

```
#ifndef SPISAKPASAPORASI_H
#define SPISAKPASAPORASI_H

#include <QDialog>
#include "konekcija.h"

namespace Ui {
class SpisakPasaPoRasi;
}

class SpisakPasaPoRasi : public QDialog
{
    Q_OBJECT

public:
    explicit SpisakPasaPoRasi(QWidget *parent = 0);
    ~SpisakPasaPoRasi();

private slots:
    void on_pushButtonIzadji_clicked();

    void on_pushButtonPrikazi_clicked();

private:
    Ui::SpisakPasaPoRasi *ui;
};

#endif // SPISAKPASAPORASI_H
```

```

#include "spisakpasaporasi.h"
#include "ui_spisakpasaporasi.h"
#include <QMessageBox>

SpisakPasaPoRasi::SpisakPasaPoRasi(QWidget *parent) :
    QDialog(parent),
    ui(new Ui::SpisakPasaPoRasi)
{
    ui->setupUi(this);

    Konekcija baza;

    if(!baza.dbOpen())
    {
        QMessageBox::critical(this, "Kriticna greska!!!", "Ne mozemo da ostvarimo
ispravnu komunikaciju sa bazom podataka!!!");
    }
    else
    {
        QSqlQuery upit;

        upit.prepare("SELECT IzlozbaID FROM Izlozba ORDER BY IzlozbaID;");

        if(!upit.exec())
            QMessageBox::critical(this, "Kriticna greska!!!", "SQL naredba nije
izvrshena!!!");
        else
        {
            QSqlQueryModel *model;
            model = new QSqlQueryModel();
            model->setQuery(upit);
            ui->comboBoxIzlozba->setModel(model);
        }

        upit.prepare("SELECT NazivRase FROM Rasa ORDER BY NazivRase;");

        if(!upit.exec())
            QMessageBox::critical(this, "Kriticna greska!!!", "SQL naredba nije
izvrshena!!!");
        else
        {
            QSqlQueryModel *model;
            model = new QSqlQueryModel();
            model->setQuery(upit);
            ui->comboBoxRasa->setModel(model);
        }

        baza.dbClose();
    }
}

SpisakPasaPoRasi::~SpisakPasaPoRasi()
{
    delete ui;
}

void SpisakPasaPoRasi::on_pushButtonIzadji_clicked()
{
    this->close();
}

void SpisakPasaPoRasi::on_pushButtonPrikazi_clicked()
{
    Konekcija baza;

    if(!baza.dbOpen())
    {
        QMessageBox::critical(this, "Kriticna greska!!!", "Ne mozemo da ostvarimo
ispravnu komunikaciju sa bazom podataka!!!");
    }
}

```

```

    }
    else
    {
        QString izlozbaID = ui->comboBoxIzlozba->currentText();
        QString nazivRase = ui->comboBoxRasa->currentText();
        QSqlQuery upit;

        upit.prepare("SELECT COUNT (*) FROM Rezultat WHERE IzlozbaID = :Izlozba AND
Rezultat IS NOT NULL;");
        upit.bindValue(":Izlozba", izlozbaID);

        if(!upit.exec())
            QMessageBox::critical(this, "Kriticna greska!!!", "SQL naredba nije
izvrsena!!!");
        else
        {
            QString brojPrijavljenih;
            upit.next();
            brojPrijavljenih = upit.value(0).toString();
            ui->labelBrojPasaNaIzlozbi->setText(brojPrijavljenih);
        }

        upit.prepare("SELECT COUNT (*) FROM Rezultat INNER JOIN Pas ON Rezultat.PasID =
Pas.PasID INNER JOIN Rasa ON Pas.RasaID = Rasa.RasaID WHERE Rezultat.IzlozbaID = :Izlozba
AND Rasa.NazivRase = :NazivRase AND Rezultat.Rezultat IS NOT NULL;");
        upit.bindValue(":Izlozba", izlozbaID);
        upit.bindValue(":NazivRase", nazivRase);

        if(!upit.exec())
            QMessageBox::critical(this, "Kriticna greska!!!", "SQL naredba nije
izvrsena!!!");
        else
        {
            QString brojPasaTrazeneRase;
            upit.next();
            brojPasaTrazeneRase = upit.value(0).toString();
            ui->labelBrojPasaTrazeneRase->setText(brojPasaTrazeneRase);
        }

        upit.prepare("SELECT Pas.Ime AS [Ime psa] FROM Pas INNER JOIN Rezultat ON
Rezultat.PasID = Pas.PasID INNER JOIN Rasa ON Pas.RasaID = Rasa.RasaID WHERE
Rezultat.IzlozbaID = :Izlozba AND Rasa.NazivRase = :NazivRase AND Rezultat.Rezultat IS
NOT NULL GROUP BY Rezultat.KategorijaID;");
        upit.bindValue(":Izlozba", izlozbaID);
        upit.bindValue(":NazivRase", nazivRase);

        if(!upit.exec())
            QMessageBox::critical(this, "Kriticna greska!!!", "SQL naredba nije
izvrsena!!!");
        else
        {
            QSqlQueryModel *model = new QSqlQueryModel();
            model->setQuery(upit);
            ui->tableView->setModel(model);
        }

        baza.dbClose();
    }
}

```