Hustlr - (Sprint 4)

✅ Sprint Backlog

Task	Owner	Status	Due Date	Story Points
Refactor UI for Home page	Front-End	Completed	04/05/25	4
Design UI for Admin job retrieval and deletion	Front-End	Completed	30/04/25	4
Design UI for Accepted Jobs and Rejected Jobs	Front-End	Completed	19/04/25	5
Implement Recent Activities Page	Team	Completed	30/04/25	4
Implement Accepted Jobs and Rejected Jobs Logic	Team	Completed	03/05/25	6
Backend endpoint for applications	Back-end	Completed	30/04/25	5

Store Applications in DB	Back-end	Completed	30/04/25	3
Fetch Applications in DB	Back-end	Completed	01/05/25	3
Creation of UML diagrams	Team	Completed	30/04/25	2
Implement Logic for Freelancer applications and validation	Team	Completed	02/05/25	5
Deploy on Azure	Front-End	Completed	04/05/25	6
UATs in correct format, code coverage tool integrated on GitHub	Back-end	Completed	03/05/25	5
60% Code coverage	Team	Completed	04/05/25	7
Implement Logic for milestones.	Team	Completed	29/05/25	5
Implement Logic for “Payments”	Team	Pending	30/04/25	6

📱 Sprint User Stories

User Story 1: Create Milestones as Client ✨

Who : Client

What: Add milestone to a listing

Why : Ensures freelancers have knowledge of tasks

Acceptance Criteria

- ✓ **Given:** the client is on the jobs page. **When:** they fill in the job form with milestones.
  Then: the job gets posted with milestones.

User Story 2: Apply to jobs as Freelancer � 🔐

Who : Freelancer

What : : Can apply to job listings

Why : : Ensures Client knows there's potential applicants

Acceptance Criteria

- ✓ **Given:** a freelancer is logged in and navigates to the job listings page. **When:** the job they want is shown **Then:** they click 'apply' to apply to the job

User Story 3: Client can access Applicants🔐

Who : Client

What : : I want to view applicants to a job

Why : Ensures Client can keep track of applicants

Acceptance Criteria

- ✓ **Given:** the client is on their job dashboard. **When:** they click the view button on a job **Then:** they are show current applicants

---

User Story 4: Accept Applicant as Client 👨‍💼

Who : Client

What: Accept an applicant listing with details like budget, deadline, and skills required.

Why : Ensures a client can accept an applicant for a job

Acceptance Criteria

- ✓ **Given:** the client is on their job listings page. **When:** they view the list of applicants. **Then:** they can accept a particular applicant

User Story 5: Reject Applicant as Client 👨‍💼

Who : Client

What: Reject an applicant

Why : Ensures a client can reject an applicant for a job

Acceptance Criteria

- ✓ **Given:** the client is on their job listings page. **When:** they view the list of applicants . **Then:** they can reject a particular applicant.

User Story 6: View accepted jobs as a Freelancer

Who : Freelancer

What: I want to view the jobs I've been accepted for

Why : So I know what Jobs I can start working on

Acceptance Criteria

- ✓ **Given:** a freelancer accesses the jobs dashboard **When:** they view the page . **Then:** the user sees the jobs they have been accepted to do

User Story 7: View rejected jobs as a Freelancer🙋‍♂️

Who : Freelancer

What: I want to view the jobs I've been rejected for.

Why : So I know what jobs I can stop being hopeful to work on.

Acceptance Criteria

- ✓**Given:** a freelancer accesses the jobs dashboard .**When:** the page loads. **Then:** the user sees the jobs they have been rejected to do

User Story 8: View any jobs as an Admin🙋‍♂️

Who : Admin

What: I want to view all the jobs available on the platform

Why : **So I can monitor the jobs and their descriptions**

Acceptance Criteria

- ✓**Given:** an admin accesses the jobs dashboard .**When:** the page loads. **Then:** the admin sees all the jobs that have been posted.

User Story 9: Delete any jobs as an Admin 👨‍💼

Who : Admin

What: I want to be able to delete any job on the platform

Why : So I can remove any malicious or fake jobs on the platform

Acceptance Criteria

- ✓ **Given:** an admin accesses the jobs dashboard. **When:** view the page . **Then:** the admin can delete any job permanently

Requirements Gathering

Client Page Requirements

- Should clients vet a freelancer(receive an application), or should it be first come basis?

Milestone Requirements

- What information is required for a milestone (budget, deadline, etc.)?

Admin Views

- Should admins be able to see all Jobs ?
- Should admins be able to remove any job ?

📝Product Backlog

User Story	Priority	Story Points	Status
As a freelancer, I want to view all my ongoing projects in one place so I can track progress.	High	5	Completed
As a client, I want to create and manage contracts and task	High	6	Completed

milestones for my jobs.			
As an admin, I want to view reports and job metrics so I can monitor platform activity.	High	6	Completed
As a user, I want a well-designed homepage and milestones UI so I can navigate easily.	Medium	4	Completed
As a freelancer, I want to view recent activities and stats on a dashboard for quick insights.	Medium	5	Completed
As a client or freelancer, I want the ability to make and track payments securely.	High	6	Completed
As an admin, I want to export financial reports so I can perform audits.	Medium	4	Completed
As a user, I want the About page to explain what the platform does.	Low	2	Completed
As a developer, I want to have 60%+ test coverage so that the app is maintainable.	High	8	Completed
As a team, we want CI/CD with GitHub integration to automate testing and coverage checks.	High	5	Completed
As a stakeholder, I want the product to be deployed on Azure for demonstration and use.	High	7	Completed
As a developer, I want to store and retrieve payment records from the DB.	High	6	Completed
As a user, I want a unique icon in the header to personalize my experience.	Low	2	Completed
As an admin, I want to see UML diagrams to understand the system structure.	Low	2	Completed

🔁Sprint Retrospective

What Went Well:

- All tasks were completed on time, showcasing strong collaboration and well-distributed workload across the front-end, back-end, and full team tasks.
- Code coverage target (60%+) was met and successfully integrated into GitHub CI.
- Smooth deployment to Azure by the final sprint day with no critical post-deploy bugs.
- Payment logic and database integration were implemented and tested with minimal friction.
- UATs and UML diagrams were submitted in the correct format, improving clarity and maintainability.

What Could Be Improved:

- Multiple tasks overlapped near the mid-sprint period (13th–16th), which caused short delays in merging and verifying features.

Action Items:

- Reinforce end-of-day summary check-ins, especially in the final sprint phase.

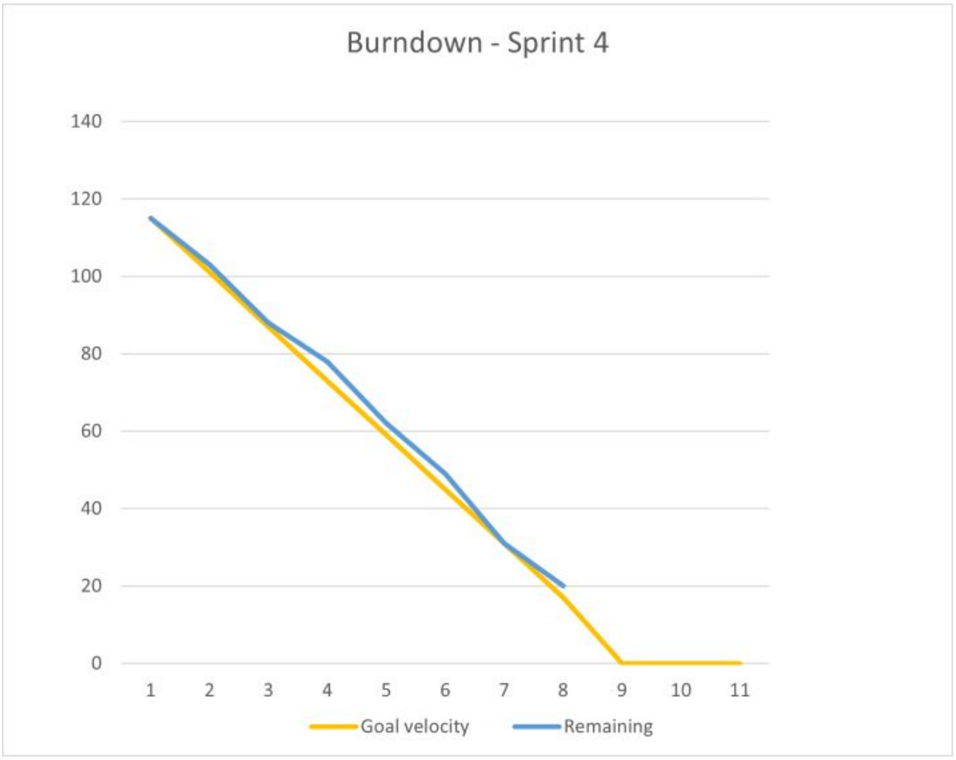
📅Daily Stand-Up Summaries

Day	Summary
Day 1	Sprint planning finalized; UI refactor for Home and Milestone retrieval initiated.
Day 2	Admin UI design complete; contracts & tasks logic discussed and assigned.
Day 3	Ongoing Projects page developed; logic for Reports page planned; token reuse confirmed.

Day 4	Quick Stats and Recent Activities UI completed; Backend began payment logic.
Day 5	Freelancer and Client payment functionality implemented; DB schema for payments finalized.
Day 6	Azure deployment scripts tested; payments stored/fetched from DB successfully.
Day 7	Reports export logic completed; UAT and code coverage tools merged into main.
Day 8	UML diagrams finalized; Admin logic for jobs and reports pushed; final QA and Azure deploy successful.

📈Sprint 4 Burndown Chart

Days	**Goal**	**Done**	**Goal velocity**	**Remaining effort**
0	0	0	115	115
1	14	12	101	103
2	14	15	87	88
3	14	10	73	78
4	14	16	59	62
5	14	13	45	49
6	14	18	31	31
7	14	11	17	20
8	17	20	0	


Burndown - Sprint 4
140
120
100
80
60
40
20
0
1
2
3
4
5
6
7
8
9
10
11
Goal velocity
Remaining