1. Smart To-Do List with Task Prioritization

Features:

- User registration/login
- Create/edit/delete tasks
- AI-based task priority suggestion (keywords, deadlines)
- Dark/light theme toggle
- Functional/unit testing for task actions

DevOps:

- Dockerize frontend and backend services
- Use Docker Compose for service orchestration
- Jenkins pipeline for CI/CD deployment (lint → test → build → deploy)

QA:

- Unit testing for task CRUD operations
- Functional testing for login and theme toggle
- Load testing using JMeter for 1000 concurrent users

Use Case:

- A busy college student uses the app to prioritize their assignments and deadlines based on urgency and importance.

2. College Event Management System

Features:

- Event creation, registration
- Admin panel for event control
- RSVP and QR check-in
- AI prediction: which users are likely to attend based on history
- Bug reporting module

DevOps:

- Dockerized app with admin and user services
- Jenkins pipeline for automated testing and deployments
- Environment variable setup for DB and email services

QA:

- Scenario testing for registration and RSVP workflows
- Manual and automated testing for QR check-in flow
- Load testing for registration form handling (simulate event rush)

Use Case:

- A college admin can create an event, students RSVP and check in using QR codes at the entrance. AI helps predict which events need more promotion.

3. Book Recommendation Web App

Features:

- Browse books, give ratings
- User profile with reading history
- AI recommendation system (based on genre, author, behavior)
- Backend APIs with filtering options

DevOps:

- Backend and frontend Docker containers
- Jenkins auto-pipeline for code push to deployment server
- Version control for APIs using GitHub and semantic versioning

QA:

- Test cases for book search, filters, and recommendation accuracy
- Regression testing on user history updates
- Load testing: stress test book browsing endpoint

Use Case:

- A reader gets book suggestions after rating a few, and can search specific genres or authors easily.

4. Online Food Ordering Platform

Features:

- Filterable restaurant menu
- Cart, checkout flow with mock payment
- Backend for orders, status tracking
- AI model for personalized meal suggestions

DevOps:

- Full stack Dockerization with MongoDB/PostgreSQL
- Jenkins multi-stage pipeline: test → build → deploy
- Deployment to cloud using Docker Compose on VPS or AWS EC2

QA:

- UI testing for add-to-cart, payment pages
- Load testing for peak lunch-hour traffic
- Functional testing for order updates

Use Case:

- A user selects dishes based on personal taste and AI suggestion, adds to cart, checks out — restaurant receives the order instantly.

5. Student Feedback Sentiment Analysis

Features:

- Feedback submission for courses/instructors
- Admin dashboard for viewing reports
- AI model to detect sentiment (Positive, Negative, Neutral)
- API endpoints to store and analyze feedback

DevOps:

- Containerized ML model + backend
- Jenkins setup to retrain and redeploy model automatically
- Use GitHub Actions for testing

QA:

- Form validation testing
- Model output validation (cross-check with labeled data)
- Load testing for submission API (simulate exam-season feedback)

Use Case:

- University receives anonymous feedback and uses AI to analyze trends — e.g., which teacher received negative feedback and why.

6. Personal Finance Tracker

Features:

- Log income, expenses
- Monthly category-wise spending charts
- Budget alerts (email/SMS)
- AI-based insights (e.g., "You overspend on food")

DevOps:

- Docker-compose setup for DB, backend, and visualization service
- Jenkins cron jobs for daily expense report generation
- Secure secret management for budget alert services

QA:

- Testing on transaction validation (e.g., no negative expenses)
- Load test for monthly report generation
- UI testing for spending charts rendering

Use Case:

- A user gets a warning when they exceed their entertainment budget for the month.

7. Simple Chat Application with Emotion Detection

Features:

- Real-time chat using WebSockets
- User authentication and chat history
- Emotion detection from messages (happy, sad, angry)
- Emoji reactions and timestamped messages

DevOps:

- Dockerized WebSocket server and ML emotion detector
- Jenkins pipeline with WebSocket test coverage
- Use environment-based deployment for staging vs prod

QA:

- WebSocket connection test cases
- Sentiment accuracy validation using test dataset
- Load test: simulate 100 active users chatting concurrently

Use Case:

- Students chatting in a study group can see emoji reactions based on emotion detected in each message.