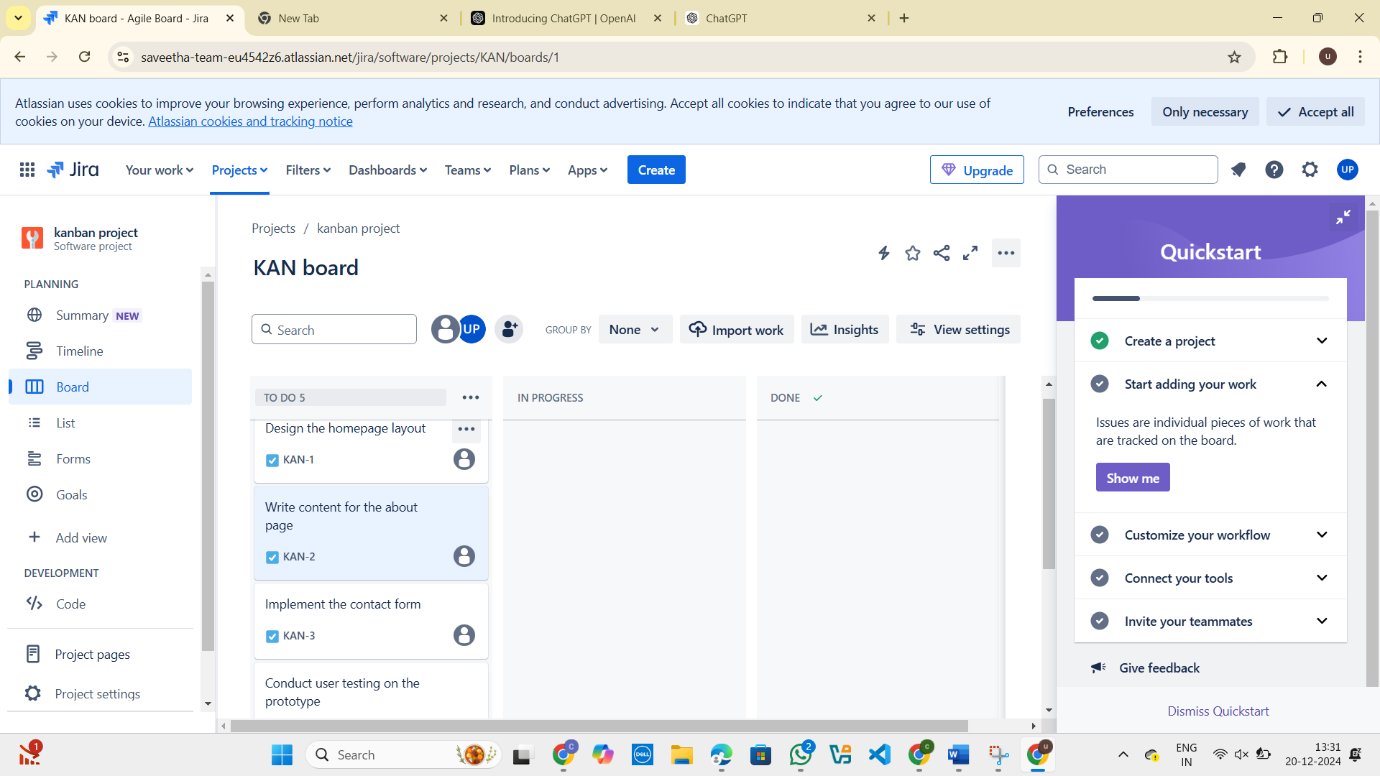
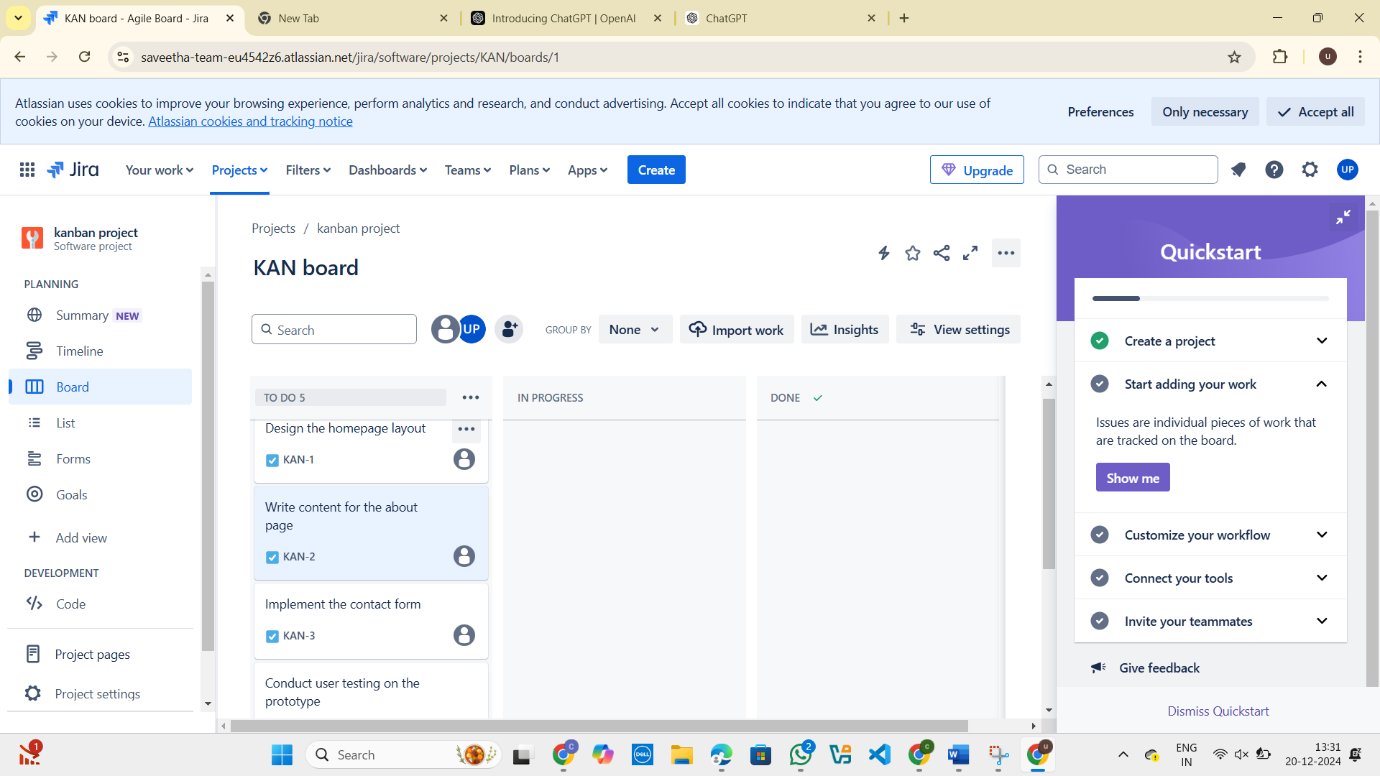
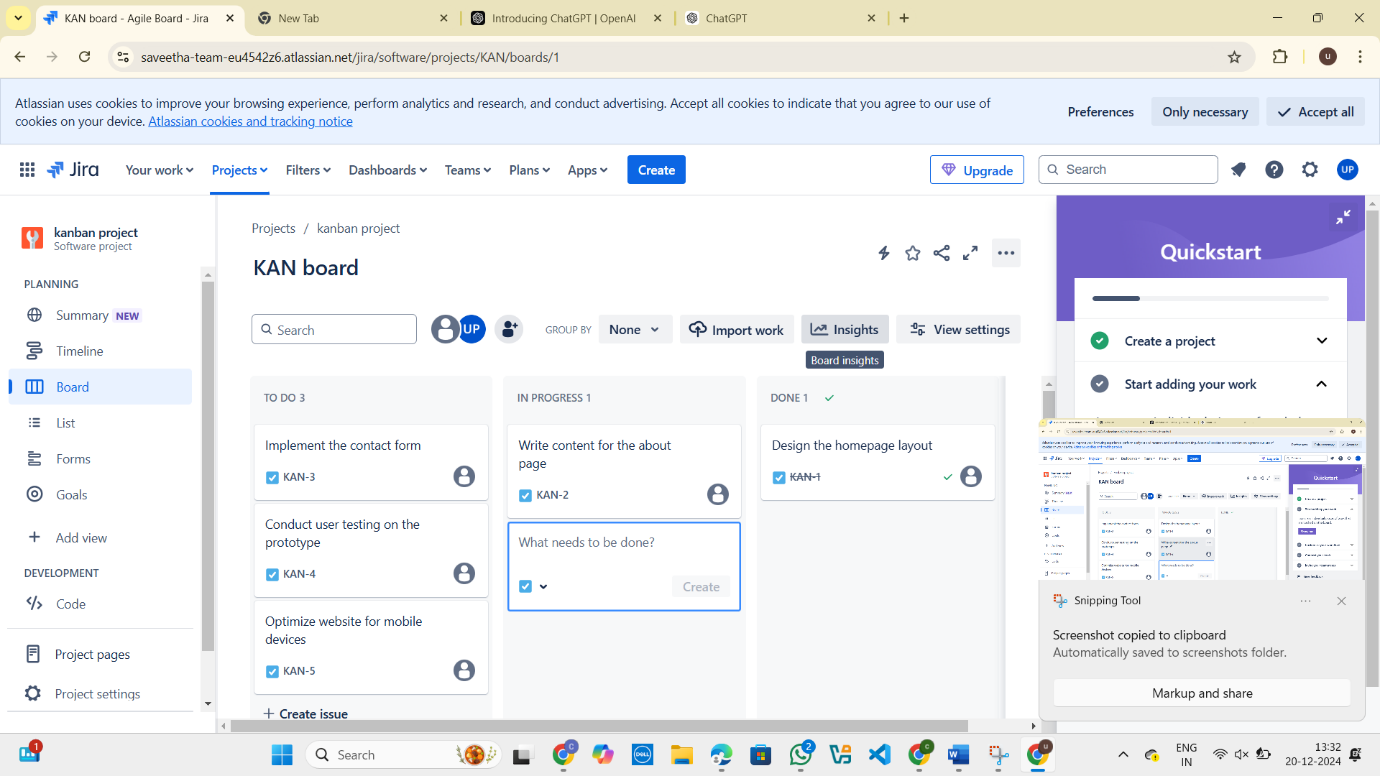
EXP 1:

KANBAN PROJECT:





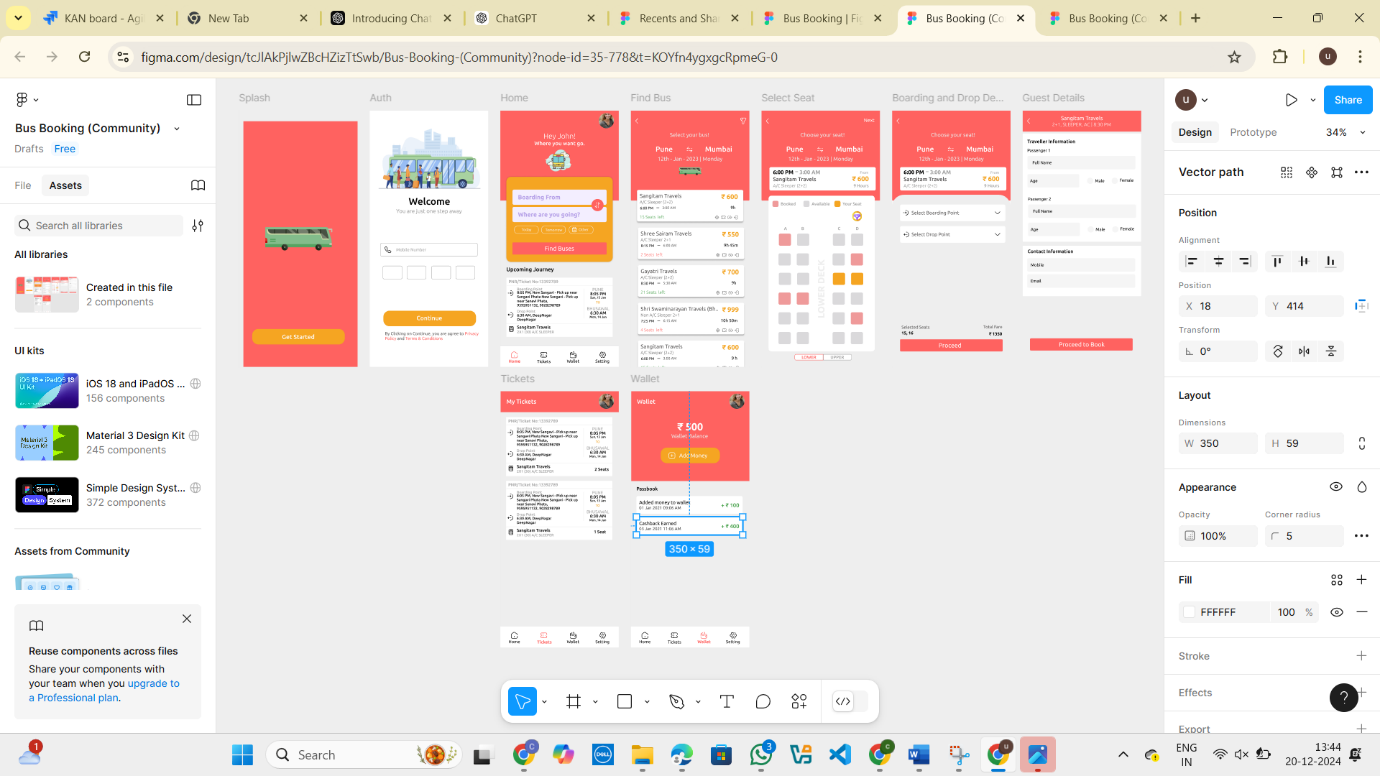


EXP 2:

FIGMA:

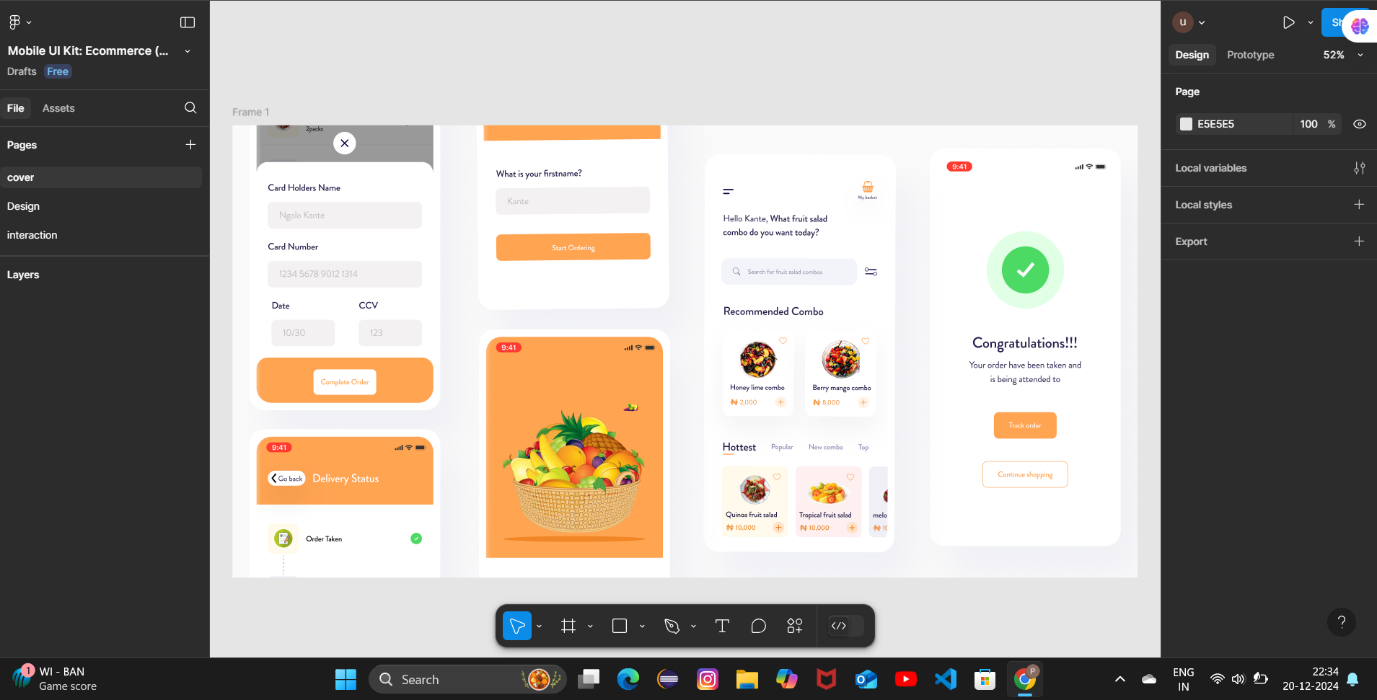
BUS TICKET BOOKING SYSTEM:





EXP 3:

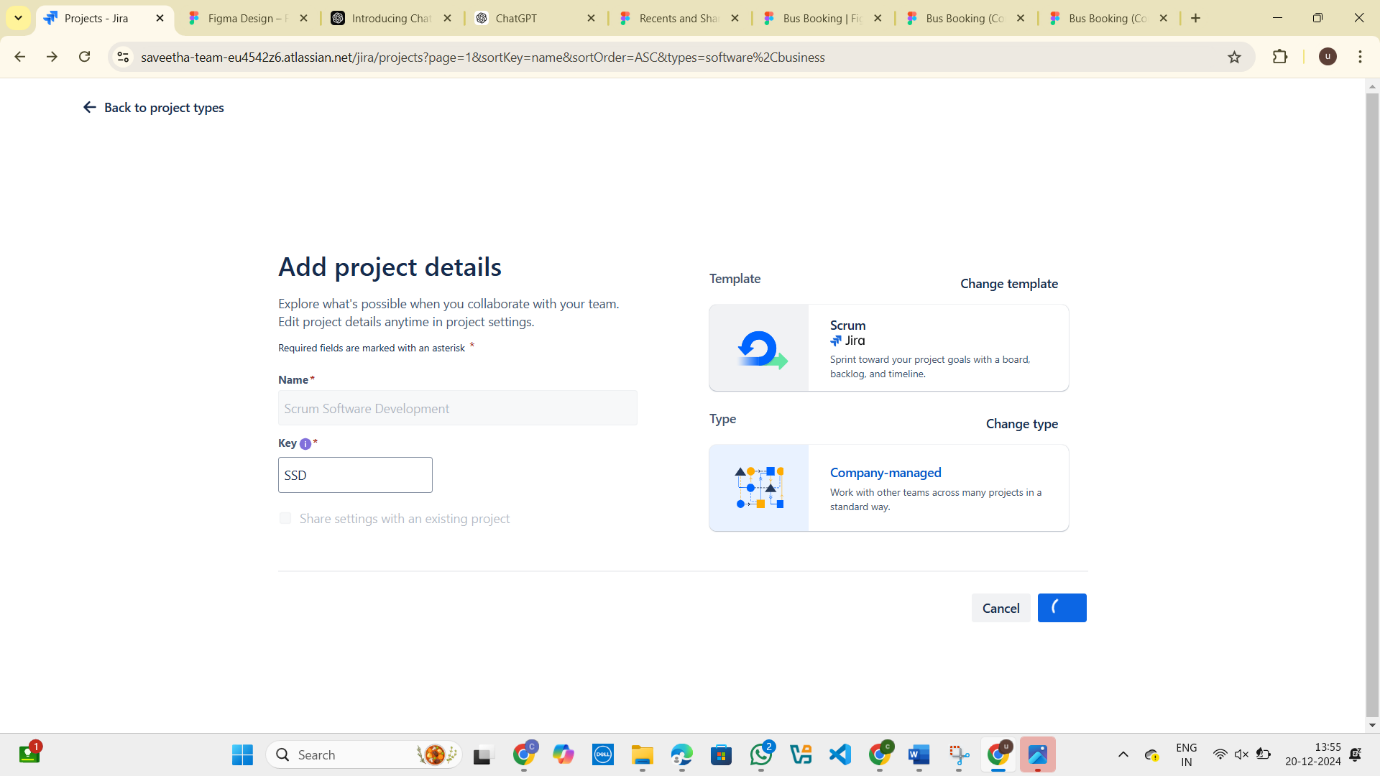
The stakeholders have conflicting views on the user interface design for an E-Commerce mobile app. Create a prototype using Figma tool to discuss with the stakeholders to get their feedback and approval.

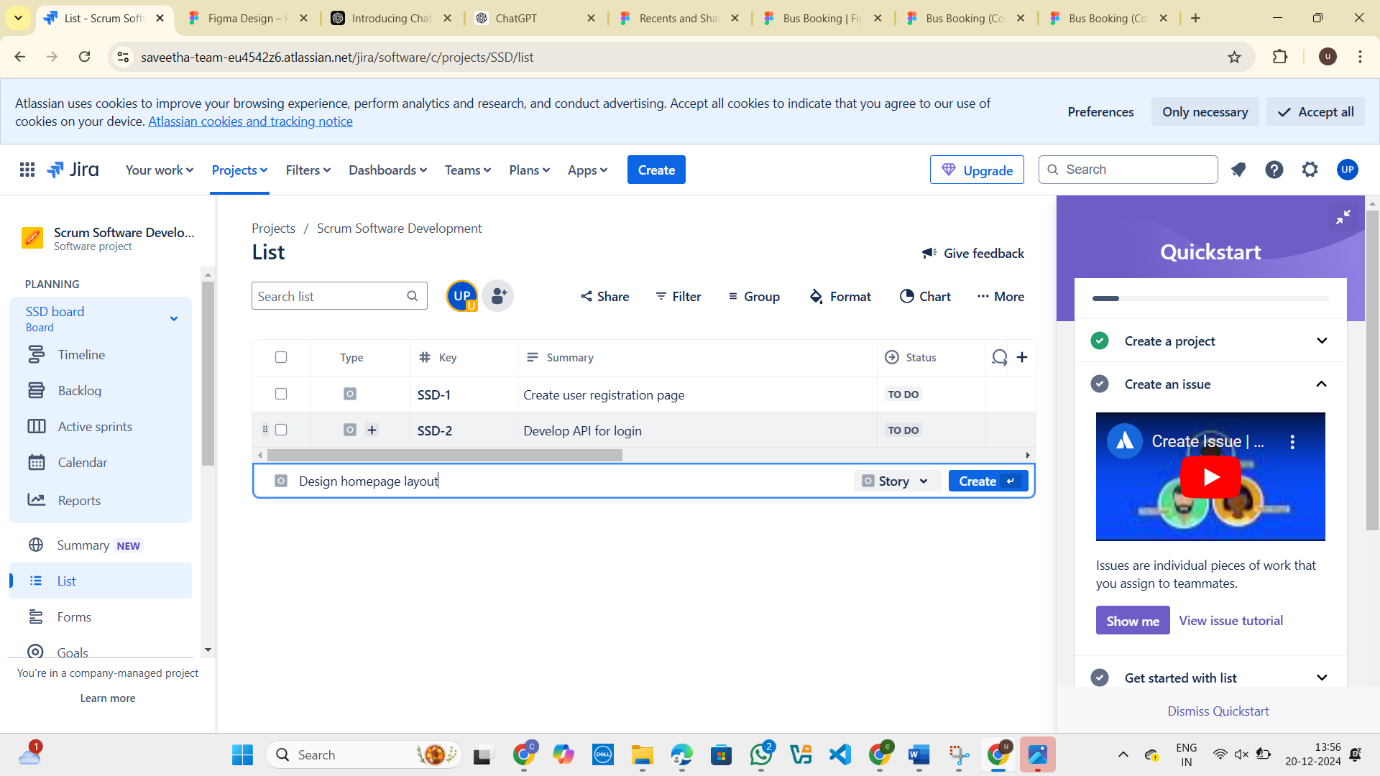


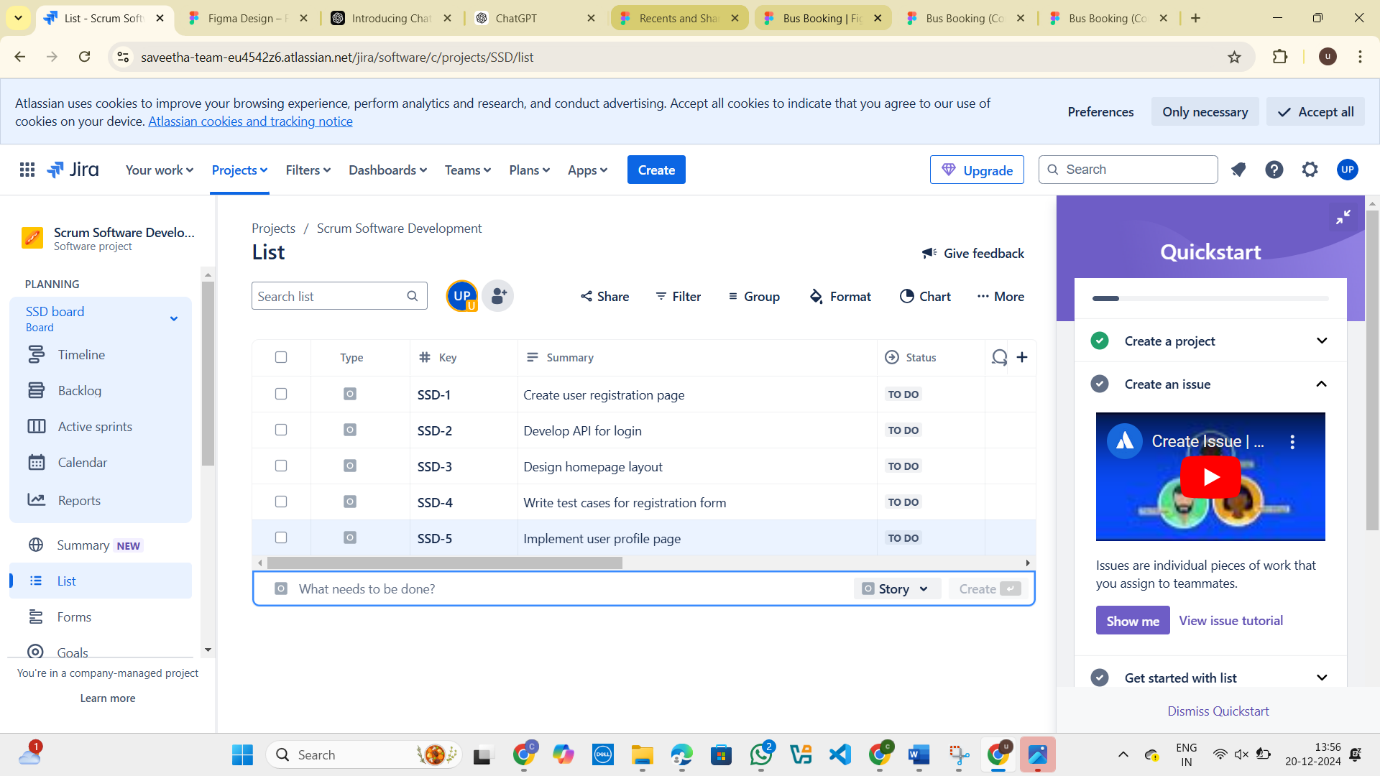
EXP 4:

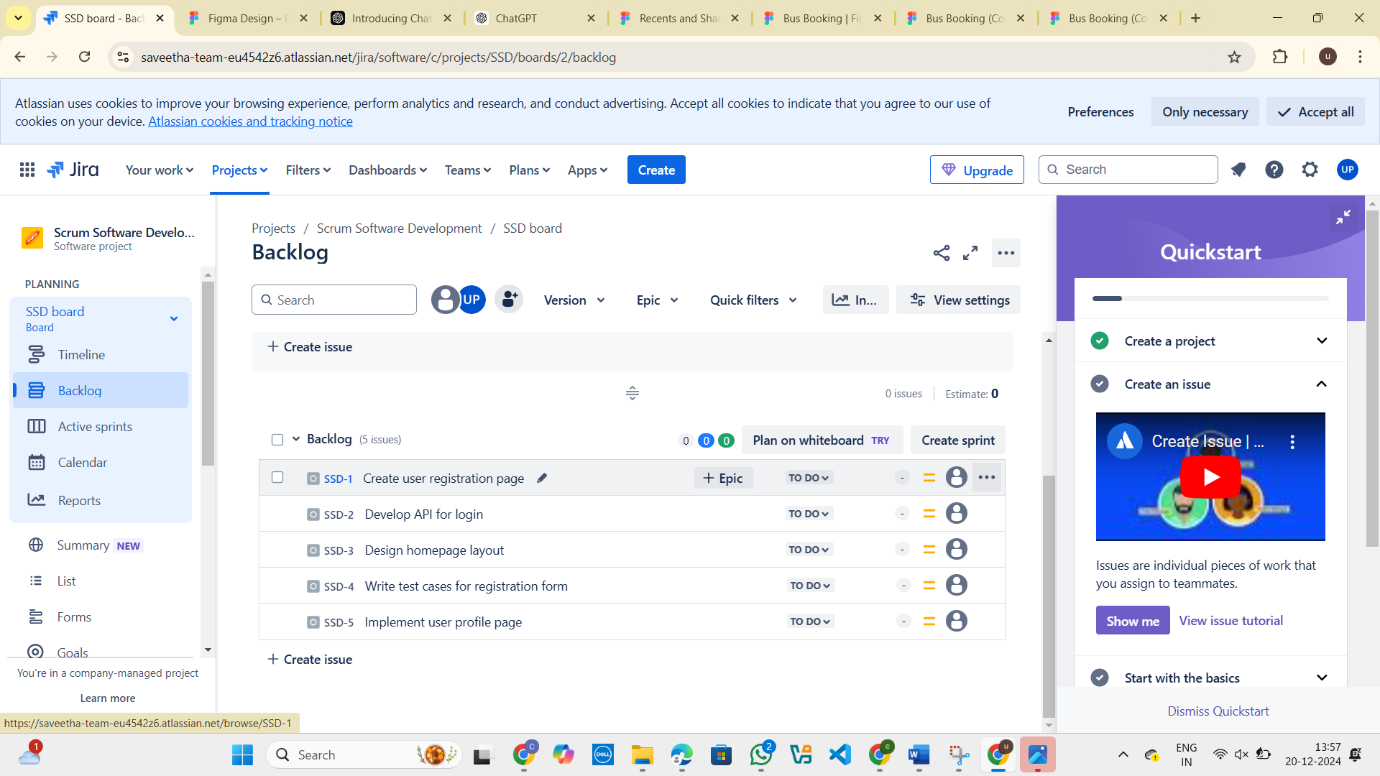
USING SCRUM PROJECT:

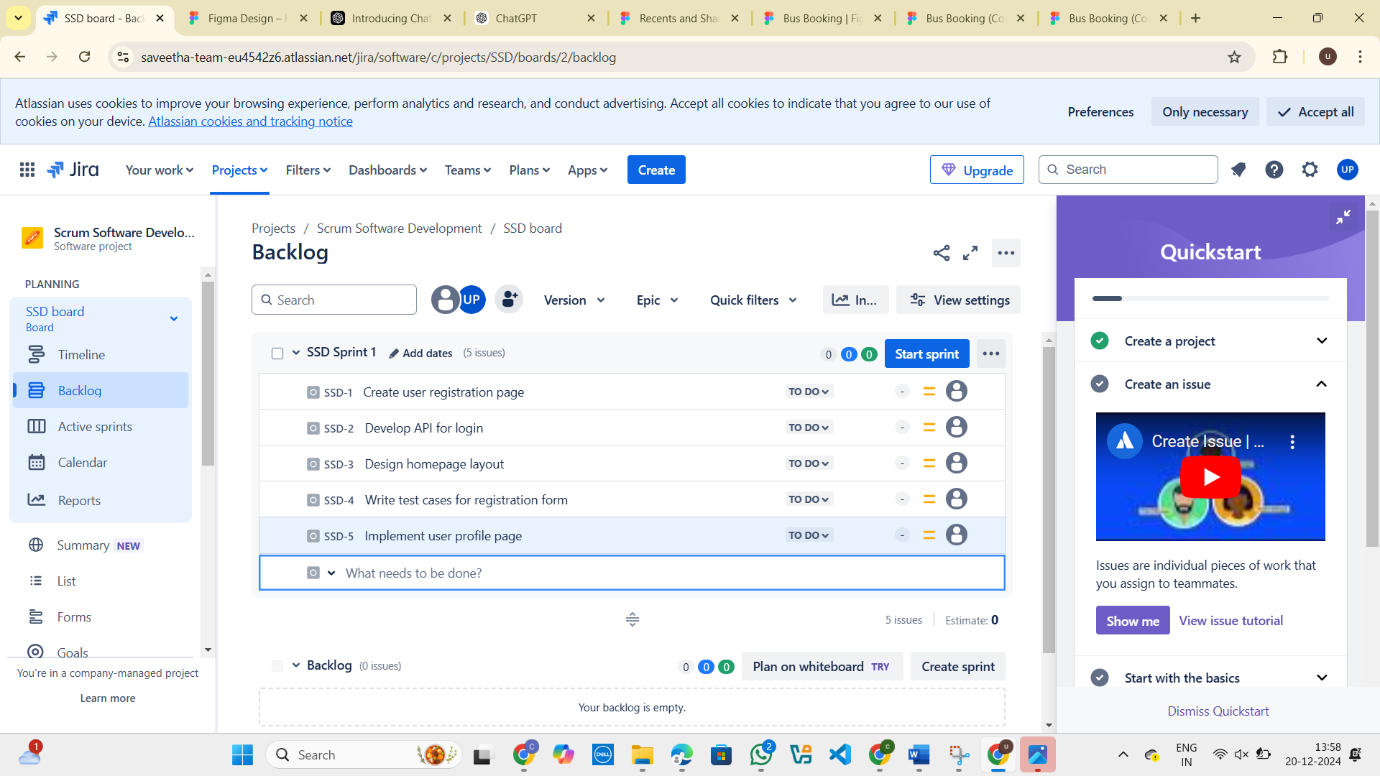
SCRUM SOFTWARE DEVELOPMENT:

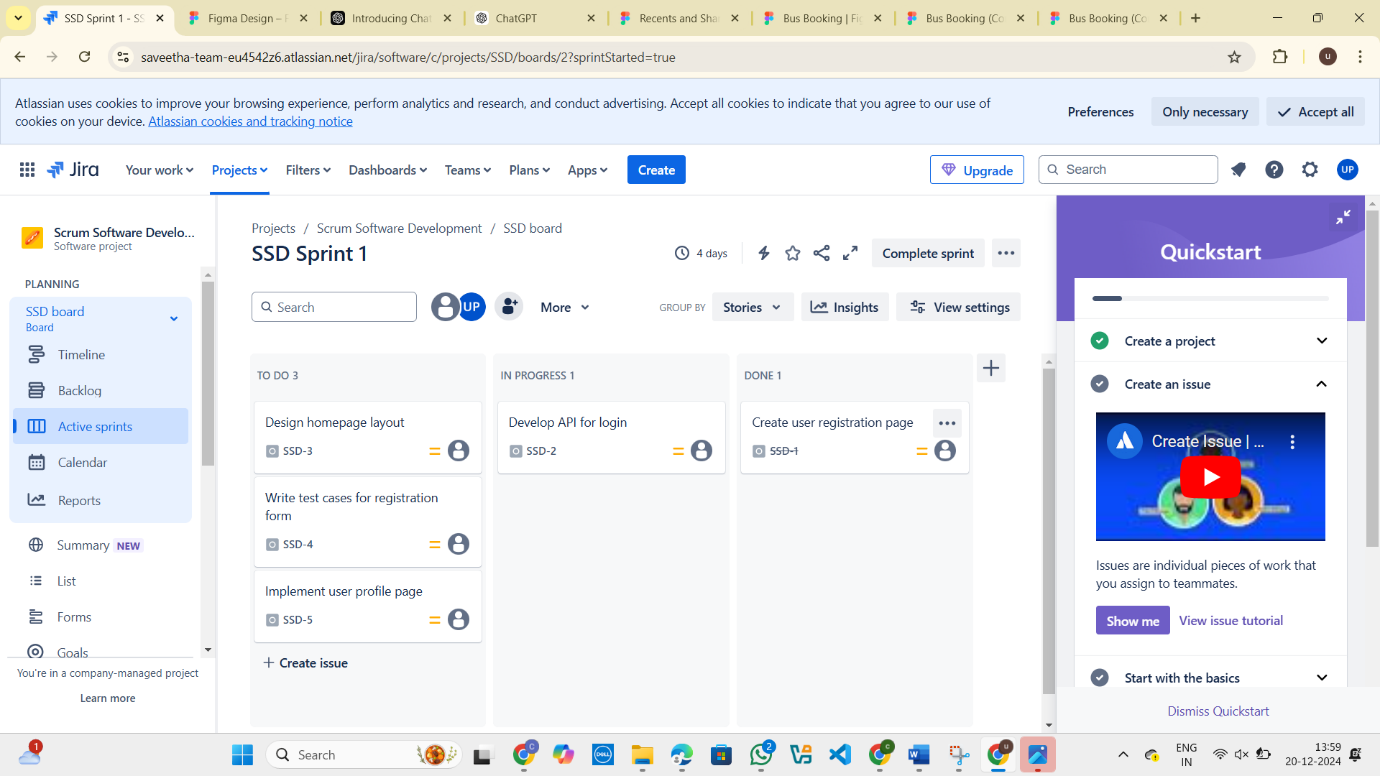












EXP 5:

Use the following requirements for a Library Management System:

• Add a feature to search books by title and author.

• Implement an online book reservation system.

• Generate monthly reports on borrowed books for administrators.

• Enable email notifications for overdue books.

• Add support for QR code scanning for borrowing and returning books.

• Create a user-friendly dashboard for librarians.

• Allow users to review and rate books.

• Integrate a chatbot for user assistance.

• Develop a mobile app version of the system.

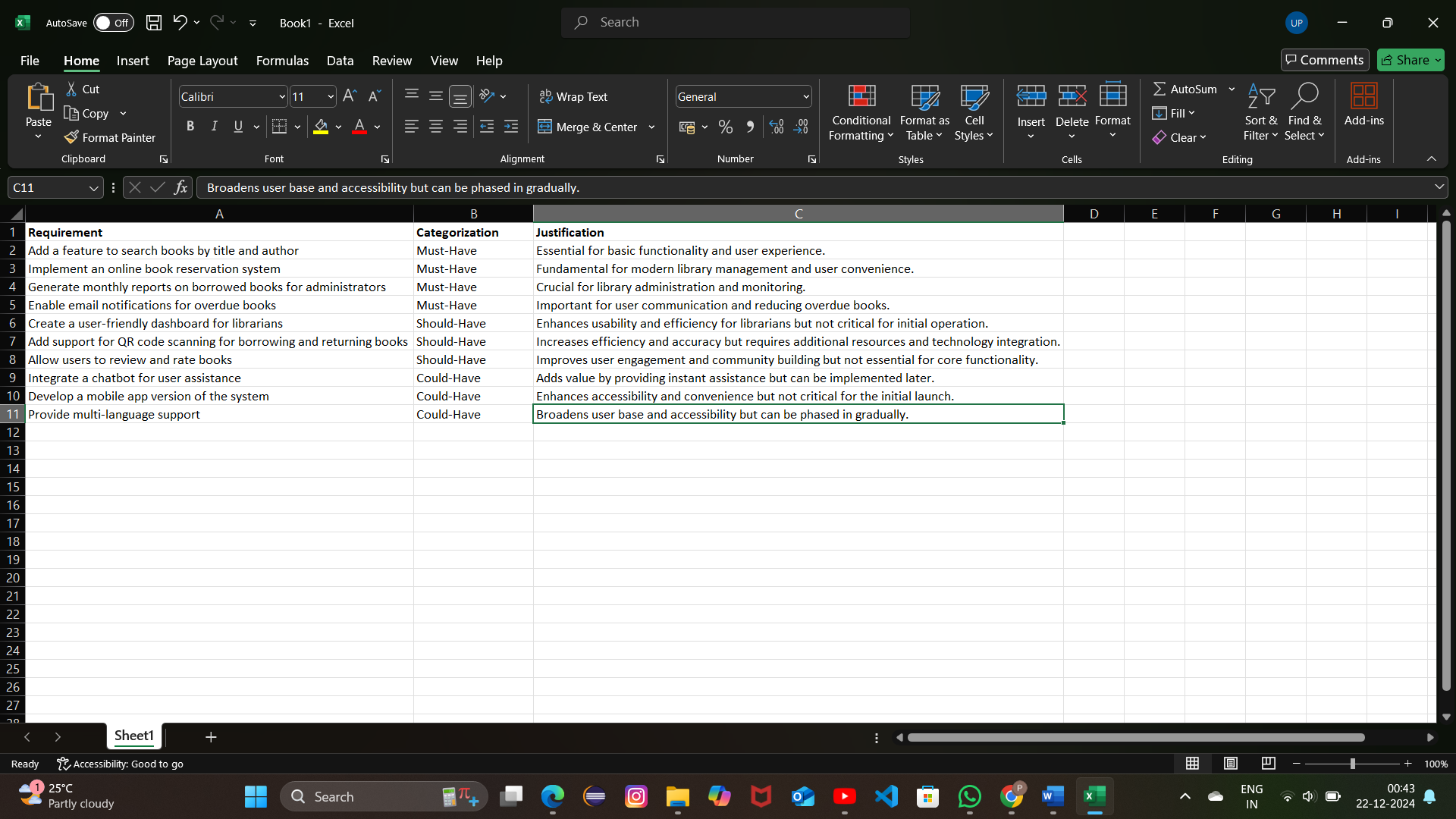
• Provide multi-language support.

Categorize each requirement using MOSCOW Method (Must-Have, Should-Have, Could-Have, or Won’t-Have) based on the following criteria:

• Impact on the users and stakeholders.

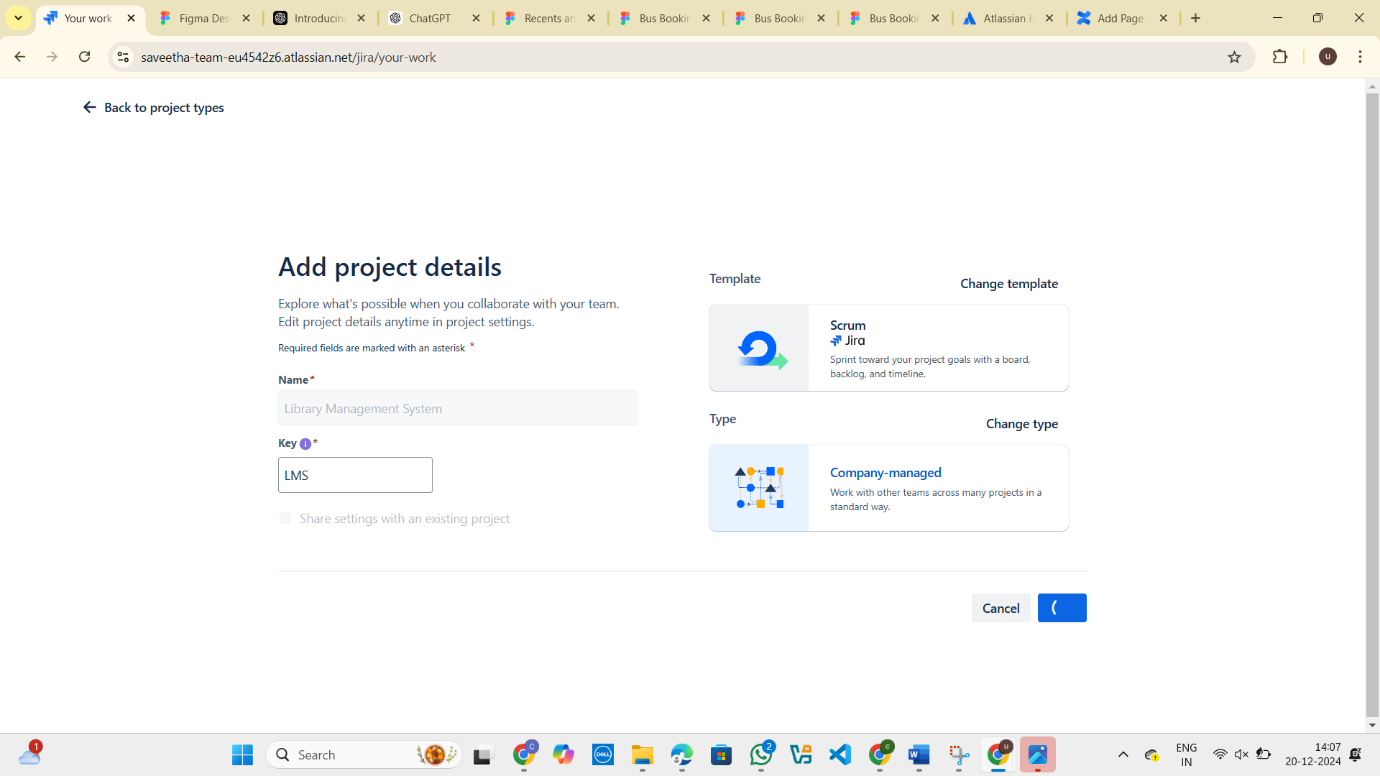
• Feasibility considering time, budget, and resource constraints.

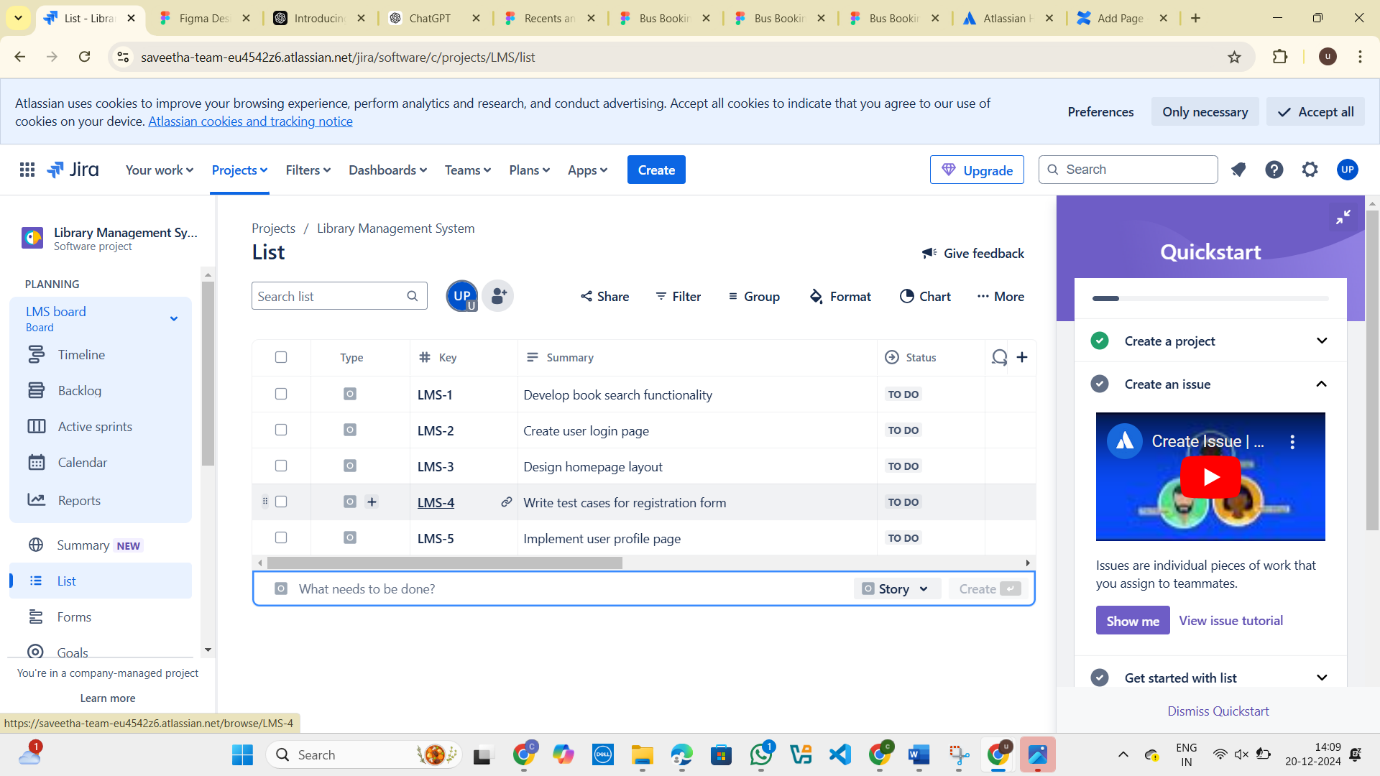
Finally Submit the completed Google Sheet or Excel file with all requirements categorized and justified.

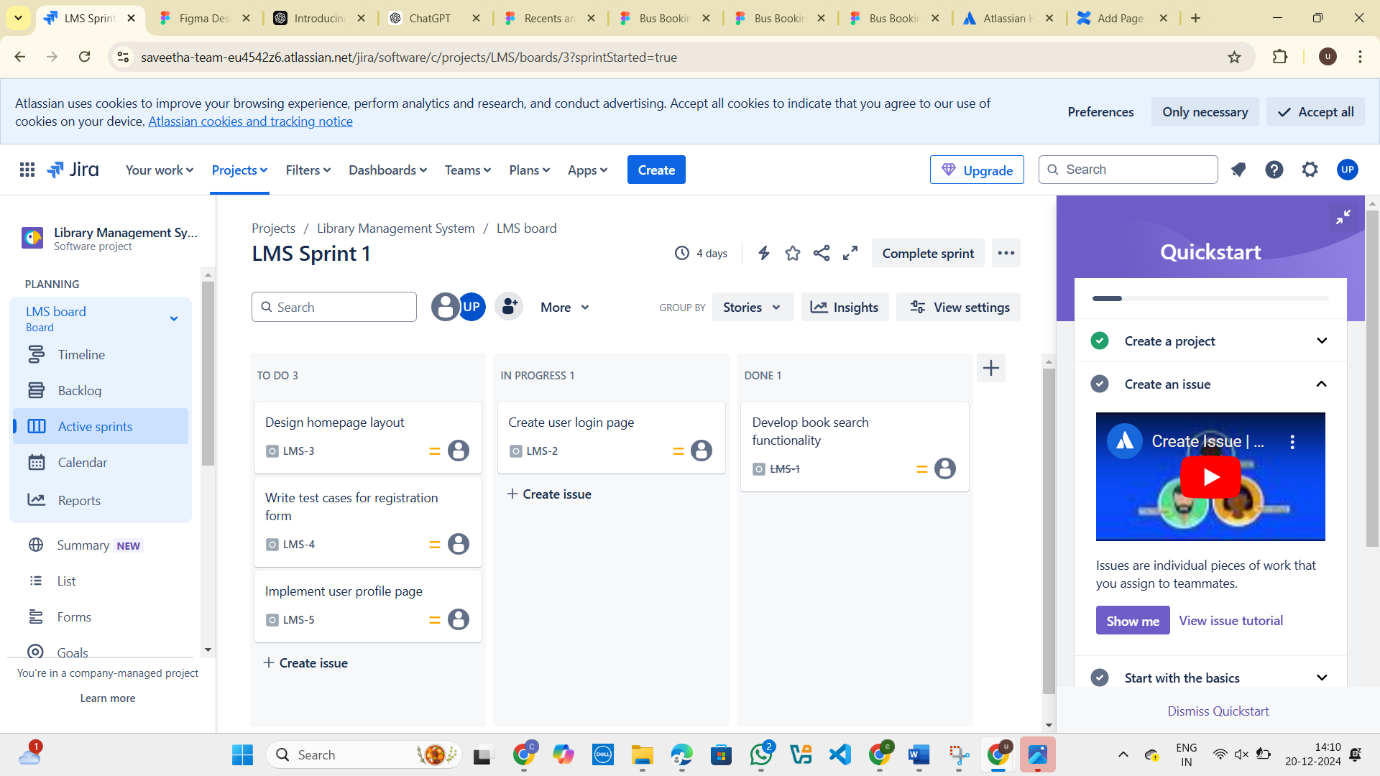


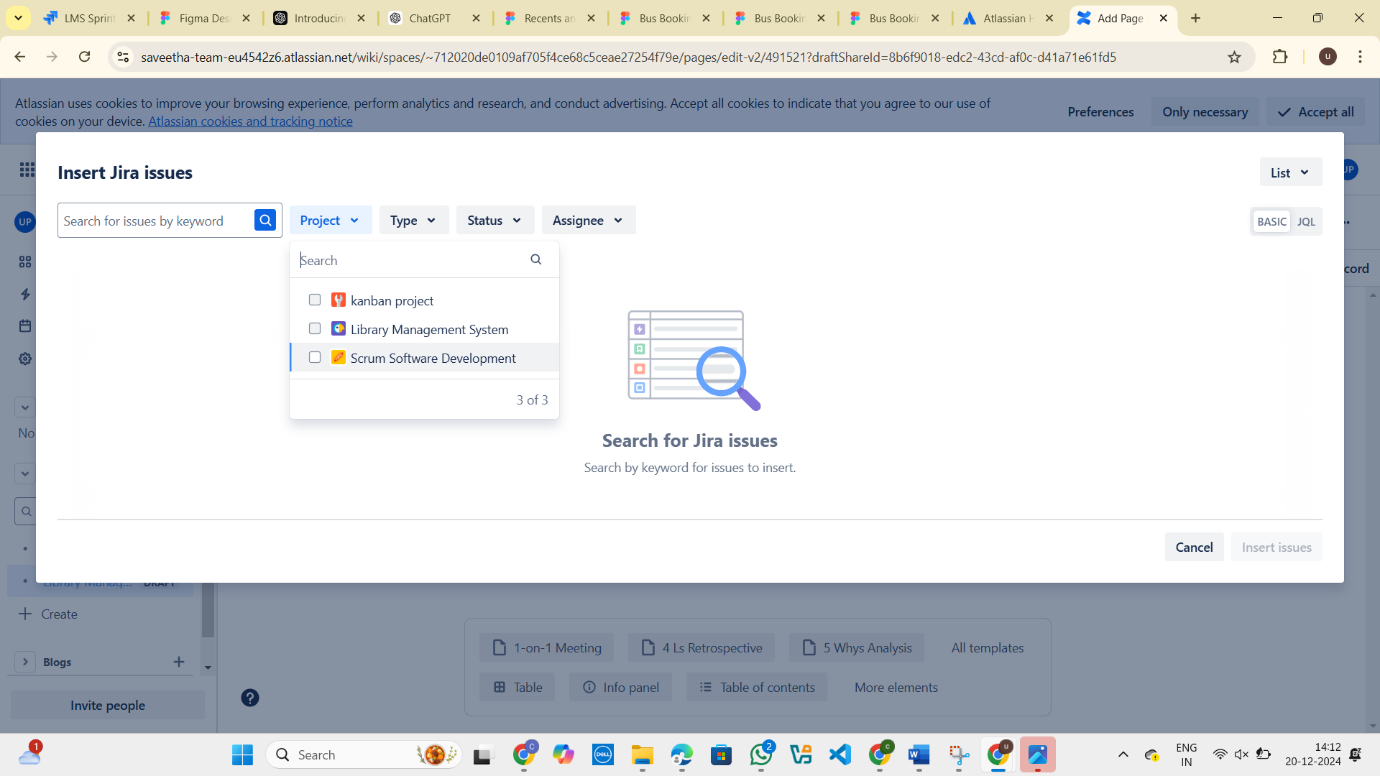
EXP 6:

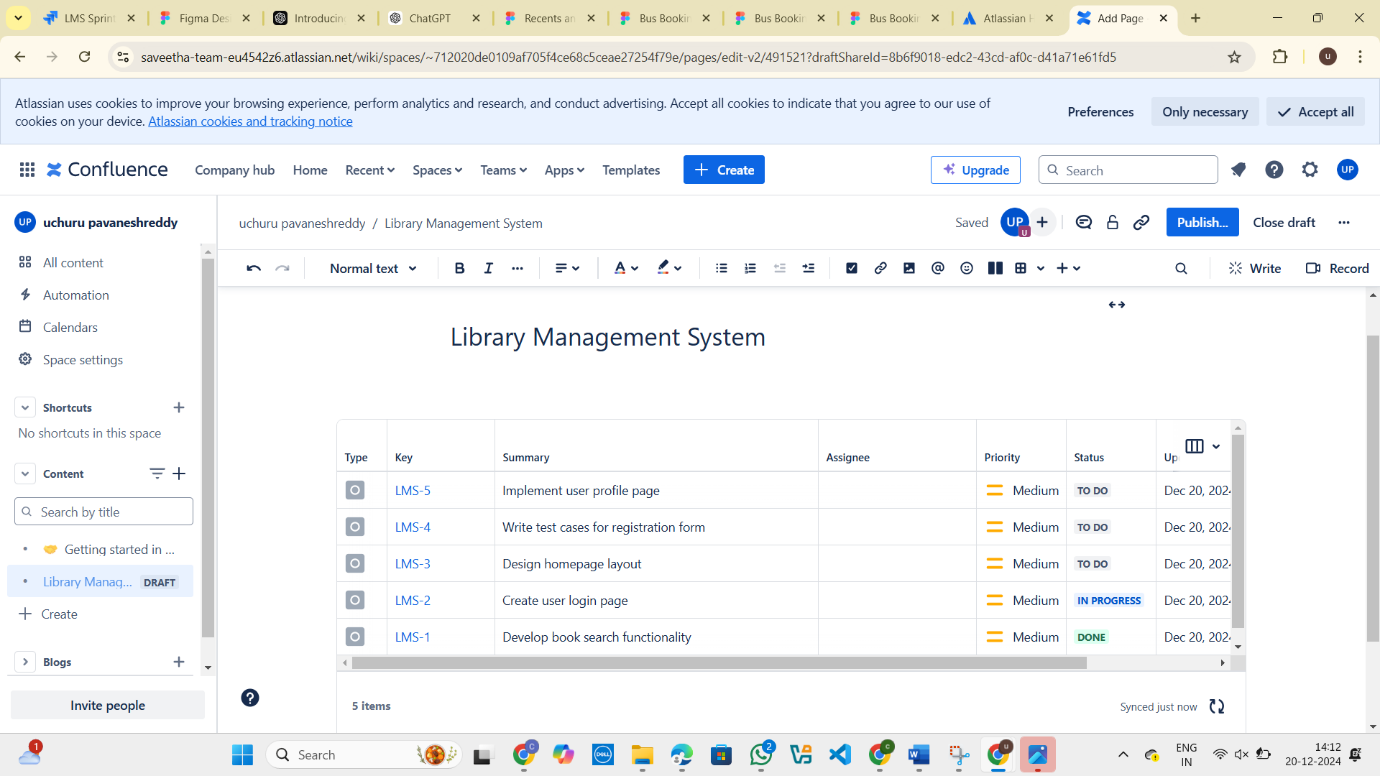
LIBRARY MANAGEMENT SYSTEM:

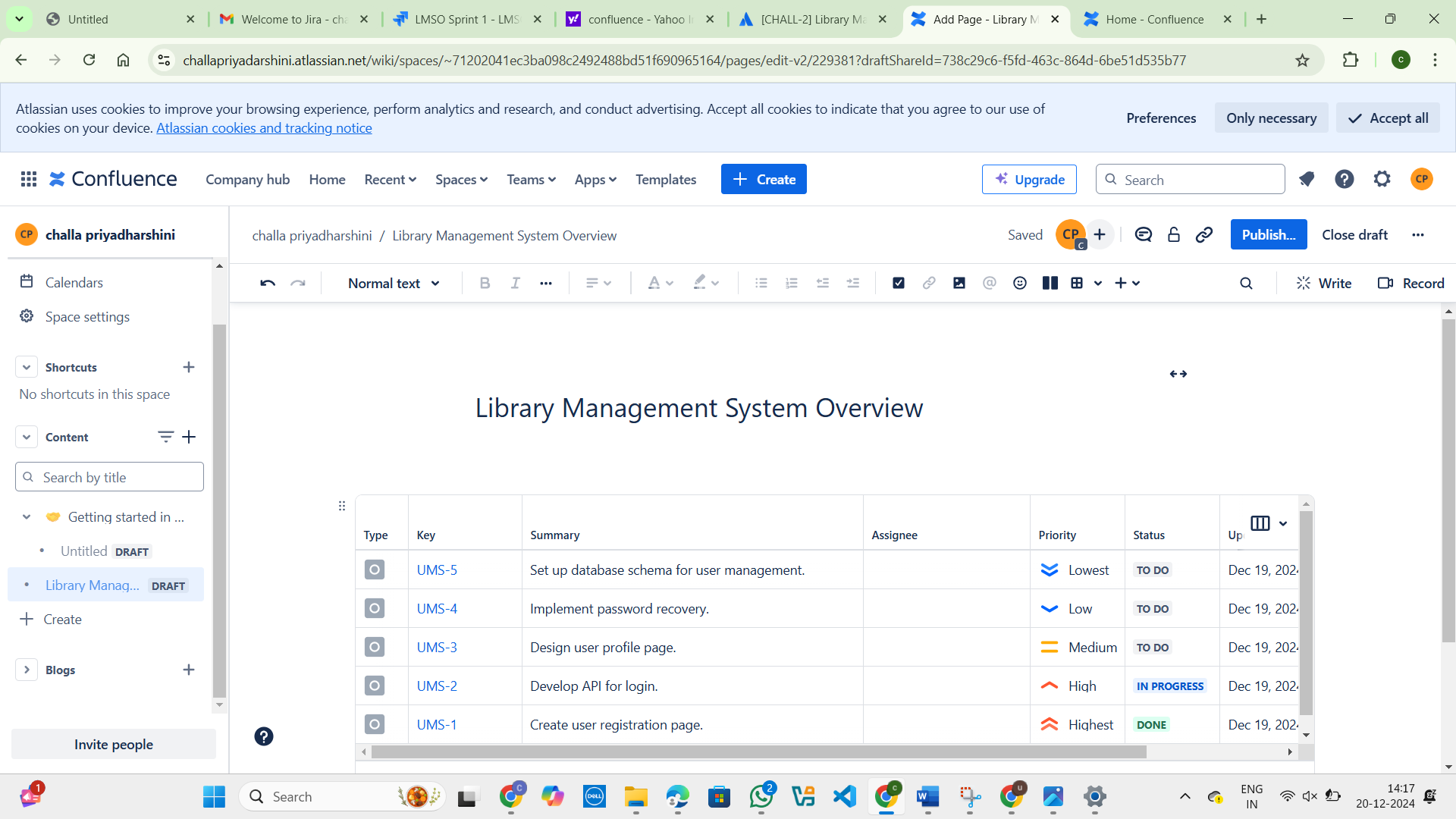






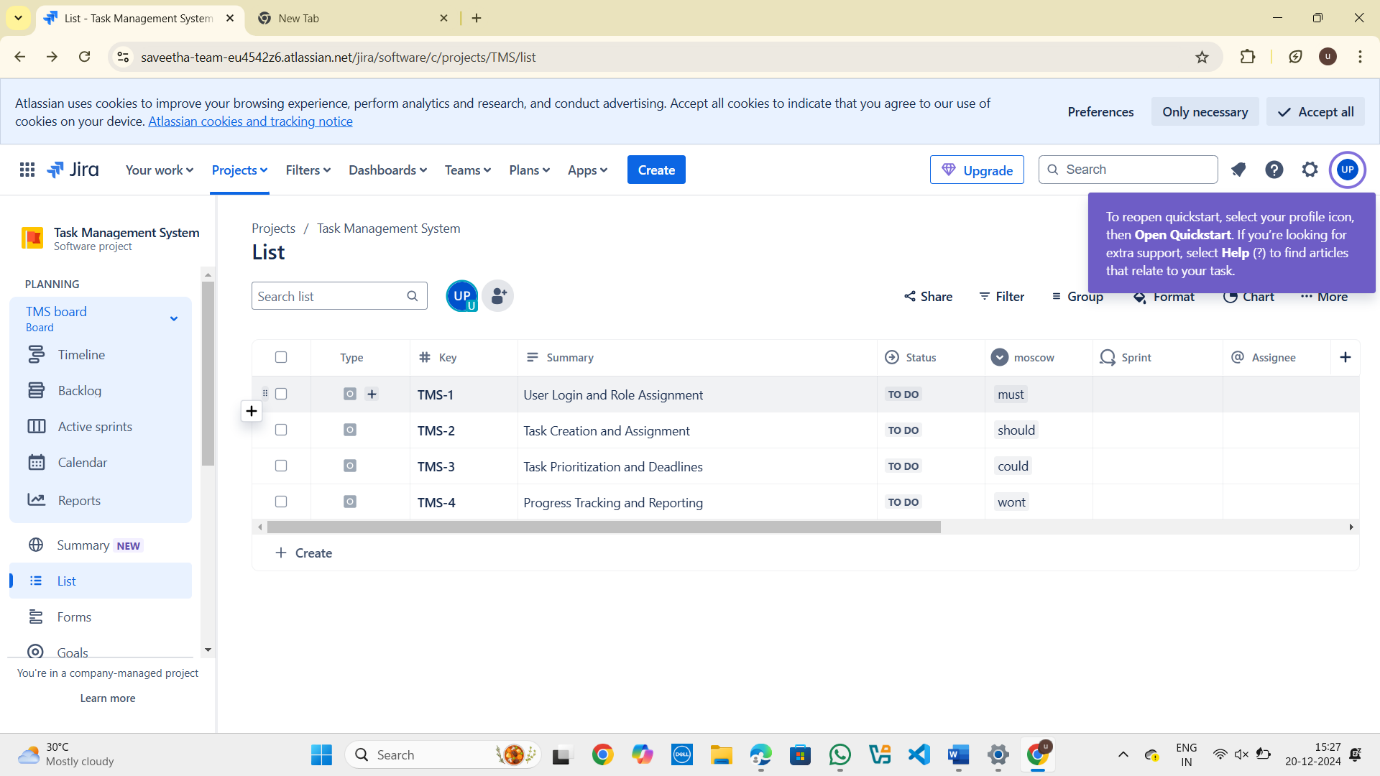


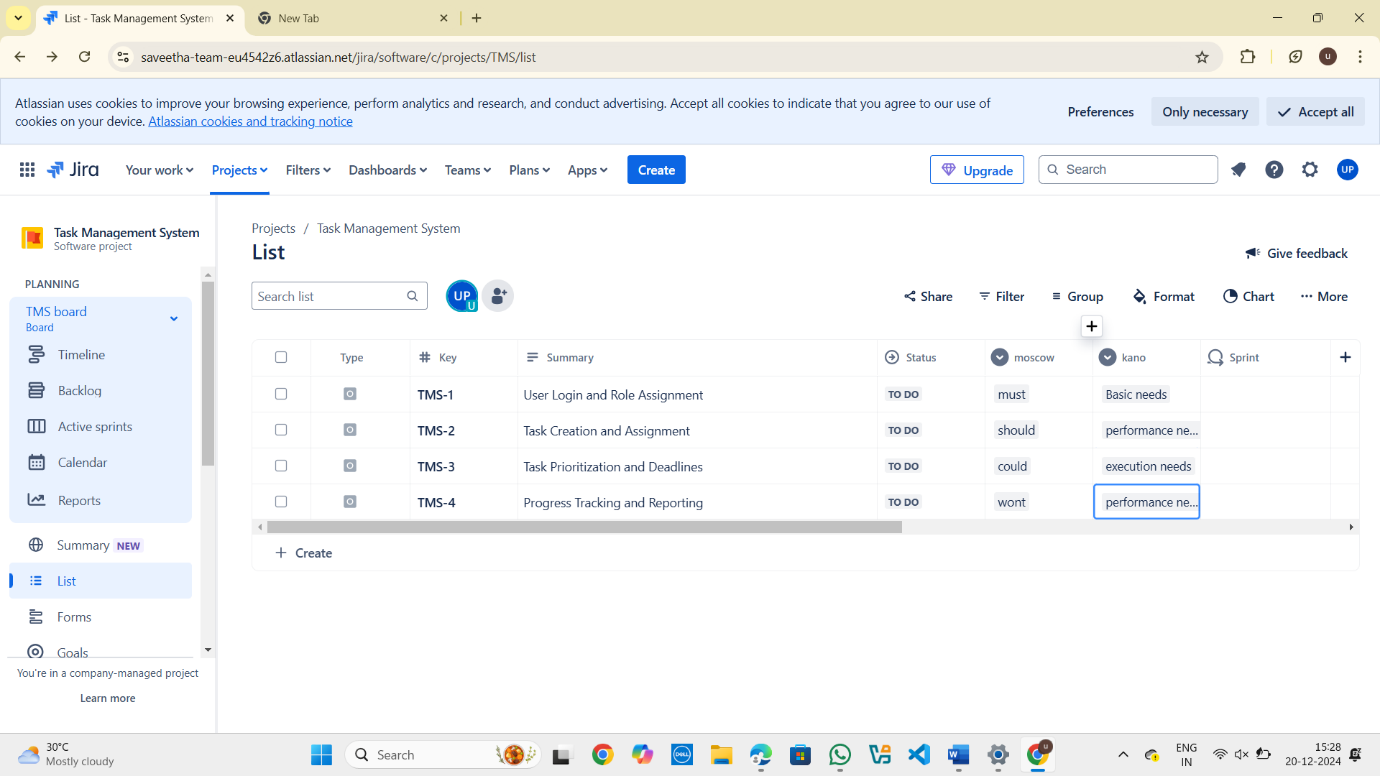




EXP 7:

TASK MANAGEMENT SYSTEM:





Exp 8:

You are tasked with developing an Online Learning Platform. The platform should include the following functionalities:

1. Course Enrollment and Registration

2. Video Lecture Streaming

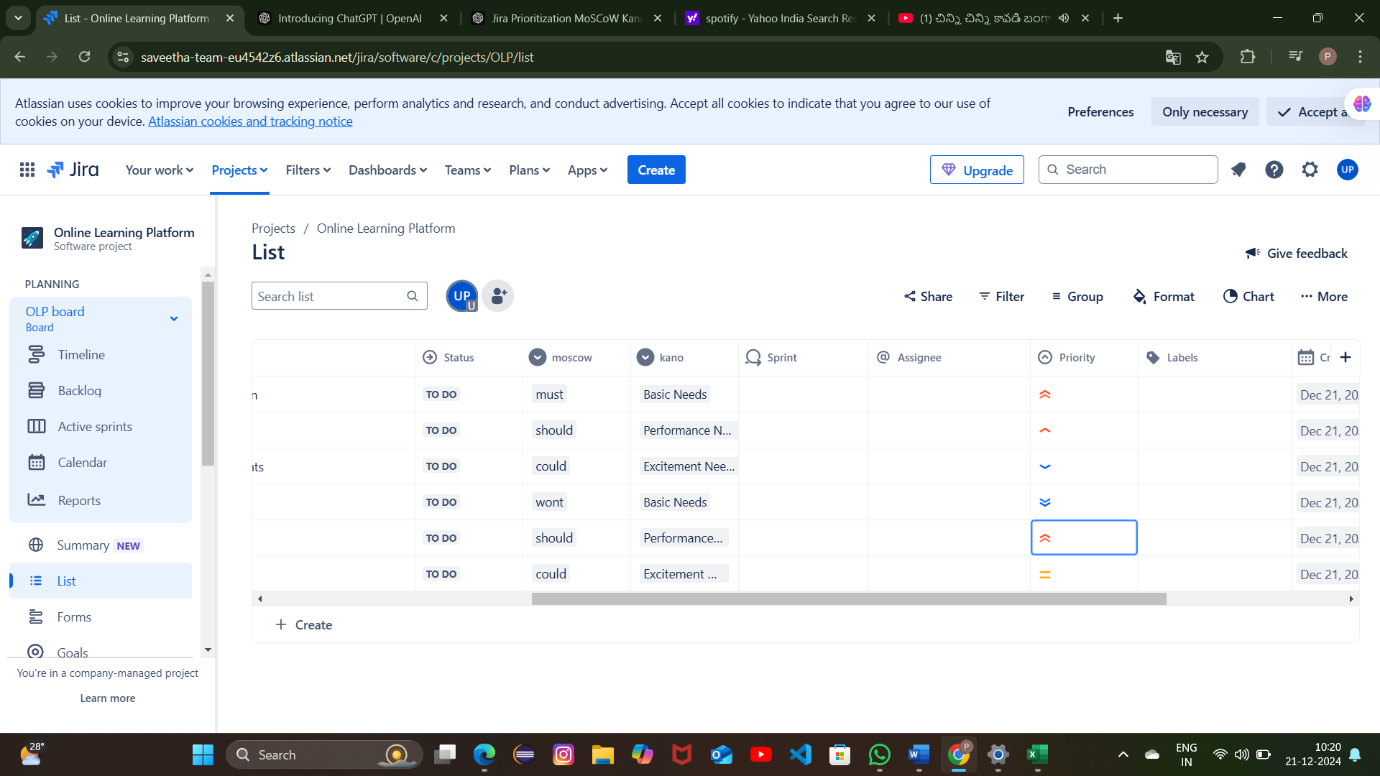
3. Interactive Quizzes and Assignments

4. Progress Tracking Dashboard

5. Peer-to-Peer Discussion Forums

6. Certificate Generation

Use Jira to categorize and prioritize these requirements using MoSCoW and Kano techniques.



EXP 11:

Create a Static Website and Containerize, Build & Serve it using Docker.

Tasks:

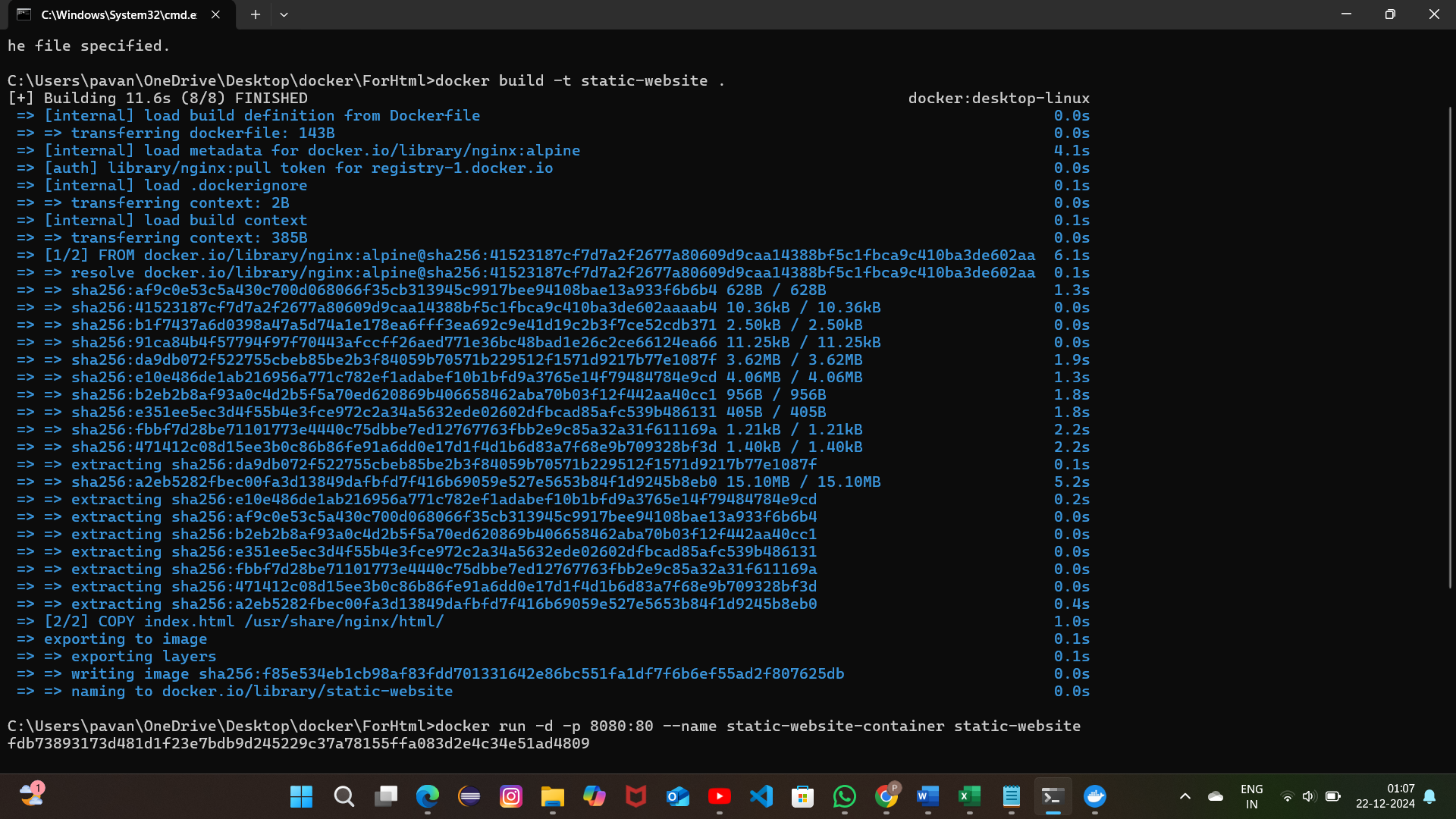
1. Create a Simple Static Website (index.html file) with basic HTML content.

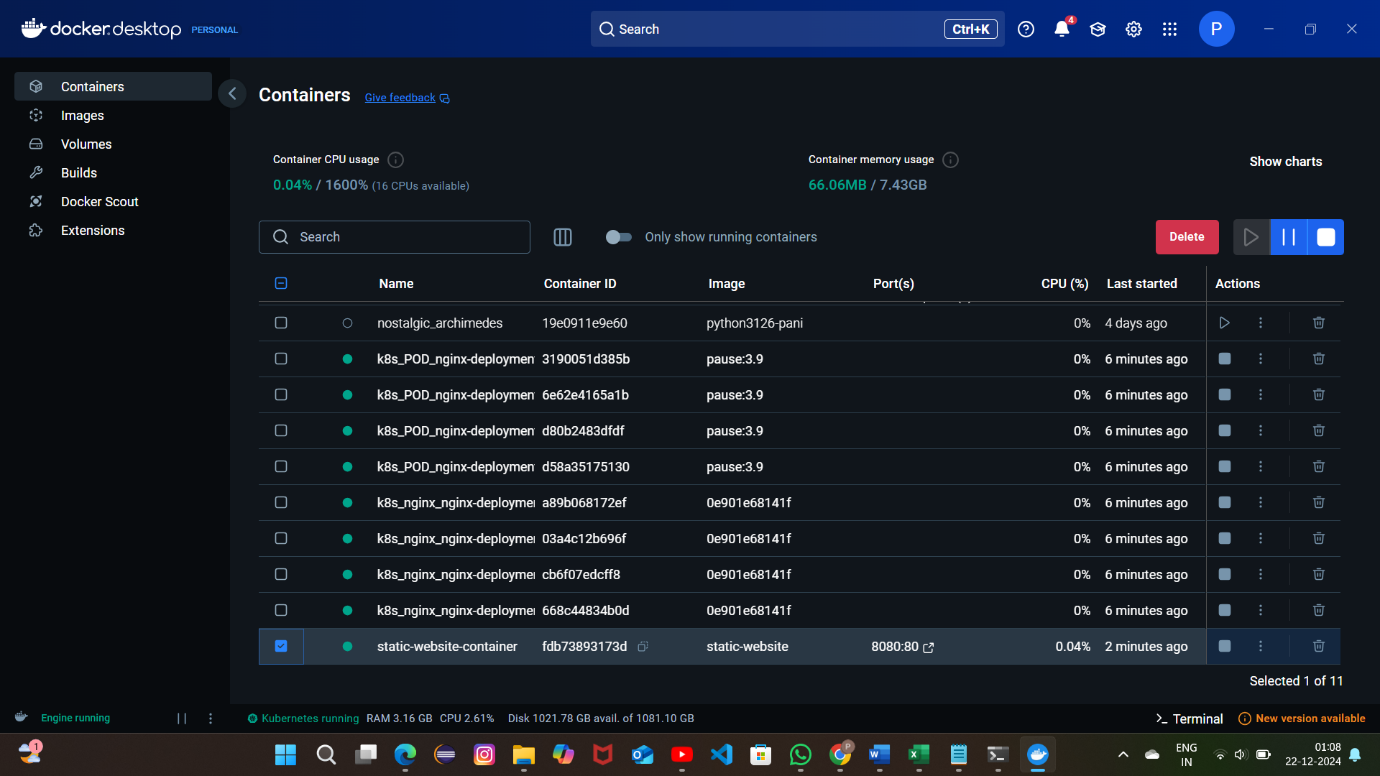
2. Write/create a Dockerfile to serve the website using Nginx.

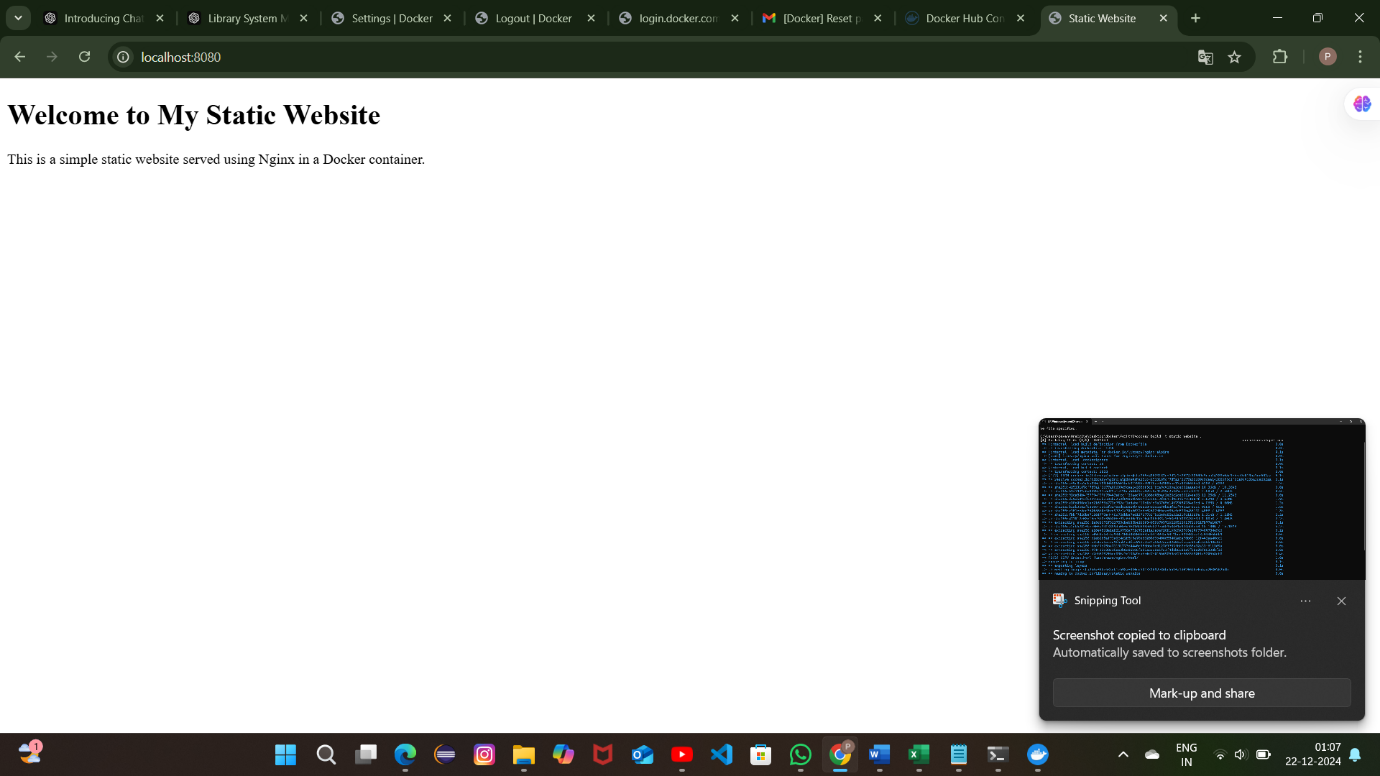
3. Build the Docker Image

4. Run the container:

5. Access the Website using a Browser







EXP 12:

Create a Simple Python Flask API, Containerize the Application, Build & Push the Image using Docker and Deploy the Application using Kubernetes.

Tasks:

1. Create a Simple Flask API by writing a Python file (app.py) with basic endpoints.

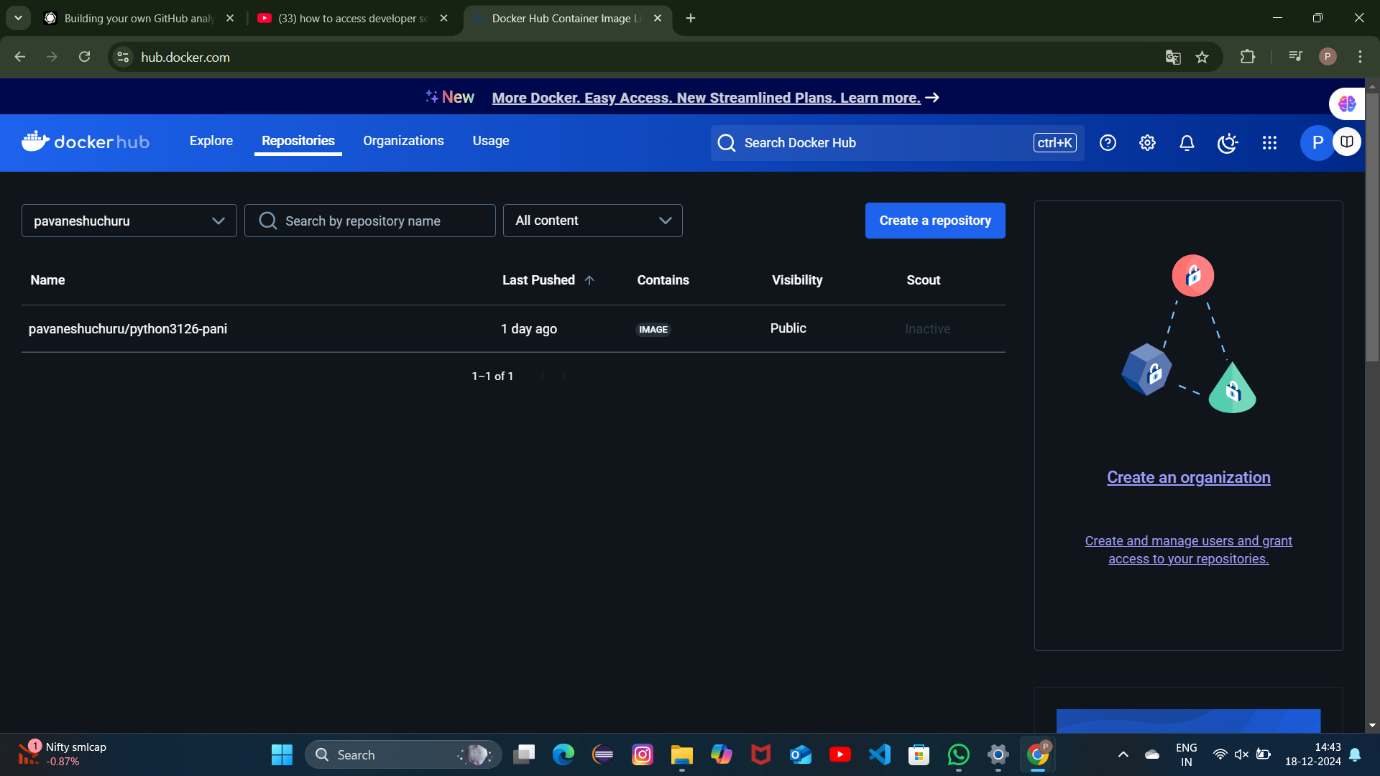
2. Containerize the Flask App using Dockerfile.

3. Build the Image using Docker.

4. Push the Image to Docker Hub.

5. Create Kubernetes manifests (Deployment YAML & Service YAML) to deploy the application.

6. Apply the Manifests and Access the API via NodePort.



EXP 13:

Set up a CI/CD pipeline to automate the building, testing, and deployment of a containerized application.

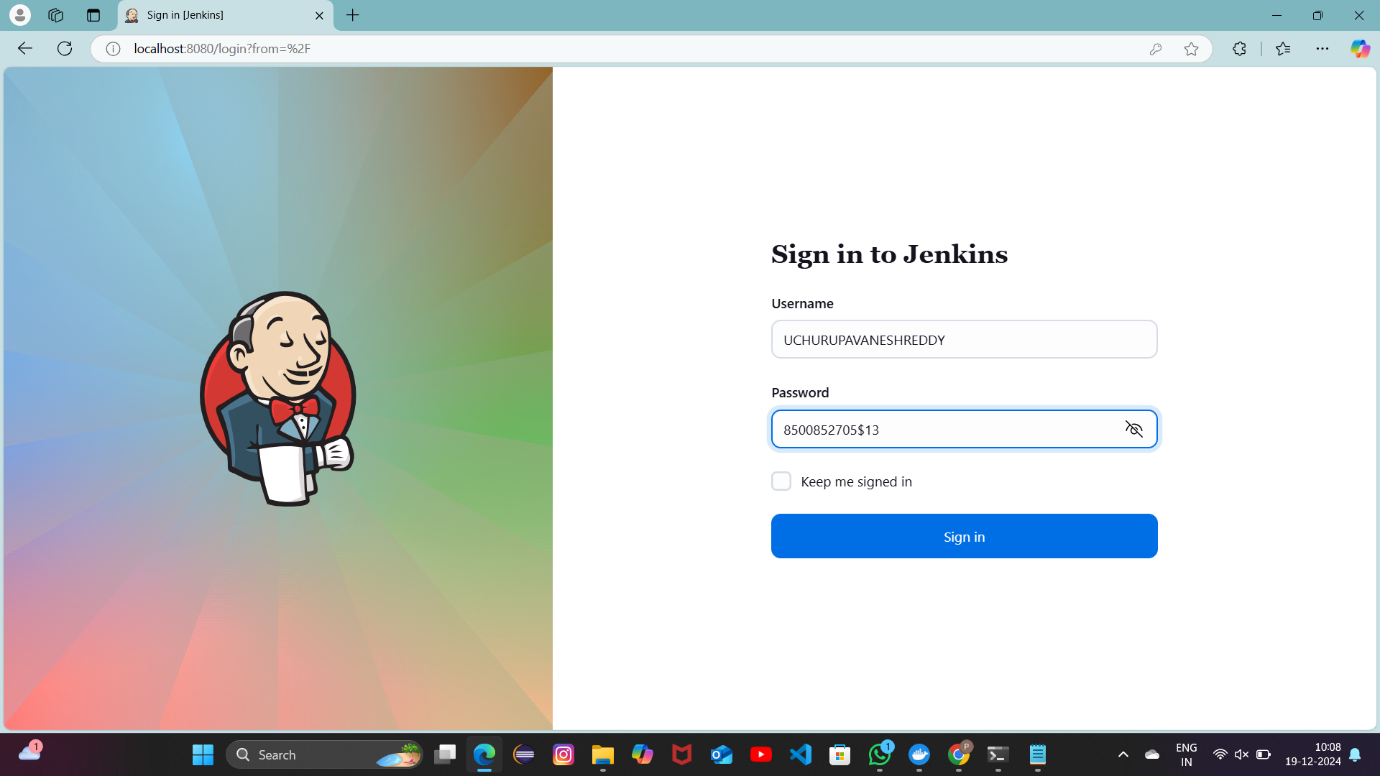
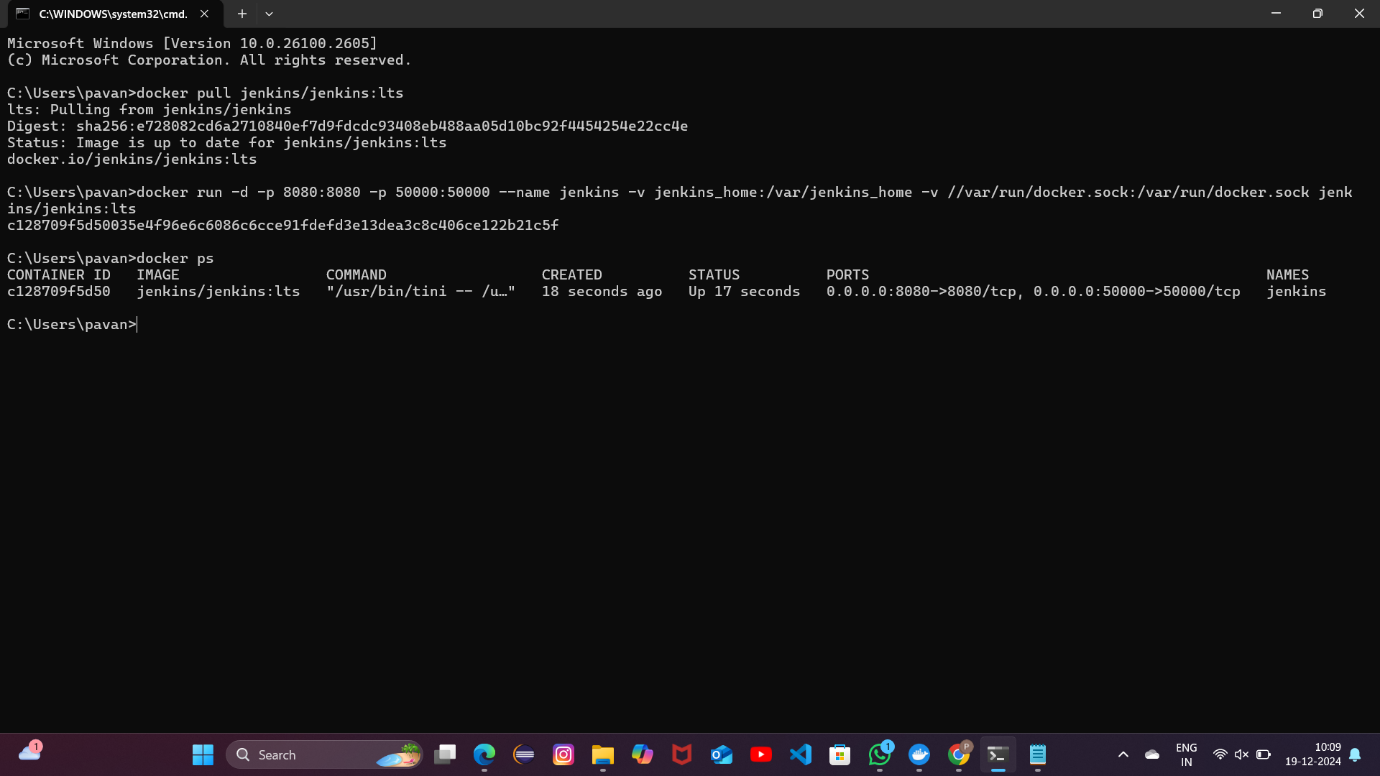
Tasks:

1. Set up Jenkins on a local machine or server. Docker CO4

2. Create a Dockerfile to containerize a sample application.

3. Write a Jenkinsfile to automate the process of building the Docker container, running tests, and deploying to a cloud platform (e.g., AWS or GCP).

4. Configure Jenkins to trigger builds upon code commits or pull requests.



EXP 17:

Push and Pull Docker Images Using Docker Hub

Tasks:

• Create a Docker image for a static HTML website.

• Tag and push the image to Docker Hub.

• Pull the image from Docker Hub and run it on another machine.

• Submit Docker commands and screenshots for each step

