

IA Project Ideas

1- DevOps Boards

project called "DevOps Boards", a powerful project management tool that can help you and your team stay organized, collaborate more effectively, and deliver high-quality software on time and on budget (As JIRA).

- **Actors (2 types of users):**

- Team Leader
- Developer

- **Requirements:**

- Team leader can manage projects (create, list, update, delete). Add developer to specific project
- Team leader can manage cards/tasks (create, list, update, delete), besides filtering the cards by the unique identification id or by the assigned developer.
- The card will include: Title, description, status (to, in progress, done), and comment section.
- Developer can accept or reject joining specific project.
- Developer can update card's status. When status is done, developer should upload his work (attachment). Developer can see other developers' cards but can't update them.
- Team leader and developer can exchange comments.
- Team leader can view uploaded attachments

- **Users Roles:**

1. Team Leader:

- Login/ Logout.
- Manage Projects (CRUD) and add developers to project
- Manage Cards/tasks (CRUD) and assign developers

- Filter cards by (id) or assigned developer.
- View card status and open attachment if the card is done.
- Add a comment on specific card.

2. Normal User:

- Login/ Logout.
- Register.
- Accept/Reject joining a project
- View all assigned Cards and Update its status.
- List all cards assigned to other developers but can't update them.
- Upload attachment when update the card to done
- Add a comment/reply on a card.

- **Database Models:**

- DB design (Schema) is required

2- RideMe

Project called "RideMe" (Ride-Sharing System), is a user-friendly ride-sharing application that connects passengers and drivers, offering a convenient and efficient way to share rides for commuting, travel, or other transportation needs. (AS Uber)

Actors (3 types of users):

- Admin
- Passengers/Riders
- Drivers

Requirements:

- Admin can manage (accept or reject) new registered accounts (Passengers or Drivers).
- Admin can view rides' results (completed, not completed). Admin can block drivers with rate less than 2. As a result, this driver will not appear for any passenger.
- Each ride contain: passenger, driver, source, destination, status, trip price.
- Passenger can search and filter driver by car type, smoking/non-smoking rides, city, and region. Passenger can book a driver if he is not in a ride.
- Passenger can rate the driver from 1 to 5, provide feedback, show rides history. Passenger can pay the ride and end the ride.
- Driver can accept or decline received request. Driver can know the total income for specific day or in a specific month.

- **Users Roles:**

1. Admin:

- Login/ Logout
- Manage new accounts (accept or reject).
- View Rides' Results and feedbacks
- Block drivers with low rates

2. Passenger:

- Login/ Logout/ Register.
- Search for rides and view drivers profiles
- Book rides based on availability
- Rate the driver and provide feedback
- Pay for the ride
- Show rides' history

3. Driver:

- Login / Logout / Register.
- Accept/Decline received ride requests
- Get total income for all rides in specific day or month

- **Database Models:**

- DB design (Schema) is required

3- VaxScheduler

Project called "VaxScheduler" (Vaccination Scheduling Platform), is an online platform designed to streamline and simplify the process of scheduling and managing vaccination appointments. The platform aims to provide a user-friendly interface for individuals to schedule their vaccinations.

- **Actors (3 types of users):**

- Admin
- Patient
- Vaccination Center

- **Requirements:**

- Admin can add manage vaccination centers (Create, list, update, delete). Admin can manage vaccines (create, list, update, delete) for every vaccination center.
- Accept or reject registered users.
- Patient can list all vaccination centers and its vaccines.
- Vaccine includes: name, precautions, vaccination center, time gap between first and second dose.
- Patient can reserve vaccine (first and second dose). Patient can't reserve second dose if the first dose is not accepted yet by the vaccination center. Also, after accepting first dose, you need to choose a date after the gap period described in the vaccine.
- Patient can view certificate after accepting second dose from vaccination center.
- Vaccination center and list his registered users (who registered vaccines from this center).
- Vaccination center needs to approve the first and second reservation. Vaccination center should upload certificate after accepting second dose.

- **Users Roles:**

1. Admin:

- Login/ Logout
- Manage vaccination centers (CRUD)
- Manage vaccines (CRUD)
- Accept/Reject registered Users

2. Patient:

- Login/ Logout/ Register

- List vaccination centers and its vaccines
- Reserve vaccination (first and second dose)
- Prevent reserving second dose before accepting first dose from center
- View Certificate after second dose

3. Vaccination Center:

- Login/ Logout
- View patients that reserved vaccine from that vaccination center
- Approve (first and second dose)
- Upload patient certificate (after second dose)
- Prevent patient from reserving taken vaccine again

- **Database Models:**

- DB design (Schema) is required

4- JobConnect

Project called "JobConnect" (Job Portal Platform), is a comprehensive online platform designed to connect job seekers with employers, providing a convenient and efficient way for individuals to search for job opportunities and for employers to find suitable candidates for their job openings.

- **Actors (3 types of users):**

- Admin
- Job Seekers
- Employers/Recruiters

- **Requirements:**

- Admin can manage employers (Create, list, update, delete). Admin can accept or reject jobs written by the employer. If the job post is accepted, the job can appear for any job seeker.
- Job seeker can view all posted jobs. Job seekers can search for job based on criteria such as location, industry, job title, salary range, and Date.
- Job includes: employer name, Job Type (part-time, full-time, remote), Job Budget, post creation date, Job description, Number of proposal submitted
- Job seeker can apply for a job. The proposal can contain an attachments such as CV. Also Job seeker can save important jobs.
- Employer can post new jobs. Employer can review proposals and accept or reject them. Job is removed if a proposal is accepted.
- Employer and job seeker can communicate if the proposal is accepted.

- **Users Roles:**

1. Admin

- Login/ Logout
- Manage employers (CRUD)
- Accept or refuse job posts

2. Job Seekers:

- Login/ Logout/ Register.

- View all jobs and Search for job
- Apply for a job (sending proposal)
- Save important posts (to submit later)
- Communicate with employers (after acceptance)

3. Employers:

- Login/ Logout.
- Post new jobs
- Review submitted proposals (accept or reject)
- Communicate with job seeker (after acceptance)

- **Database Models:**

- DB design (Schema) is required

5- Online Libarary System

- **Actors (2 types of users):**

- librarian
- Normal user

- **Requirements:**

- librarian can manage books (create, list, update, delete), besides filtering the books by the unique identification code (ISBN) and the rack number for each book.
- Librarian can manage the user requests of borrowed books, and limit number of books that can be borrowed by the user, and the date of return.
- Normal user cannot log in until the librarian approve their account.
- User can search for the books available in the system.
- The user can borrow books and show the history of borrowed books.

1. librarian User:

- Login/ Logout
- Manage books (CRUD)
- Filter books by (ISBN) and rack number.
- Manage users accounts (approve or reject)
- Manage borrowed requests.
- Can generate reports about borrowed and available books

2. Normal User:

- Login/ Logout.
- Register.
- Search for books by all the available attributes

- Send request to borrow a book to the admin.
- Show a list of borrowed books.
 - **Database Models:**
- DB design (Schema) is required

6- Bus Booking Dashboard

- **Actors (2 types of users):**
 - Admin
 - Traveler

Requirements:

- Admin and Traveler must first log into the “Bus Booking” dashboard.
- Admin can manage the bus appointments (add, update, delete, and list).
- Each appointment has a specific number of travelers.
- Admin can manage travelers (add, update, delete, and list).
- Travelers can request an appointment.
- Travelers can show the requested appointments.

1. Admin User:

- Login/ Logout
- Manage appointments (CRUD)
- Manage bus destinations (CRUD)
- Manage Traveler (CRUD)
- Accept or decline requests by the Traveler (CRUD)
- Show the history of traveler requests in all appointments

2. Traveler User:

- Login/ Logout.
- Can view and filter appointments.
- Send a request to the admin with the required appointment.
- Show a history of appointment searches related to his account only.
 - **Database Models:**
- DB design (Schema) is required

7- Online Pharmacy Dashboard

- **Actors (2 types of users):**
 - Admin
 - Patient

Requirements:

- Admin and Patient must first log into the “online pharmacy” dashboard.
- Admin can manage the medicines (add, update, delete, and list).
- Admin can manage categories of medicines (add, update, delete, and list).
- Admin can manage patients (add, update, delete, and list).
- Patient can request medicines.
- Patient can show the requested medicines.

1. Admin User:

- Login/ Logout
- Manage medicines (CRUD)
- Manage categories of medicines (CRUD)
- Manage Patients (CRUD).
- Accept or decline requests by the Patient (CRUD).
- Show the history of patient requests for medicines.

2. Patient User:

- Login/ Logout.
- Can view and filter medicines.
- Send a request to the admin with the required medicines.
- Show a history of medicine searches related to his account only.
 - **Database Models:**
- DB design (Schema) is required

8- Logistics Dashboard

- **Actors (2 types of users):**
 - Admin
 - Supervisor
- **Requirements:**
 - To manage their inventory, admin or supervisors must first log into the Logistics dashboard.
 - Admin can manage warehouses (add, update, delete and list).
 - Each warehouse has its own set of products, which the admin control.
 - Admin can manage supervisors (add, update, delete and list).
 - Supervisor should be assigned to only one warehouse.
 - Supervisor can request to increase or decrease the stock quantity of the product per warehouse, and this request should be approved by the admin.

1. Admin User:

- Login/ Logout
- Manage warehouses (CRUD)
- Manage products per warehouses (CRUD)
- Manage supervisor (CRUD)
- Accept or declined stock requests by supervisor (CRUD)

- Show history of stock requests for all supervisors

2. Supervisor User:

- Login/ Logout.
- See products related to the warehouse assigned to him.
- Send request to the admin with the increase or decrease stock quantity of a specific product.
- Show a history of stock requests related to his account only.
 - **Database Models:**
- DB design (Schema) is required