at which another parts order needs to be placed.

**Model:** This table holds information on kennel models offered. *model\_id – int(11)* is a column that uniquely identifies each kennel. It also allows for the generation of bills of materials when used in conjunction with the *Model\_Details* table. *model\_name – varchar(60)* stores a description of the model. *model\_price – decimal(10,2)* lists the sale price of a kennel. *model\_cost – decimal(10,2)* lists the cost in parts of a model to be produced.

**Model\_Details:** This table contains details on what a given model contains. *model\_id – int(11)* is a foreign key from the *Model* table which allows the user to create bills of materials. *part\_id – int(11)* is a foreign key from the *Inventory* table which identifies which parts are used in creating a model. *model\_quantity – decimal(5,2)* displays how many of a given part are used in the construction of a model.

**Orders:** This table contains the information on the orders placed by customers. *order\_id – int(11)* is a unique identifier that is generated when an order is placed. *cust\_id – int(11)* is a foreign key from the *Customer* table that shows which customer placed the order. *order\_date – date* shows the date at which an order was placed. *ship\_date – date* records when an order was shipped to a customer. *ship\_fee – decimal(10,2)* records the shipping fee for an order, if any exists. *ship\_name – varchar(50)* records the name of the recipient. *ship\_street – varchar(50), ship\_city – varchar(30), ship\_state – char(2), and ship\_zip – varchar(10)* store the address destination for an order. *credit\_card – int(16)* stores the credit card number used to make an order. *status\_id – int(11)* is a foreign key from the *Order\_Status* table which shows the status of an order.

**Order\_Details:** This table contains details on which products were desired in each order. *order\_id – int(11)* is a foreign key from the *Orders* table that ties products to an order. *model\_id – int(11)* is a foreign key from the *Model* table that displays which kennel was desired. *custom\_id – int(11)* is a foreign key from the *Customization* table that shows if the customer wished to add any customizations to their order. *order\_quantity - int(11)* shows how many of a listed product were requested. *discount – int(11)* shows if any discount was allowed on the order.

**Order\_Status:** This table shows the different statuses that an order can have. These statuses allow for an order to be designated as open, assigned to production, closed, filed as a return, and confirmed as a return. *status\_id – int(11)* is a column that uniquely identifies a status. *status\_name – varchar(20)* describes a status.

**Part\_Orders:** This table keeps a record of all the part orders placed. *placed\_date – date* shows when a given part order was placed. *part\_id – int(11)* is a foreign key from the *Inventory* table that shows which part was ordered. *units\_received – decimal(5,2)* shows how many units of a part were ordered. *received\_date – date* shows the date at which a parts order was received. *supplier\_id – int(11)* is a foreign key from the *Suppliers* table that shows to which supplier an order was placed.

**Suppliers:** This table shows the different suppliers from which the inventory is ordered. *supplier\_id – int(11)* is a column of unique identifiers for the suppliers. *supplier\_name – varchar(60)* displays the name of a given supplier. *contact\_name – varchar(50)* shows the name of the contact representative for a supplier. *contact\_phone – varchar(12)* and *contact\_email – varchar(50)* shows the contact information for the contact representative.