Step by step process

First I created a SQL database using MySQL workbench and created a database named pets. I then made a table called products and added 4 pets to it (partially shown below).

✓	8	14:24:11	select * from pets.product LIMIT 0, 1000	1 row(s) returned
✓	9	14:24:26	INSERT INTO pets.product (id, name, color, price) VALUE (2, "Hamster", "Brown", 30.00)	1 row(s) affected
✓	10	14:24:26	INSERT INTO pets.product (id, name, color, price) VALUE (3, "Cayman", "Neon Pink", 500.00)	1 row(s) affected
✓	11	14:24:26	INSERT INTO pets.product (id, name, color, price) VALUE (4, "Tarantula", "Red", 100.00)	1 row(s) affected

Next I created a dynamic web project on Eclipse and added the mySQL jar file to the path. I then created index.html and added an href, text box, and submit button to the page.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <meta charset="UTF-8">
5  <title>Retrieve Product Details using ID</title>
6  </head>
7  <body>
8  <a href="list">Click here to view the table</a><br><br>
9      <form name="loginForm" method="post" action="list">
10     Enter the product id number of the pet you would like to search for:<br> <
11     <input type="submit" value="Submit" />
12 </form>
13 </body>
14 </html>
15
```

I created a new class called DBConnection to establish the connection to the server. Then I created a new class called ProductDetails which uses JDBC to perform actions on the webpage. I modified the doGet method to query the table and get the information, then find what the user typed into the box. If the search box contents were null then the contents of the table are displayed on the page. Otherwise the row corresponding to the id number is printed to the page.

```
51                  Statement stmt = conn.getConnection().createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE, ResultSet.CONCUR_READ_ONLY);
52                  stmt.executeUpdate("insert into eproduct (name, price, date_added) values ('New Product', 17800.00, now())");
53
54                  //query the table and get all information
55                  ResultSet rst = stmt.executeQuery("select * from pets.product");
56
57                  //find what the user typed into the search box
58                  String productSearch = request.getParameter("search");
59                  //out.println(productSearch);
60
61                  //user hasn't typed anything so display table
62                  if(productSearch == null)
63                  {
64                      out.println("The following are the elements in the Pets table" + "<Br>" + "<Br>");
65                      //simple while loop to print all elements in table
66                      while (rst.next()) {
67                              out.println(rst.getInt("ID") + ": " + rst.getString("color") + " "
68                                      + rst.getString("name") + " costs: $" + rst.getDouble("price") + "<Br>");
69                      }
70                  }
71                  //user typed something
72                  else
73                  {
74                      //select the row corresponding to the id number
75                      String sql_res= "select * from pets.product where id=" + productSearch;
76                      ResultSet inTable = stmt.executeQuery(sql_res);
77
78                      //if not empty then print all product details
79                      if(inTable.next())
80                          out.println(inTable.getInt("ID") + ": " + inTable.getString("color") + " "
81                                  + inTable.getString("name") + " costs: $" + inTable.getDouble("price") + "<Br>");
82                      //empty so print error message
83                      else
84                          out.println("There was no element with product ID: " + productSearch + " found in the table, please try again");
85
86                  }
87
88                  stmt.close();
89
```