# Odoo

*Prepared by Mr. Mohit Hadiya*

## Understanding the Scaffold

Define what a scaffold is in Odoo development.

Walk through the directory structure and files created by the scaffold.

Discuss the purpose of each file (e.g., __init__.py, __manifest__.py, etc.).

## Exploring the Files

Explore the __init__.py, __manifest__.py, and other key files generated.

Discuss the importance of the manifest file (__manifest__.py) and its role in module configuration.

Understand how to define dependencies, data files, and other module properties in the manifest.

## Understanding Fields and Their Uses

Introduce different field types available in Odoo (e.g., Char, Integer, Selection, Many2one, One2many, etc.).

Explain the purpose of each field and how they represent data relationships.

Demonstrate how to add fields to a model and define their properties.

Highlight the importance of proper field naming and data types.

## Creating a Model and View

Guide students in creating the school.student model using Python code.

Define essential fields such as 'name', 'age', 'gender', etc.

Explain the purpose of each field and how it contributes to the student information.

Demonstrate the creation of a form view,  list view and search view.

## Default Values in Fields

Discuss the concept of default values in Odoo fields.

Explain how default values can be set for fields during record creation.

Illustrate how to define default values in the model and view definitions.

Show examples of using default values to streamline data entry.

## Chatter

Define Chatter as a powerful communication tool within Odoo modules.

Explain how Chatter enables users to collaborate, discuss, and attach files to records.

Demonstrate how to add Chatter to models and use it effectively.

Show examples of using Chatter in real-world scenarios, such as discussing student assignments.

## Compute and Onchange Methods

Introduce the concept of Compute and Onchange methods in Odoo models.

Discuss the difference between the two and when to use each method.

Demonstrate how to write Compute methods for automatic field computation.

Explain how Onchange methods dynamically update fields based on user inputs.

## Defining Actions Using Buttons

Introduce the concept of button actions in Odoo.

Explain how to create custom buttons in the module's views.

Illustrate how to define actions for these buttons to perform specific operations.

Show examples of common use cases, such as triggering email notifications.

## Attrs and Domain for Field Conditions

Define the use of Attrs and Domain in Odoo field definitions.

Discuss how Attrs control the visibility and behavior of fields on the view.

Explain how Domain filters records based on specified conditions.

Guide students in writing Attrs and Domain expressions for various scenarios.

## Notebook in Sheet (Form View)

Introduce Notebooks as a way to organize data in multiple tabs within a form view.

Explain how to create and use Notebooks in a sheet (form view).

Demonstrate how to add fields and controls to different tabs in the Notebook.

Encourage students to organize their forms effectively using this feature.

## Smart Buttons

Introduce Smart Buttons and their significance in Odoo views.

Explain how to create Smart Buttons that execute actions when clicked.

Illustrate how to use Smart Buttons to display dynamic data and perform computations.

Guide students in writing code for Smart Buttons in their modules.

Show examples of practical use cases for Smart Buttons, such as order processing and status updates.

## Constraints

Define Constraints and their role in maintaining data integrity.

Discuss the importance of enforcing rules and restrictions on field values.

Demonstrate how to define Constraints on Odoo models to validate data entries(sql and python).

Explain how Constraints prevent the creation of invalid or inconsistent records.

Encourage students to think of real-world scenarios where Constraints are beneficial.

## Groups and Access Control

Introduce Groups as a means of managing user access permissions.

Explain how to create and assign users to specific groups based on their roles.

Illustrate how to apply group-based access control to models and views.

Guide students in implementing Groups to restrict access to sensitive data.

## Cron Jobs

Define Cron Jobs and their significance in automating repetitive tasks.

Explain how to schedule periodic actions using Cron Jobs.

Demonstrate the creation of Cron Jobs to perform tasks such as sending automated emails.

## Sequences

Introduce Sequences as a way to generate unique reference numbers automatically.

Discuss the use of Sequences in different models to maintain data integrity.

Guide students in creating Sequences for their modules, such as generating unique student IDs.

## Server Actions

Introduce Server Actions as a powerful tool to automate actions in response to specific events.

Explain how to create Server Actions for various scenarios, such as sending notifications or updating records.

Illustrate the use of Server Actions in conjunction with Triggers, such as On Create or On Update.

## Report Creation

Introduce the concept of reporting in Odoo and its importance for data analysis.

Demonstrate how to create basic reports using Odoo's reporting engine.

Guide students in generating reports for their school management module, such as student attendance reports.