

# Multiple-choice questions (MCQs)

## Odoo Basics

1. What is the primary programming language used in Odoo development?
  - a) Java
  - b) C++
  - c) Python
  - d) Ruby
  
2. Which database does Odoo use by default?
  - a) MySQL
  - b) PostgreSQL
  - c) SQLite
  - d) Oracle
  
3. What are the key components of an Odoo module?
  - a) Controllers, Models, Views
  - b) Templates, Routes, Handlers
  - c) Tables, Columns, Rows
  - d) Forms, Reports, Charts

## Development Skills

4. Which Odoo ORM method is used to create a new record?

- a) ``write()``
- b) ``search()``
- c) ``create()``
- d) ``unlink()``

5. What is the purpose of QWeb in Odoo?

- a) Database management
- b) Backend processing
- c) Front-end templating
- d) API integration

6. How do you define a many-to-one relationship in Odoo models?

- a) ``fields.One2many()``
- b) ``fields.Many2one()``
- c) ``fields.Many2many()``
- d) ``fields.One2one()``

## Module Development

7. How do you add a new menu item in Odoo?
- a) By modifying the ``views`` directory
  - b) By creating a new ``ir.ui.menu`` record
  - c) By editing the ``ir.model`` file
  - d) By updating the ``res.users`` model
8. What is the function of access control lists (ACLs) in Odoo?
- a) To manage database schema
  - b) To control user access to models and records
  - c) To define workflows
  - d) To handle system logging
9. How can you extend an existing model in Odoo?
- a) By creating a new model with the same name
  - b) By modifying the database directly
  - c) By inheriting the existing model
  - d) By duplicating the model file

# Database Management

10. What command is used to backup an Odoo database?

- a) ``odoo-backup``
- b) ``pg_dump``
- c) ``db_backup``
- d) ``backup_db``

11. How do you perform a database migration in Odoo?

- a) By using ``odoo-migrate``
- b) By exporting and importing CSV files
- c) By creating and applying new data models
- d) By updating the module and using ``-u`` option

12. What is the role of XML files in Odoo modules?

- a) To handle database connections
- b) To define views, menus, and actions
- c) To manage user sessions
- d) To store binary data

## Version Control

13. What is the purpose of a pull request in GitHub?

- a) To delete a branch
- b) To request code reviews and merges
- c) To clone a repository
- d) To commit changes locally

14. How do you resolve a merge conflict in Git?

- a) By creating a new branch
- b) By deleting the conflicting files
- c) By manually editing the conflicting files
- d) By running ``git resolve``

15. Which command is used to clone a repository in Git?

- a) ``git fetch``
- b) ``git init``
- c) ``git clone``
- d) ``git merge``

## Problem-Solving and Debugging

16. Which Odoo log level would you use to debug an issue?

- a) INFO
- b) DEBUG
- c) WARNING
- d) ERROR

17. What is the purpose of the `pdb` module in Python?

- a) To create databases
- b) To debug Python code
- c) To manage package dependencies
- d) To perform performance testing

18. How do you start Odoo in developer mode?

- a) By setting `dev\_mode=True` in the configuration file
- b) By using the `--dev all` command line option
- c) By modifying the database schema
- d) By installing the `dev\_tools` module

# Project Management and Collaboration

19. What are the key principles of Agile methodology?

- a) Fixed project timelines and detailed documentation
- b) Iterative development and customer collaboration
- c) Top-down management and rigid processes
- d) Extensive planning and risk management

20. How do you use GitHub issues to manage project tasks?

- a) By committing changes directly to the issue
- b) By tracking tasks, bugs, and enhancements
- c) By merging branches into the main branch
- d) By creating backup copies of the repository

# Coding Challenges

***Choose only one from the next 3 challenges***

## **Instructions for All Challenges**

- Documentation: Include comments in your code to explain the functionality.
- Testing: Ensure your module works as expected by testing the new features.
- Submission: Provide a brief report explaining your approach and any challenges faced.



# Event Manager

## Part 1: Create a Custom Module

Task: Develop a custom Odoo module named `event\_manager` to manage company events. The module should include a model `event.event` with the following fields:

- Event Name (Char)
- Event Date (Date)
- Location (Char)
- Description (Text)

### **Requirements:**

1. Create the module structure with necessary manifest and init files.
2. Define the `event.event` model with the specified fields.
3. Create form and tree views for the `event.event` model.
4. Add a menu item under a new main menu `Events` to access the `event.event` records.

## Part 2: Customize the Sales Module

Task: Use the `event\_manager` module to link events with sales orders. Add a field `event\_id` (Many2one) to the Sales Order model (`sale.order`) to select an event related to the sales order.

### **Requirements:**

1. Inherit the `sale.order` model and add the `event\_id` field.
2. Create an inherited view to include the `event\_id` field in the Sales Order form view.
3. Ensure the event is linked properly in the sales order.

**Timing: 1 hour**

# Employee Manager

## Part 1: Create a Custom Module

Task: Develop a custom Odoo module named `employee\_manager` to manage employee certifications. The module should include a model `employee.certification` with the following fields:

- Certification Name (Char)
- Certification Date (Date)
- Certification Authority (Char)
- Employee (Many2one to `hr.employee`)

### **Requirements:**

1. Create the module structure with necessary manifest and init files.
2. Define the `employee.certification` model with the specified fields.
3. Create form and tree views for the `employee.certification` model.
4. Add a menu item under the HR menu to access the `employee.certification` records.

## Part 2: Customize the HR Module

Task: Use the `employee\_manager` module to link certifications with employees. Add a smart button to the employee form view to display related certifications.

### **Requirements:**

1. Inherit the `hr.employee` model to add a computed field `certification\_count`.
2. Create an inherited view to add the smart button to the employee form view.
3. Ensure the button shows the count of certifications and links to the certification records.

**Timing: 1 hour**

# Product Manager

## Part 1: Create a Custom Module

Task: Develop a custom Odoo module named ``product_manager`` to manage product warranties. The module should include a model ``product.warranty`` with the following fields:

- Warranty Name (Char)
- Warranty Period (Integer)
- Warranty Description (Text)
- Product (Many2one to ``product.product``)

### Requirements:

1. Create the module structure with necessary manifest and init files.
2. Define the ``product.warranty`` model with the specified fields.
3. Create form and tree views for the ``product.warranty`` model.
4. Add a menu item under the Products menu to access the ``product.warranty`` records.

## Part 2: Customize the Products Module

Task: Use the ``product_manager`` module to link warranties with products. Add a field ``warranty_id`` (Many2one) to the Product model (``product.product``) to select a warranty for the product.

### Requirements:

1. Inherit the ``product.product`` model and add the ``warranty_id`` field.
2. Create an inherited view to include the ``warranty_id`` field in the Product form view.
3. Ensure the warranty is linked properly in the product form.

**Timing: 1 hour**