GitHub Repo Link:-

https://github.com/kkasliwal05/To_do_list

This is my GitHub repo link in which I have implemented the TO-DO-List app.

Assignment 1: Product Backlog

Product Vision

Build a user-friendly To-Do List App that allows users to manage their daily tasks effortlessly with an intuitive interface.

Product Backlog

ID	User Story	Priority	Story Points
1	As a user, I want to add a new task so I can keep track of things to do.	High	3 SP
2	As a user, I want to see a list of all my tasks so I can plan my day.	High	3 SP
3	As a user, I want to mark tasks as completed to track progress.	High	2 SP
4	As a user, I want to delete tasks that are no longer needed.	Medium	2 SP
5	As a user, I want completed tasks to appear differently (e.g., strikethrough) so I can distinguish them.	Medium	1 SP
6	As a user, I want to edit existing tasks to make changes when needed.	Medium	3 SP
7	As a user, I want tasks to be saved locally so I don’t lose them on refresh.	High	5 SP

Assignment 2: Sprint Planning

Sprint Duration: 1 Week
Sprint Goal:
Deliver a basic working version of the To-Do List App that allows adding, viewing, completing, and deleting tasks, with data saved in the browser.

Step 1: Reviewed Product Backlog
Went through the backlog and picked the most important features. Just focused on what's needed to get the basic app up and running.

Step 2: Selected Items for Sprint

Selected ID	**User Story**
1	Add a new task
2	View task list
3	Mark task as completed
4	Delete task
5	Store tasks locally

Step 3: Task Breakdown

User Story ID	**Tasks**
1	- Design input form - Add "Add Task" button - Store task in localStorage
2	- Load tasks from storage - Display list dynamically
3	- Add checkbox/action to mark as complete - Update status in storage

User Story ID	Tasks
4	- Add delete option per task - Remove task from storage

Step 4: Story Point Estimation

Task	Estimated Points
Add Task	3 SP
Display Tasks	3 SP
Mark Complete	2 SP
Delete Task	2 SP
Local Storage	5 SP

Step 5: Sprint Backlog

Task	Status
Create task input & Add button	
Save task using localStorage	
Display all tasks on screen	
Mark task complete (style change)	
Delete task from list/storage	
Load tasks on page reload	

Outcome:
A clearly defined sprint scope with focused deliverables and estimated efforts, aligned with the vision of creating a functional core for the To-Do List App.

Assignment 3: Daily Stand-Ups

Daily Scrum Log

Day	What I Did Yesterday	What I Plan Today
1	Set up project files and base UI with HTML/CSS	Begin task addition feature with JavaScript
2	Implemented "Add Task" and connected to localStorage	Work on loading tasks and styling task view
3	Fetched and styled task list from localStorage	Add "Mark as Completed" feature with strikethrough
4	Implemented task completion logic and styling	Start developing delete functionality
5	Finished delete feature and refined UI/persistence	Final test + minor UI tweaks

Outcome:
Daily progress tracking helped maintain development momentum, identify early issues, and stay on track.

Assignment 4: Sprint Review

Work Delivered

Feature	**Status**	**Remarks**
Add Task	Completed	Tasks can be added and saved
View Tasks	Completed	All tasks displayed in list view
Mark as Completed	Completed	Visually updated with strikethrough
Delete Task	Completed	Tasks removable with button
Local Storage	Completed	Tasks persist after refresh
Edit Task	Completed	Inline edit option added
Clean UI	Completed	Simple, clutter-free design

Self-Review Notes

- Functional core was delivered successfully.
- App is intuitive and meets the MVP target.
- Future improvements: animations, mobile view optimization, clear completed tasks, task priority, and due dates.

Assignment 5: Sprint Retrospective

What Went Well

- All main functionalities were implemented on time.
- Estimations were realistic and accurate.
- UI was kept clean and consistent.
- No blockers; smooth workflow.

What Could Be Better

- Manual testing slowed down final days.
- UI needs better responsiveness for mobile devices.
- No real-time feedback when actions occur.
- Testing automation was missing.

Actionable Next Sprint Goals

Task	Description	Priority
Toast Notifications	Add user confirmation feedback	Medium
Responsive Design	Improve layout for small screens	High
Clear Completed	Add button to remove all completed tasks	Medium
Setup Testing	Start with basic automated testing	Low

Assignment 6: Impediment Log

ID	Date	Issue	Resolution	Status	Owner
–	–	No blockers faced during development	–	–	Khushi Kasliwal

Outcome:

No technical or workflow blockers were encountered, allowing uninterrupted and timely delivery of the sprint objectives.

Assignment 7: Definition of Done (DoD)

A feature or task is only marked "Done" if all the following are true:

1. The user story has been fully implemented.
2. Manual testing has confirmed correct functionality.
3. No critical bugs or errors exist.
4. The feature matches the app's UI style.
5. Data is saved and retrievable via localStorage.
6. Code is clean, readable, and well-documented.
7. Reviewed against original requirements and accepted.
8. Integrated successfully with other features.
9. App functionality remains intact post-integration.
10. Feature status updated in project tracker/backlog.

DoD Adoption:
Used consistently for all features to ensure quality, completeness, and smooth integration.