პროექტი არის სტუდენტების მართვის სისტემის შესახებ, კოდი Odoo-ში არის დაწერილი, Odoo არის ღია კოდის ERP (Enterprise Resource Planning) სისტემა, რომელიც უზრუნველყოფს ბიზნეს აპლიკაციების შექმნის ჩარჩოს.

იმპორტი:

- კოდი ახდენს საჭირო მოდულების და ფუნქციების იმპორტს:

- base64: ბინარული მონაცემების კოდირებისთვის და დეკოდირებისთვის.

- BytesIO io-დან: ორობითი მონაცემების მეხსიერებაში დასამუშავებლად.

- სხვადასხვა მოდული და ფუნქციები Odoo-დან: `models`, `fields`, `api`, `registry`, `tools` და `exceptions`.

- `logging`: შეტყობინებების აღრიცხვისთვის.

- `Workbook` `xlsxwriter`-დან: Excel ფაილების შესაქმნელად.

Პირველი კლასი არის ‘StudentInformation’, რომელიც წარმოადგენს სტუდენტის ინფორმაციას. ის მოიცავს ველებს, როგორიცაა სრული სახელი, სახელი, გვარი, პირადობის მოწმობის ნომერი, სქესი, დაბადების თარიღი, სურათი, მშობლის სახელი, მობილური ტელეფონი, მისამართი, სიის ნომერი და შეფასება.

მეთოდი ‘compute\_full\_name’ არის გამოთვლილი ველი, რომელიც აერთიანებს სახელსა და გვარს სრული სახელის შესაქმნელად. management\_system\_views-ში button-ზე არის მინიჭებული ეს მეთოდი.

Მეორე კლასი ‘AddStudent’, ეს მოდელი მემკვიდრეობას იღებს res.partner`-ისგან, მოდელი `AddStudent` იძენს წვდომას ყველა ველსა და ფუნქციონალზე, რომელიც უკვე არის განსაზღვრული `res.partner` მოდელში. ეს მოიცავს ველებს, როგორიცაა სახელი, მისამართი, ტელეფონის ნომერი და ა.შ., ისევე როგორც ნებისმიერი მეთოდი ან ქცევა, რომელიც განსაზღვრულია `res.partner`-ში

**addstud()** ეს მეთოდი ქმნის სტუდენტის ახალ ჩანაწერს მოწოდებული ინფორმაციის საფუძველზე.

**generate\_report()**, ეს მეთოდი პასუხისმგებელია Excel-ის რეპორტის გენერირებაზე, რომელიც შეიცავს სტუდენტის ინფორმაციას. Დეტალურად რომ აღვწეროთ:

**output = BytesIO()**- შედეგი გამოაქვს ბაიტის სახით.

**book = Workbook(output)**- ქმნის ექსელის ფანჯარას, სადაც წერს აუთფუთში მოცემულ მონაცემს.

**sheet = book.add\_worksheet('Student Information')** - book-ს ამატებს ახალ worksheet-ს სათაურით "Student Information"

**sheet.set\_column('A:AZ', 15.0)**- აყენებს A-დან AZ-მდე სვეტების სიგანეს 18.0-მდე

**sheet.set\_row(0, 60)** - ეს ფუნქცია გამოიყენება მწკრივის სიმაღლის დასაყენებლად. პირველი პარამეტრი 0 განსაზღვრავს იმ მწკრივის ინდექსს, რომლის შეცვლაც გსურთ, მეორე პარამეტრი 60 განსაზღვრავს მწკრივის სიმაღლეს. ამ შემთხვევაში, ის დაყენებულია 60 პიქსელზე.

**sheet.freeze\_panes(1, 0)** - აფიქსირებს პირველ რიგს და მარცხენა სვეტს (სტრიქონი 1, სვეტი 0). ეს ნიშნავს, რომ სამუშაო ფურცლის სქროლვისას, დაფიქსირებული რიგები და სვეტები ხილული დარჩება.

**header\_style = book.add\_format(...)** - სათაურისთვის ქმნის ფორმატს ახალი თვისებებით

**FROM res\_partner**

**where id = {self.id}**

**"""**

Sql-ის სელექთით ამოაქვს ინფორმაცია res.partner-დან, where id -ით ვუთითებ კონკრეტულ ერთეულს

**self.env.cr.execute(query)** - უშვებს სელექთს და იღებს ყველა შედეგს

ის იმეორებს მოტანილ შედეგებს და წერს მონაცემთა თითოეულ რიგს Excel-ის სამუშაო ფურცელში `write\_row()` მეთოდის გამოყენებით.

**book.close()** - book close() მეთოდი გამოიყენება Excel ფაილის დასასრულებლად

**output.seek(0**) - BytesIO ობიექტის `seek()` მეთოდი გამოიყენება კურსორის გადასაადგილებლად. როდესაც გამოიძახება `seek(0)`, ის კურსორს გადააქვს დასაწყისში (offset 0). ეს ნიშნავს, რომ წაკითხვის ან ჩაწერის შემდგომი ოპერაციები თავიდან დაიწყება.

**excel\_file = base64.b64encode(output.getvalue())** - შიფრავს Excel ფაილის შიგთავსს, რომელიც ინახება `output` ცვლადში base64 ფორმატში. `output.getvalue()` ამოიღებს `output` ცვლადში შენახულ შინაარსს.

**`report()`** მეთოდი იწყება `generate\_report()` მეთოდის გამოძახებით. Რომელიც ინახება result ცვლადში, results = self.generate\_report().

**attachment = self.env['ir.attachment'].create({**

**'datas': results.get('excel\_file'),**

**'name': results.get('filename'),**

**'mimetype': 'application/vnd.ms-excel'**

**})**

generate\_report()`-დან მიღებული შედეგების გამოყენებით იქმნება ახალი ჩანაწერი `ir.attachment` ტიპის. ეს ჩანაწერი წარმოადგენს Excel-ის რეპორტს ათაჩმენთად ოდუში

მოკლედ, report() მეთოდი ორკესტრირებს Excel-ის ანგარიშის გენერირებას, ათაჩემნთის შექმნას Odoo-ში ამ რეპორტის შესანახად და ათაჩემნთის დაკავშირებას მიმდინარე ჩანაწერთან, რითაც მომხმარებლებს აძლევს წვდომას გენერირებულ რეპორტზე.

**‘SearchStudent’** ეს მოდელი გამოიყენება სტუდენტების მოსაძებნად მათი სიის ნომრის მიხედვით.

**student\_list = fields.Many2many('student.information')**  - ეს ველი არის Many2many ველი. ის ამყარებს ურთიერთობას `student.information` მოდელთან, რაც საშუალებას აძლევს მრავალ სტუდენტს დაუკავშირდეს ამ საძიებო ჩანაწერს.

**studentsearch()** ეს მეთოდი ეძებს სტუდენტებს მათი სიის ნომრის მიხედვით.