```
virtualenv venv
source ./venv/bin/activate
```

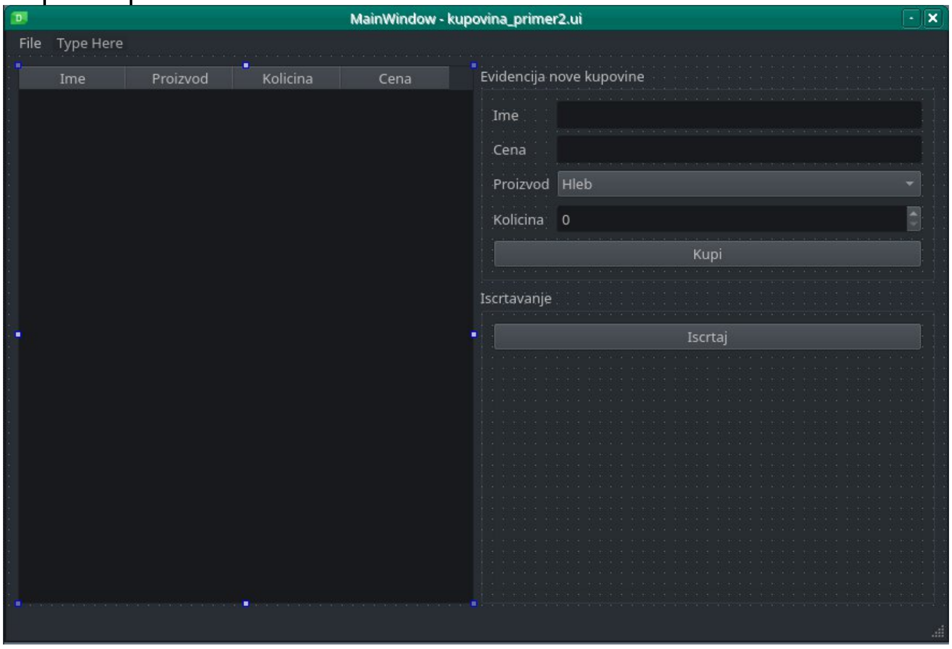
Signali i slotovi sa slajderom, tasterom i lcd-om.

```
import sys
from PySide6 import QtCore, QtGui, QtWidgets

class MyWidget(QtWidgets.QWidget):
   def __init__(self, parent=None):
      QtWidgets.QWidget.__init__(self, parent)

      self.resize(QtCore.QSize(800,600))

      quit = QtWidgets.QPushButton("Quit")
      #quit.setFont(QtGui.QFont("Times", 18, QtGui.QFont.Bold))

      self.lcd = QtWidgets.QLCDNumber(2)

      self.slider = QtWidgets.QSlider(QtCore.Qt.Horizontal)
      self.slider.setRange(0, 99)
      self.slider.setValue(0)

      self.slider.valueChanged.connect(self.data)
      quit.clicked.connect(self.quit)

      layout = QtWidgets.QVBoxLayout()
      layout.addWidget(quit)
      layout.addWidget(self.lcd)
      layout.addWidget(self.slider)
      self.setLayout(layout)

   def data(self,num):
      self.lcd.display(num)

   def quit(self):
      self.close()

app = QtWidgets.QApplication(sys.argv)
widget = MyWidget()
widget.show()
sys.exit(app.exec())
```

Stekovanje više elemenata odjednom

```
import sys
from PySide6 import QtCore, QtGui, QtWidgets

class LCDRange(QtWidgets.QWidget):
   valueChanged = QtCore.Signal(int)
   def __init__(self, parent=None):
      QtWidgets.QWidget.__init__(self, parent)

      self.lcd = QtWidgets.QLCDNumber(2)

      self.slider = QtWidgets.QSlider(QtCore.Qt.Horizontal)
      self.slider.setRange(0, 99)
      self.slider.setValue(0)

      self.slider.valueChanged.connect(self.slider_changed)

      #self.connect(self.slider, QtCore.SIGNAL("valueChanged(int)"),
      #            self, QtCore.SIGNAL("valueChanged(int)"))

      layout = QtWidgets.QVBoxLayout()
      layout.addWidget(self.lcd)
      layout.addWidget(self.slider)
      self.setLayout(layout)

   def value(self):
      return self.slider.value()

   def slider_changed(self,num):
      self.lcd.display(num)

   @QtCore.Slot(int)
   def setValue(self, value):
      self.slider.setValue(value)

class MyWidget(QtWidgets.QWidget):
   def __init__(self, parent=None):
      QtWidgets.QWidget.__init__(self, parent)

      quit = QtWidgets.QPushButton("Quit")
      #quit.setFont(QtGui.QFont("Times", 18, QtGui.QFont.Bold))
      quit.clicked.connect(self.quit)
      grid = QtWidgets.QGridLayout()
      previousRange = None

      layout = QtWidgets.QVBoxLayout()
      layout.addWidget(quit)
      layout.addLayout(grid)
      self.setLayout(layout)

      for row in range(3):
         for column in range(3):
            lcdRange = LCDRange()
            grid.addWidget(lcdRange, row, column)

            if previousRange:
               lcdRange.valueChanged.connect(previousRange.setValue)

            previousRange = lcdRange

   def quit(self):
      QtWidgets.QApplication.quit()


app = QtWidgets.QApplication(sys.argv)
widget = MyWidget()
widget.show()
sys.exit(app.exec())
```

U dizajneru napraviti prozor



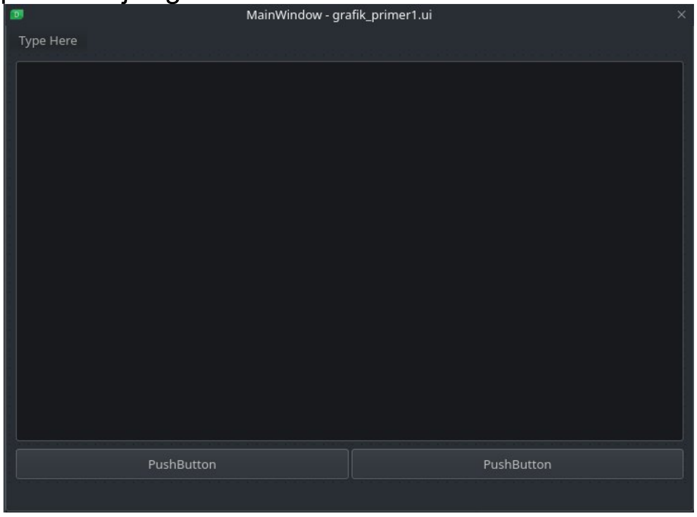
U padajući meni dodati File - Save, Close
Sa leve strane postaviti QTableWidget sa kolonama Ime, Proizvod, Kolicina, Cena
Sa desne strane gore postaviti groupBox i u njega dodati labelu Ime i lineEdit, zatim dodati labelu Cena i lineEdit2, labelu Proizvod i ComboBox sa elementima (Hleb, Vino, Mleko), labelu količina i spinBox, ispod svega dodati buy_button i preimenovati ga u Kupi. Celokupan GroupBox postaviti da bude u Form Layout-u.
Sa desne strane dole postaviti groupBox2 i u njega dodati draw_button sa natpisom Iscrtaj. Ispod njega dodati widget, zatim kliknuti desni klik - Promote to - u novom prozoru uneti Promoted class name: QChartView header file QtCharts.
Main window odabrati da bude u Grid layout-u.
Generisati pyuic-om gui.

```
from PySide6.QtWidgets import QApplication, QHeaderView, QMainWindow, QMessageBox, QTableWidgetItem
from PySide6.QtCharts import QPieSeries, QChart
from naziv generisanog fajla import Ui_MainWindow
import csv

class MainWindow(QMainWindow):
   def __init__(self):
      QMainWindow.__init__(self)
      self.ui = Ui_MainWindow()
      self.ui.setupUi(self)

      self.ui.tableWidget.horizontalHeader().setSectionResizeMode(QHeaderView.Stretch)

      self.ui.buy_button.clicked.connect(self.buy)
      self.items = 0

      self.ui.draw_button.clicked.connect(self.draw)

      self.ui.actionSave.triggered.connect(self.save)
      self.ui.actionClose.triggered.connect(self.close)

   def buy(self):
      name = self.ui.lineEdit.text()
```

```
        price = self.ui.lineEdit_2.text()
        product = self.ui.comboBox.currentText()
        quantity = str(self.ui.spinBox.value())

        self.ui.tableWidget.insertRow(self.items)
        self.ui.tableWidget.setItem(self.items, 0, QTableWidgetItem(name))
        self.ui.tableWidget.setItem(self.items, 1, QTableWidgetItem(product))
        self.ui.tableWidget.setItem(self.items, 2, QTableWidgetItem(price))
        self.ui.tableWidget.setItem(self.items, 3, QTableWidgetItem(quantity))
        self.items += 1

    def draw(self):
        products_dict = {
            "Hleb" : 0,
            "Vino" : 0,
            "Mleko" : 0,
        }

        for i in range(self.items):
            products_dict[self.ui.tableWidget.item(i, 1).text()] += int(self.ui.tableWidget.item(i,
3).text())

        series = QPieSeries()
        for key, value in products_dict.items():
            if value > 0:
                series.append(key, value)
        chart = QChart()
        chart.addSeries(series)
        #chart.legend().setAlignment(Qt.AlignLeft)
        self.ui.widget.setChart(chart)

    def save(self):
        rows = self.ui.tableWidget.rowCount()
        if rows:
            with open("data.csv", "w", newline="") as f:
                writer = csv.writer(f, delimiter=',')

                for i in range(rows):
                    name = self.ui.tableWidget.item(i, 0).text()
                    product = self.ui.tableWidget.item(i, 1).text()
                    price = self.ui.tableWidget.item(i, 2).text()
                    quantity = self.ui.tableWidget.item(i, 3).text()
                    writer.writerow([name,product,price,quantity])
                    QMessageBox.information(self, "Save","Saved file successfully")

    def close(self):
        QApplication.quit()

if __name__ == "__main__":
    app = QApplication()

    window = MainWindow()
    window.show()

    app.exec()
```

Korišćenje pyqtgraph-a

U dizajneru napraviti prozor koji izgleda kao:



Dodati dva tastera pushButton i pushButton1, dati im imena crtaj i brisi

U sredinu dodati Graphics View pod padajućim menijem Display Widgets. Nakon dodavanja desni klik -> Promoted Widget -> dobija se novi prozor. U polje Promoted class name upisati Plot Widget, a u polje header file pyqtgraph.

Izgenerisati pySide6-uic

```
import sys
from PySide6.QtWidgets import QApplication, QMainWindow
from (ime generisanog fajla) import Ui_MainWindow
from numpy.random import uniform

class MainWindow(QMainWindow):
    def __init__(self):
        super(MainWindow, self).__init__()
        self.ui = Ui_MainWindow()
        self.ui.setupUi(self)

        self.ui.pushButton.clicked.connect(self.draw)
        self.ui.pushButton_2.clicked.connect(self.clear)

    def draw(self):
        x = uniform(-1,1,1000)
        y = uniform(-1,1,1000)
        self.ui.graphicsView.plot(x,y, pen=None, symbolSize=1, symbolPen='g', symbol='x')
## setting pen=None disables line drawing

    def clear(self):
        self.ui.graphicsView.clear()

if __name__ == "__main__":
    app = QApplication(sys.argv)

    window = MainWindow()
    window.show()

    sys.exit(app.exec())
```