## horizontal line



Odoo Training

**─**

Muhammad Jaffar Raza / İlayda Kuran

E-tuğra EBG - IT Department

# Overview

We are currently working with **Odoo version 15** in our projects. Therefore, this training case is designed in alignment with the features and capabilities of Odoo 15. For reference and further learning, you may consult the official documentation available at: <https://www.odoo.com/documentation/15.0/>

**How to Setup Odoo 15 (Developer Environment)**

🛠️ **If you have not yet installed Odoo 15**, we recommend setting up your environment before starting. You can follow this detailed installation guide for Ubuntu 20.04 LTS: [https://www.cybrosys.com/blog/how-to-install-odoo-15-on-ubuntu-2004-lts-server](https://www.cybrosys.com/blog/how-to-install-odoo-15-on-ubuntu-2004-lts-server" \t "_new)

This training document guides you through building a medical appointment system for doctors and patients using the Odoo 15 framework. It is designed as a sample case for technical training. **If you're new to Odoo development,** we recommend familiarizing yourself with basic concepts such as models, views, actions, menus, and access rights before getting started.

# Stakeholders

1. Doctors
2. Patients

# Models (Classes)

* Doctor
* Patient
* Department
* Appointment
* Treatment

- Doctor

*Fields:*

* First Name (Char)
* Last Name (Char)
* Full Name (Char with compute)
* Date of Birth
* Age (Integer and readonly)
* Phone (Char)
* Email (Char)
* Department (Many2one)
* Shift start (Float with time widget)
* Shift end (Float with time widget)

*Note: Use constraint to not duplicate email. Auto calculate age when user enter date of birth using onchange decorator.*

- Patient

*Fields:*

* Patient ID (Char and auto generate)
* First Name (Char)
* Last Name (Char)
* Full Name (Char with compute)
* Date of Birth (Date)
* Age (Integer and readonly)
* Address Text)
* Phone (Char)
* Email (Char)
* National ID No (Char)

*Note: Use constraint to not duplicate National ID No. Also use Odoo’s ir.sequence to auto generate new code for Patient ID at the time of creation. Auto calculate age when user enter date of birth using onchange decorator*

- Department

*Fields:*

* Name (Char)
* Code (Char)

*Note: Use constraint to not duplicate Code and set code as rec name.*

- Appointment

*Fields:*

* Appointment Date time (Datetime)
* Code (Char)
* Doctor (Many2many)
* Patient (Many2one)
* Stage (Selection with option draft, in-progress, done, cancel)
* Treatment (One2many)

*Note: Use Odoo’s ir.sequence to auto generate new code for Patient ID at the time of creation and use tree attribute for Treatment one2many.*

- Treatment

*Fields:*

* Name (Char)
* Is Done (Boolean)

Create form views, tree views and menus for all above mentioned models and use Odoo tracking as well.

# Functionalities

## Appointment wizard:

Create a menu first. Then from that menu you will open a wizard that will ask for the patient’s name and doctors names. With the create appointment button on wizard you will create an appointment in draft stage for the patients with mentioned doctors. Note doctors could be multiple.

## States Change Button

Create following buttons in header of appointment form view.

* Set to In-Progress. This button will change the stage to In-Progress. (Should be only visible at draft stage)
* Set to Done. This button will change the stage to Done. (Should be only visible at In-Progress Stage)
* Set to Draft. This button will change the stage to Draft. (Should be only visible at Done or In-Progress Stage)
* Set to Cancel (Should be visible on all stages except Cancel)

## Attributes:

Make all fields readonly at Done and Cancel stages of appointment

## Validations:

Don’t let user delete appointment if it is in Done stage.

Odoo Training: Phase 2

- Configuration:

*Install Sales and Invoicing module and create products:*

* Doctor Fee (Product Type: Service)
* Medicines (Product Type: Consumable)
* Room/Ward Charges (Product Type: Service)
* Injections (Product Type: Consumable)
* Drips (Product Type: Consumable)

- Appointment:

*New Fields to be added:*

* Total Amount (Compute)
* Pending Amount (Compute)
* Sale Order Line (One2many)
* Sale Orders Smart Button
* Invoice Smart Button
* Payment Smart Button

*Note: Use Existing Odoo’s model “sale.order.line” for Sale Order Line (One2many) Use Odoo’s models “sale.order”, “accoune.move” and “account.payment” for respectives smart buttons.*

- Sale Order:

*New Fields to be added:*

* Appointments Smart Button

- Invoice:

*New Fields to be added:*

* Appointments Smart Button

- Payment:

*New Fields to be added:*

* Appointments Smart Button

# Functionalities (Phase 2)

## Create Sale Order:

Create a button in the header of the appointment. Upon pressing the button you will create a sale order using the sale order lines in the appointment and associate that Sale Order with the appointment such that it should be seen in the Sale Order Smart Button.

## Create Invoice from Sale Order:

When you confirm the sale order, a button will appear on the Sale Order to create an invoice. It is odoo’s default functionality. You will not change this. What you need to do is, when Invoice(account.move) will be created you just associate that invoice (account.move) with the appointment.

## Create Payment from Invoice:

Same association you need to do with payment.

## Create Security Groups:

Create below mentioned security groups with access rights

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| User | Doctor Model | Patient Model | Appointment Model | Treatment Model |
| Admin | All Access | All Access | All Access | All Access |
| Doctor | Read and Write access | Read Access | Read, Write and create access | Read, write and create access |
| Patient | No Access | Read Access | Read Access | Read Access |

**Project Submission**

After completing the project, please upload the entire Odoo module to your personal GitHub repository. Once uploaded, share the link to the repository with us as your final project submission. Make sure the repository includes a proper README file, commit history, and any necessary installation or configuration instructions.