This is a simple table that includes the functions that we used for the anvil web app.

Since anvil is a drag and drop type tool, this code will not be the most beneficial way to understand the functions we worked on in the Anvil web app.

|  |  |
| --- | --- |
|  |  |
| Home | class Home(HomeTemplate):    def \_\_init\_\_(self, \*\*properties):      # Set Form properties and Data Bindings.      self.init\_components(\*\*properties)      # Any code you write here will run when the form opens.      #self.image\_1.source = "https://2s7gjr373w3x22jf92z99mgm5w-wpengine.netdna-ssl.com/wp-content/uploads/2019/09/brain\_AI\_shutterstock\_Jozsef-Bagota.jpg"      def reset\_links(self, \*\*event\_args):      self.link\_1.role = ''      self.link\_2.role = ''      self.link\_3.role = ''      self.link\_4.role = ''      self.link\_5.role = ''      def button\_1\_click(self, \*\*event\_args):      """This method is called when the button is clicked"""      open\_form('Table', my\_parameter="an\_argument")    def link\_1\_click(self, \*\*event\_args):      """This method is called when the link is clicked"""      self.reset\_links()      self.link\_1.role = 'selected'      self.content\_panel.clear()      # Add Page1 to the content panel      self.content\_panel.add\_component(webScraper())    def link\_2\_click(self, \*\*event\_args):      """This method is called when the link is clicked"""      # Clear the content panel      self.reset\_links()      self.link\_2.role = 'selected'        self.content\_panel.clear()      # Add Page2 to the content panel      self.content\_panel.add\_component(Visualizations())      def link\_3\_click(self, \*\*event\_args):      self.reset\_links()      self.link\_3.role = 'selected'      """This method is called when the link is clicked"""       # Clear the content panel      self.content\_panel.clear()      self.link\_3.role = 'selected'      # Add Page2 to the content panel      self.content\_panel.add\_component(HomeForm())    def form\_show(self, \*\*event\_args):      """This method is called when the HTML panel is shown on the screen"""      pass    def link\_4\_click(self, \*\*event\_args):      """This method is called when the link is clicked"""      self.content\_panel.clear()      # Add Page2 to the content panel      self.content\_panel.add\_component(HeatMap())    def link\_5\_click(self, \*\*event\_args):      """This method is called when the link is clicked"""      self.content\_panel.clear()      # Add Page2 to the content panel      self.content\_panel.add\_component(Plots())    def stats\_click(self, \*\*event\_args):      """This method is called when the link is clicked"""      self.content\_panel.clear()      # Add Page2 to the content panel      self.content\_panel.add\_component(Stats()) |
| Quick Stats | class Stats(StatsTemplate):    def \_\_init\_\_(self, \*\*properties):      # Set Form properties and Data Bindings.      self.init\_components(\*\*properties)      bar\_graph\_data = app\_tables.clean\_products.search()      x = [x['descriptMain'] for x in bar\_graph\_data]      # Any code you write here will run when the form opens.      self.average\_ratingfunc()          def average\_ratingfunc(self):      bar\_graph\_data = app\_tables.clean\_products.search()      rating = [x['product\_rating'] for x in bar\_graph\_data]      avg\_rating = round(sum(rating) / len(rating), 2)      self.average\_rating.text = 'Average Rating: {}'.format(avg\_rating)        rating\_count = [x['rating\_count'] for x in bar\_graph\_data]      average\_rating\_count = round(sum(rating\_count) / len(rating\_count), 2)      #self.average\_rating\_count(text='Average Rating Count')      self.average\_rating\_count.text = 'Average Rating Count: {}'.format(average\_rating\_count)        answered = [x['answered'] for x in bar\_graph\_data]      avg\_answered = round(sum(answered) / len(answered), 2)      self.avg\_answered.text = 'Average Answered Questions: {}'.format(avg\_answered) |
|  |  |
|  |  |