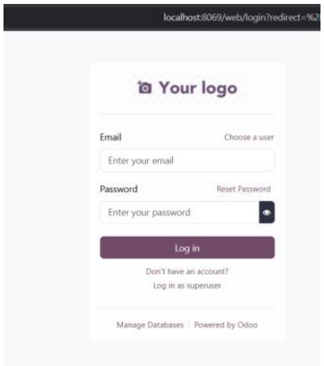
Task 04 – Library Management Module

This report covers the implementation of Task 04, which involves creating a custom Odoo module for Library Management. The objective was to build a functional module from scratch, understand Odoo's technical structure, and simulate a real-life development process.

1. Setup Steps

- Installed WSL and Ubuntu on Windows.
- Set up Odoo 18 CE by following the official documentation.
- Installed necessary dependencies: Python, PostgreSQL, wkhtmltopdf, etc.
- Created PostgreSQL database `mariam`.
- Launched Odoo on http://localhost:8069.
- Understood key folders: `addons/`, `custom-addons/`, `odoo-bin`, etc.

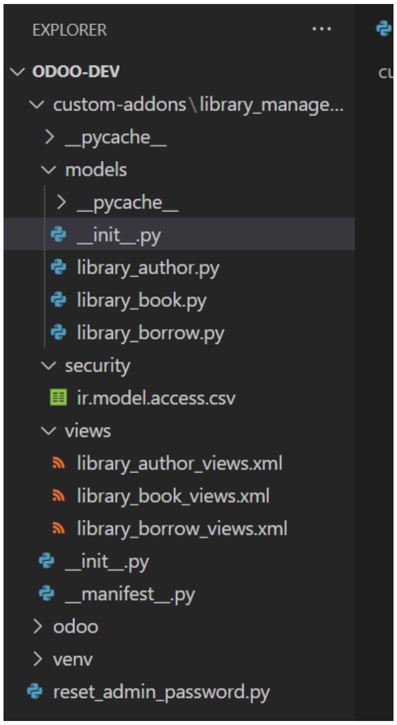


2. Module Overview

The Library Management module includes management of books, authors, and borrowing records. It features custom models, views, menus, and logic for book availability.

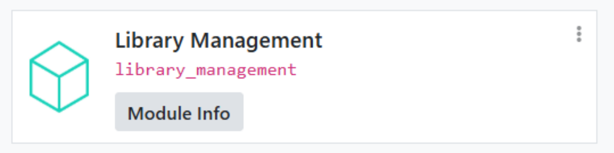
3. Module Structure

```
library_management/
├── __init__.py
├── __manifest__.py
├── models/
│   ├── __init__.py
│   ├── library_author.py
│   ├── library_book.py
│   └── library_borrow.py
├── views/
│   ├── library_author_views.xml
│   ├── library_book_views.xml
│   └── library_borrow_views.xml
├── security/
│   └── ir.model.access.csv
```



```
EXPLORER ...
ODOO-DEV
  custom-addons \ library_manage...
    __pycache__
    models
      __pycache__
      __init__.py
      library_author.py
      library_book.py
      library_borrow.py
    security
      ir.model.access.csv
    views
      library_author_views.xml
      library_book_views.xml
      library_borrow_views.xml
    __init__.py
    __manifest__.py
  odoo
  venv
  reset_admin_password.py
```

4. Features Implemented

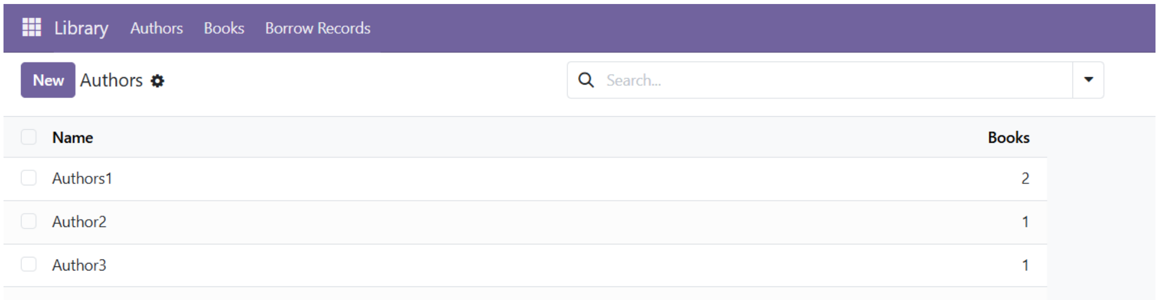


Library Management

`library_management`

Author Management

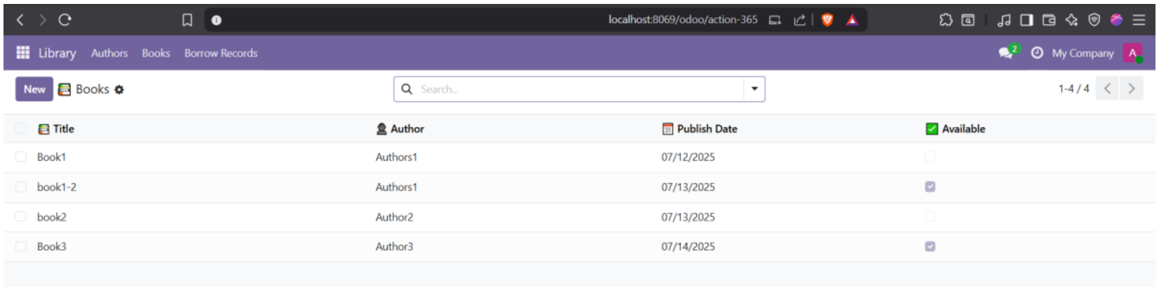
- Authors can be added and viewed.
- Each author is linked to multiple books.



□	Name	Books
□	Authors1	2
□	Author2	1
□	Author3	1

Book Management

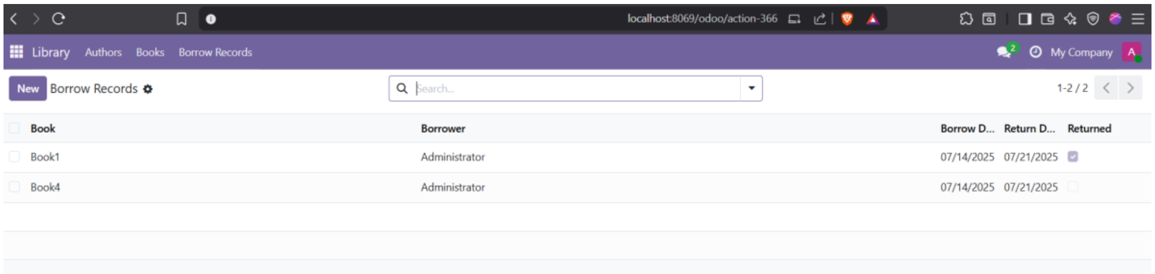
- Book model includes title, author, description, publish date, and availability.
- The 'Available' status updates based on borrow status.



□	Title	Author	Publish Date	Available
□	Book1	Authors1	07/12/2025	□
□	book1-2	Authors1	07/13/2025	☑
□	book2	Author2	07/13/2025	□
□	Book3	Author3	07/14/2025	☑

Borrowing Flow

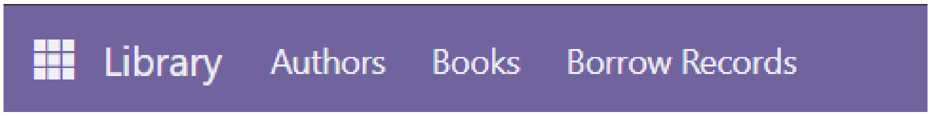
- Borrowing record includes book, borrower, borrow/return dates, and 'Returned' status.
- Auto-fill logic sets return date to 7 days after borrow date.
- Button available to mark books as returned.
- Borrowed books are marked unavailable automatically.



□	Book	Borrower	Borrow D...	Return D...	Returned
□	Book1	Administrator	07/14/2025	07/21/2025	☑
□	Book4	Administrator	07/14/2025	07/21/2025	□

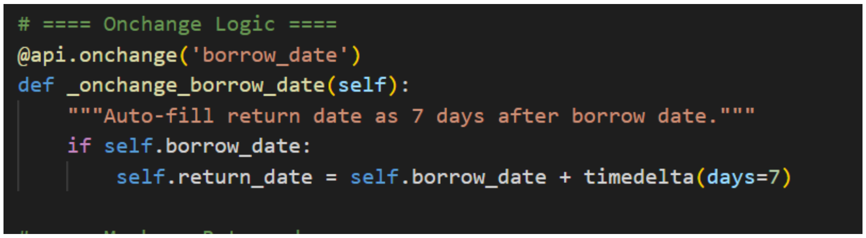
5. Navigation

- A top-level menu named 'Library' was added.
- Submenus: Books, Authors, Borrowing Records.



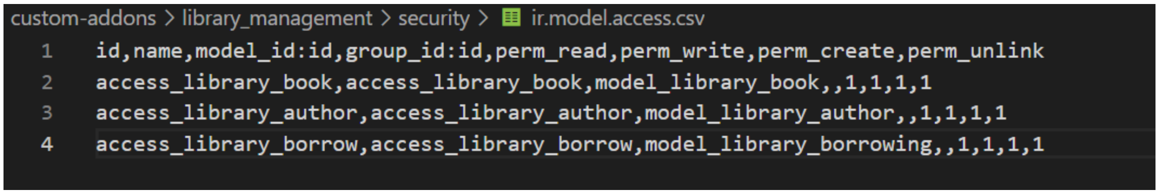
6. Implementation Details

- Used @api.onchange to auto-calculate return dates.



```python
# ==== Onchange Logic ====
@api.onchange('borrow_date')
def _onchange_borrow_date(self):
    """Auto-fill return date as 7 days after borrow date."""
    if self.borrow_date:
        self.return_date = self.borrow_date + timedelta(days=7)
```

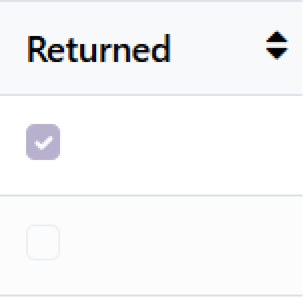
- Setup security access rights.



custom-addons > library_management > security > ir.model.access.csv

```
id,name,model_id:id,group_id:id,perm_read,perm_write,perm_create,perm_unlink
access_library_book,access_library_book,model_library_book,,1,1,1,1
access_library_author,access_library_author,model_library_author,,1,1,1,1
access_library_borrow,access_library_borrow,model_library_borrowing,,1,1,1,1
```

- Added filters for unreturned borrowings.



Returned
☑
☐

- Modular structure for clean separation.

7. Challenges & Solutions

- Manifest not detected: fixed by correcting file and format.
- Missing model error: solved with proper updates using -u flag.
- Book availability logic: added in both create() and write().

This task helped me explore custom module development in Odoo. I practiced setting up an environment, creating models, and building a smooth user flow for a real-world case.