

هدف این پروژه، طراحی و پیاده‌سازی یک وب اپلیکیشن مدیریت کتابخانه است که فرایندهای مربوط به ثبت کتاب ها، مدیریت اعضا، امانت و بازگشت کتاب ها را به صورت تحت وب انجام دهد.

این سیستم با نقش‌های اصلی مدیر (Admin) و عضو (Member) طراحی می‌شود تا هر کدام دسترسی متفاوتی داشته باشند. ۲. نقش‌ها و قابلیت‌های اصلی سیستم

#### نقش مدیر (Admin)

- افزودن، ویرایش و حذف کتاب ها
- مشاهده و مدیریت لیست امانت ها
- ثبت، تایید و ویرایش اطلاعات اعضا
- مشاهده گزارش ساده از وضعیت کتاب ها و امانت ها

#### نقش عضو (Member)

- جستجوی کتاب بر اساس عنوان یا نویسنده
- مشاهده وضعیت امانت‌های خود
- درخواست امانت کتاب
- بازگرداندن کتاب در موعد مشخص

#### چهار جدول اصلی:

users	نگهداری اطلاعات کاربران (مدیر و اعضا)
members	اطلاعات تکمیلی اعضای کتابخانه
books	مشخصات کتاب ها
borrowes	اطلاعات امانت و بازگشت کتاب‌ها

#### فناوری‌ها و ابزارهای مورد استفاده:

برای ساخت سرور و منطق سمت سرور	Node.js (Express.js)	Backend
برای ذخیره داده ها	MySQL	Database
برای ساخت رابط کاربری وب	HTML / CSS / JavaScript	Frontend
برای اجرای کدهای PHP	Apache (XAMPP)	Web Server
برای مدیریت نسخه‌های پروژه	Git + GitHub	Version Control

## ۱. جدول books (کتابها)

نام ستون	نوع داده	توضیح
id	INT PRIMARY KEY	ایدی کتاب
title	varchar	عنوان کتاب
author	varchar	نام نویسنده
publisher	varchar	نام ناشر
category	Varchar	دسته‌بندی (مثلاً رمان، تاریخی، علمی و...)
isbn	varchar	شماره شابک
year	date	سال چاپ
quantity	bool	تعداد موجودی فعلی

## ۲. جدول users (کاربران)

نام ستون	نوع داده	توضیح
id	INT PRIMARY KEY	ایدی
Name	varchar	نام کاربر
Email	varchar	ایمیل کاربر
Password	varchar	رمز عبور کاربر
role	('admin', 'member')	نقش کاربر در سیستم - یا مدیر یا عضو

## ۳. جدول members (اعضا)

برای اطلاعات تکمیلی عضوهایی که از نوع member هستند.

نام ستون	نوع داده	توضیح
id	INT PRIMARY KEY	ایدی برای هر عضو
User_id	INTINT (FK → users.id)	شناسه کاربر مربوطه
membership_no	varchar	شماره عضویت
expiry_date	DATE	تاریخ پایان عضویت

هر member دقیقاً به یک user مربوطه (رابطه ۱ به ۱).

## ۴. جدول borrow (امانت‌ها) ثبت امانت و بازگشت کتاب‌ها.

نام ستون	نوع داده	توضیح
borrow_id	INT PRIMARY KEY	شناسه یکتا برای هر امانت
member_id	Int (FK → members.id)	ایدی عضوی که کتاب رو امانت گرفته
book_id	Int (FK → books.id)	ایدی کتابی که امانت داده شده
borrow_date	data	تاریخ امانت کتاب
due_date	data	تاریخ بازگرداندن (مهلت)
return_date	date	تاریخ بازگشت واقعی (در صورت برگشت)
status	('borrowed','returned')	وضعیت امانت (فعال یا بازگشته)
fine_amount	double	مبلغ جریمه در صورت دیرکرد (اختیاری)

هر member می‌تونه چند borrow داشته باشه (۱ به چند)  
هر book می‌تونه در چند borrow ثبت بشه (۱ به چند)

### روابط بین جداول:

→ members ↔ users به یک

یعنی هر عضو فقط یک حساب کاربری داره،  
و هر کاربر (در صورتی که عضو باشه) فقط یک رکورد در members داره.

→ borrows ↔ members به چند

«یک عضو» ← «چند رکورد امانت در جدول borrows».

→ books ↔ borrows به یک

«یک کتاب» ← «چند بار امانت گرفته می‌شود».