

Vežbe 8 - Zadaci

Instalirati virtuelno okruženje:

```
sudo apt install python3-venv
```

Napraviti virtuelno okruženje:

```
python3 -m venv ime_okruzenja
```

Aktivirati virtuelno okruženje:

```
source ime_okruzenja/bin/activate
```

Instalirati PySide6 komandom:

```
pip3 install PySide6
```

Otvoriti folder u kom je napravljeno virtuelno okruženje u VScode. Prilikom realizacije .py fajla, VScode bi trebao prepoznati interpreter u donjem desnom uglu (Python) a pored se nalazi verzija pajtona gde u zagradi pored broja verzije mora pisati (*'ime_okruzenja':venv*). Ako to nije slučaj, kliknuti na verziju i iz padajućeg menija odabrati interpreter virtuelnog okruženja. Ako to nije moguće aktivirati ga manuelno kao što je objašnjeno u koracima iznad.

Dizajner bi trebao biti dostupan sa instalacijom pyside6. Pokreće se iz terminala komandom:

```
pyside6-designer
```

Za prevođenje ui dizajna u py fajl koristiti komandu: *pyside6-uic ime_fajla.ui > ime_fajla.py*

Ako se pojavi greška:

```
qt.qpa.plugin: Could not load the Qt platform plugin 'xcb' in " even though it was found. This application failed to start because no Qt platform plugin could be initialized. Reinstalling the application may fix this problem.
```

```
Available platform plugins are: eglfs, linuxfb, minimal, minimalegl, offscreen, vnc, xcb. Aborted (core dumped)
```

Rešenje:

```
sudo apt install '^libxcb.*-dev'
```

Zadatak 1. Realizovati aplikaciju u PySide6 kao instancu klase Widget sa tri QCheckBox i labelom. Kada korisnik odabere odgovarajući QCheckBox u terminalu se ispisuje poruka.

```
1 import sys
2 from PySide6.QtWidgets import (QApplication, QWidget, QCheckBox, QLabel)
3
4 class MainWindow(QWidget):
5     def __init__(self):
6         super().__init__()
7         self.initializeUI()
8
9     def initializeUI(self):
10        self.setGeometry(100, 100, 250, 150)
11        self.setWindowTitle('QCheckBox Example')
12        self.setUpMainWindow()
13        self.show()
14
15    def setUpMainWindow(self):
16        header_label = QLabel('Koji termin ti odgovara? \
17        (Odaberi iz liste)', self)
```

```

18     header_label.resize(250, 50)
19     header_label.setWordWrap(True)
20     header_label.move(20, 10)
21
22     morning_cb = QCheckBox('Jutarnji [8 AM-2 PM]', self)
23     morning_cb.move(40, 60)
24     #morning_cb.toggle() # Ako se zeli da podrazumevano bude ukljucen
25     morning_cb.toggled.connect(self.printSelected)
26     after_cb = QCheckBox('Popodnevni [1 PM-8 PM]', self)
27     after_cb.move(40, 80)
28     after_cb.toggled.connect(self.printSelected)
29     night_cb = QCheckBox('Nocni [7 PM-3 AM]', self)
30     night_cb.move(40, 100)
31     night_cb.toggled.connect(self.printSelected)
32
33     def printSelected(self, checked):
34         sender = self.sender()
35         if checked:
36             print(f'{sender.text()} selektovan.')
37         else:
38             print(f'{sender.text()} deselektovan.')
39
40 if __name__ == '__main__':
41     app = QApplication(sys.argv)
42     window = MainWindow()
43     sys.exit(app.exec())

```

Zadatak 2. Realizovati aplikaciju koja predstavlja stranicu za prijavu na sistem. Ako korisnik nije registrovan omogućiti mu registraciju.

```

1 import sys
2 from PySide6.QtWidgets import (QApplication, QWidget, QLabel, QLineEdit,
3                               QPushButton, QCheckBox, QMessageBox)
4 from PySide6.QtGui import QFont, QPixmap
5 from registracija import NewUserDialog
6
7 class LoginWindow(QWidget):
8     def __init__(self):
9         super().__init__()
10        self.initializeUI()
11
12    def initializeUI(self):
13        self.setFixedSize(360, 220)
14        self.setWindowTitle('Prijava')
15        self.setUpWindow()
16        self.show()
17
18    def setUpWindow(self):

```

```

19     self.login_is_successful = False
20     login_label = QLabel('Prijava', self)
21     login_label.setFont(QFont('Arial', 20))
22     login_label.move(160, 10)
23
24     username_label = QLabel('Korisnicko ime:', self)
25     username_label.move(20, 54)
26     self.username_edit = QLineEdit(self)
27     self.username_edit.resize(210, 24)
28     self.username_edit.move(130, 50)
29     password_label = QLabel('Lozinka:', self)
30     password_label.move(20, 86)
31     self.password_edit = QLineEdit(self)
32     self.password_edit.setEchoMode(
33     QLineEdit.EchoMode.Password)
34     self.password_edit.resize(210, 24)
35     self.password_edit.move(130, 82)
36
37     self.show_password_cb = QCheckBox(
38     'Prikazi lozinku', self)
39     self.show_password_cb.move(130, 110)
40     self.show_password_cb.toggled.connect(
41     self.displayPasswordIfChecked)
42
43     login_button = QPushButton('Prijavi se', self)
44     login_button.resize(320, 34)
45     login_button.move(20, 140)
46     login_button.clicked.connect(self.clickLoginButton)
47
48     not_member_label = QLabel('Niste registrovani?', self)
49     not_member_label.move(20, 186)
50     sign_up_button = QPushButton('Registruj se', self)
51     sign_up_button.move(160, 180)
52     sign_up_button.clicked.connect(self.createNewUser)
53
54
55     def clickLoginButton(self):
56
57         users = {} # Rečnik za skladištenje informacija
58         file = 'putanja_do_fajla/users.txt'
59         try:
60             with open(file, 'r') as f:
61                 for line in f:
62                     user_info = line.split(' ')
63                     username_info = user_info[0]
64                     password_info = user_info[1].strip('\n')
65                     users[username_info] = password_info
66

```

```

67         username = self.username_edit.text()
68         password = self.password_edit.text()
69
70         if (username, password) in users.items():
71             QMessageBox.information(self, 'Uspesna prijava!',
72                                     'Uspesna prijava!', QMessageBox.StandardButton.Ok,
73                                     QMessageBox.StandardButton.Ok)
74             self.login_is_successful = True
75             self.close() # Close the login window
76             self.openApplicationWindow()
77
78         else:
79             QMessageBox.warning(self, 'Greska',
80                                 'Korisnicko ime ili lozinka nisu ispravni.',
81                                 QMessageBox.StandardButton.Close,
82                                 QMessageBox.StandardButton.Close)
83     except FileNotFoundError as error:
84         QMessageBox.warning(self, 'Greska',
85                             f'''<p>Fajl nije pronadjen.</p>
86                             <p>Razlog: {error}</p>''',
87                             QMessageBox.StandardButton.Ok)
88
89     def displayPasswordIfChecked(self, checked):
90         if checked:
91             self.password_edit.setEchoMode(QLineEdit.EchoMode.Normal)
92         elif checked == False:
93             self.password_edit.setEchoMode(QLineEdit.EchoMode.Password)
94
95     def createNewUser(self):
96         self.create_new_user_window = NewUserDialog()
97         self.create_new_user_window.show()
98
99     def openApplicationWindow(self):
100         self.main_window = MainWindow()
101         self.main_window.show()
102
103     def closeEvent(self, event):
104         if self.login_is_successful == True:
105             event.accept()
106         else:
107             answer = QMessageBox.question(
108                 self, 'Zatvaranje aplikacije',
109                 'Da li ste sigurni?',
110                 QMessageBox.StandardButton.No | \
111                 QMessageBox.StandardButton.Yes,
112                 QMessageBox.StandardButton.Yes)
113             if answer == QMessageBox.StandardButton.Yes:
114                 event.accept()

```

```

115         if answer == QMessageBox.StandardButton.No:
116             event.ignore()
117
118 class MainWindow(QWidget):
119     def __init__(self):
120         super().__init__()
121         self.initializeUI()
122
123     def initializeUI(self):
124         self.setFixedSize(640, 426)
125         self.setWindowTitle('Dobrodošli')
126         self.setUpMainWindow()
127     def setUpMainWindow(self):
128         image = 'putanja_do_fajla/background_picture.jpg'
129         try:
130             with open(image):
131                 main_label = QLabel(self)
132                 pixmap = QPixmap(image)
133                 main_label.setPixmap(pixmap)
134                 main_label.move(0, 0)
135         except FileNotFoundError as error:
136             print(f'Slika nije pronađena.\n Greska: {error}')
137
138 if __name__ == '__main__':
139     app = QApplication(sys.argv)
140     window = LoginWindow()
141     sys.exit(app.exec())

```

Fajl 'registracija.py'

```

1 from PySide6.QtWidgets import (QDialog, QLabel, QPushButton,
2                                 QLineEdit, QMessageBox)
3 from PySide6.QtGui import QFont, QPixmap
4
5 class NewUserDialog(QDialog):
6     def __init__(self):
7         super().__init__()
8         self.setModal(True)
9         self.initializeUI()
10
11     def initializeUI(self):
12         self.setFixedSize(360, 320)
13         self.setWindowTitle('Registracija')
14         self.setUpWindow()
15
16     def setUpWindow(self):
17         login_label = QLabel('Registracija korisnika', self)
18         login_label.setFont(QFont('Arial', 20))

```

```

19     login_label.move(50, 20)
20     # Create QLabel for image 250x200
21     user_image = 'putanja_do_fajla/user.jpg'
22     try:
23         with open(user_image):
24             user_label = QLabel(self)
25             user_label.setGeometry(150,60,80,70)
26             pixmap = QPixmap(user_image)
27             w = min(pixmap.width(), user_label.width())
28             h = min(pixmap.height(), user_label.height())
29             pixmap = pixmap.scaled(w, h)
30             user_label.setPixmap(pixmap)
31
32     except FileNotFoundError as error:
33         print(f'Image not found. Error: {error}')
34
35     name_label = QLabel('Korisnicko ime:', self)
36     name_label.move(20, 144)
37     self.name_edit = QLineEdit(self)
38     self.name_edit.resize(210, 24)
39     self.name_edit.move(130, 140)
40     full_name_label = QLabel('Ime i prezime:', self)
41     full_name_label.move(20, 174)
42     full_name_edit = QLineEdit(self)
43     full_name_edit.resize(210, 24)
44     full_name_edit.move(130, 170)
45
46     new_pswd_label = QLabel('Lozinka:', self)
47     new_pswd_label.move(20, 204)
48     self.new_pswd_edit = QLineEdit(self)
49     self.new_pswd_edit.setEchoMode(
50     QLineEdit.EchoMode.Password)
51     self.new_pswd_edit.resize(210, 24)
52     self.new_pswd_edit.move(130, 200)
53     confirm_label = QLabel('Potvrđi:', self)
54     confirm_label.move(20, 234)
55     self.confirm_edit = QLineEdit(self)
56     self.confirm_edit.setEchoMode(
57     QLineEdit.EchoMode.Password)
58     self.confirm_edit.resize(210, 24)
59     self.confirm_edit.move(130, 230)
60
61     sign_up_button = QPushButton('Registruj se', self)
62     sign_up_button.resize(320, 32)
63     sign_up_button.move(20, 270)
64     sign_up_button.clicked.connect(self.confirmSignUp)
65
66     def confirmSignUp(self):

```

```

67     name_text = self.name_edit.text()
68     pswd_text = self.new_pswd_edit.text()
69     confirm_text = self.confirm_edit.text()
70     if name_text == '' or pswd_text == '':
71         QMessageBox.warning(self, 'Greska',
72             'Morate uneti korisnicko ime i lozinku.',
73             QMessageBox.StandardButton.Close,
74             QMessageBox.StandardButton.Close)
75     elif pswd_text != confirm_text:
76         QMessageBox.warning(self, 'Greska',
77             'Lozinke se ne poklapaju.',
78             QMessageBox.StandardButton.Close,
79             QMessageBox.StandardButton.Close)
80     else:
81         with open('putanja_do_fajla/users.txt', 'a+') as f:
82             f.write('\n' + name_text + ' ')
83             f.write(pswd_text)
84         self.close()

```

Zadatak 3. Napraviti aplikaciju koja čita iz fajla autora. Korisnik unosi u QLineEdit ime autora, i nakon potvrde pretrage, vrši se pretraživanje fajla. Sve akcije su praćene odgovarajućim porukama.

```

1  import sys
2  from PySide6.QtWidgets import (QApplication, QWidget, QLabel,
3                                 QMessageBox, QLineEdit, QPushButton)
4  from PySide6.QtGui import QFont
5
6  class MainWindow(QWidget):
7      def __init__(self):
8          super().__init__()
9          self.initializeUI()
10
11     def initializeUI(self):
12         self.setGeometry(100, 100, 340, 140)
13         self.setWindowTitle('QMessageBox Example')
14         self.setUpMainWindow()
15         self.show()
16
17     def setUpMainWindow(self):
18         catalogue_label = QLabel('Katalog Autora', self)
19         catalogue_label.move(100, 10)
20         catalogue_label.setFont(QFont('Arial', 18))
21         search_label = QLabel('Pretraga po imenu autora', self)
22         search_label.move(20, 40)
23         author_label = QLabel('Ime:', self)
24         author_label.move(20, 74)
25         self.author_edit = QLineEdit(self)

```

```

26     self.author_edit.move(70, 70)
27     self.author_edit.resize(240, 24)
28     self.author_edit.setPlaceholderText('Unos imena u formatu: Ime Prezime')
29     search_button = QPushButton('Pretraži', self)
30     search_button.move(140, 100)
31     search_button.clicked.connect(self.searchAuthors)
32
33     def searchAuthors(self):
34         file = 'putanja_do_fajla/autori.txt'
35
36         try:
37             with open(file, 'r') as f:
38                 authors = [line.rstrip('\n') for line in f]
39                 # Provera da li postoji autor
40             if self.author_edit.text() in authors:
41                 QMessageBox.information(self, 'Autor nije pronadjen',
42                                         'Autor pronadjen u katalogu!',
43                                         QMessageBox.StandardButton.Ok)
44             else:
45                 answer = QMessageBox.question(self,
46                                               'Autor nije pronadjen',
47                                               '''<p>Autor nije pronadjen u katalogu.</p>
48                                               <p>Da li zelite da nastavite?</p>''',
49                                               QMessageBox.StandardButton.Yes | \
50                                               QMessageBox.StandardButton.No,
51                                               QMessageBox.StandardButton.No)
52
53                 if answer == QMessageBox.StandardButton.No:
54                     print('Isključujem aplikaciju.')
55                     self.close()
56         except FileNotFoundError as error:
57             QMessageBox.warning(self, 'Greska',
58                                 f'''<p>Fajl nije pronadjen.</p>
59                                 <p>Razlog: {error}</p>
60                                 Aplikacija ce se zatvoriti.''' ,
61                                 QMessageBox.StandardButton.Ok)
62             self.close()
63
64     if __name__ == '__main__':
65         app = QApplication(sys.argv)
66         window = MainWindow()
67         sys.exit(app.exec())

```