# LIBRARY MANAGEMENT SYSTEM

DEVELOPED BY:
NURTORE D. MEIRZHAN SH.
BACKEND + FRONTEND WEB APPLICATION

# PROBLEM STATEMENT

Many small libraries:

- Use manual systems
- Lose book records
- Cannot track loans
- No automatic fine system

# SOLUTION

Our Solution
Library Management
System:

- Admin & Reader roles
- Book inventory tracking
- Loan management
- Fine calculation
- Modern web interface

# SYSTEM ARCHITECTURE

Frontend:
- HTML
- CSS
- JavaScript

Backend:
- Go (net/http)
- REST API

Database:
- JSON file (data.json)

# DATABASE STRUCTURE

**01**

Users
- id
- fullName
- email
- phone
- role

**02**

Books
- bookCode
- title
- author
- price
- stock

**03**

Loans
- reader
- book
- dueDate
- status
- fine

# ADMIN FEATURES

Each caAdmin can:

- Add new books
- Manage inventory
- View readers
- Approve loans
- Mark book as lost
- Calculate fine

mpaign has a funding goal and a deadline

# READER FEATURES

Reader can:

- Register account
- View books
- Borrow book
- See due date
- Return book

# LOAN LOGIC

Reader selects book
Stock decreases
Due date saved
If returned late → fine added
If lost → full price charged

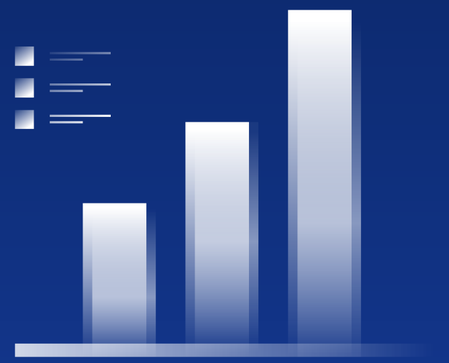
# TEST NETWORK USAGE

## Ethereum Test Network

- Network used: Sepolia
- Only free test ETH is used
- No real cryptocurrency involved
- Application is not deployed on mainnet

**Test ETH is obtained from an official faucet:**

https://cloud.google.com/application/web3/faucet/ethereum/sepolia



# SECURITY

🔐 Security Implementation

- Password hashing (bcrypt)
- Role-based access control
- Protected API routes
- Session validation

# DEMO FLOW

1. Register user
2. Login as admin
3. Add book
4. Borrow book
5. Return book
6. Check fine calculation

# TECHNOLOGIES

Technologies Used

- Go
- net/http
- bcrypt
- HTML5
- CSS3
- JavaScript
- JSON DB

# PROJECT CONCLUSION

✓ Fully functional library system
✓ Role-based architecture
✓ Real-life loan logic
✓ Modern dark UI
✓ Expandable to MongoDB