Hustlr - (Sprint 4)

✅ Sprint Backlog

Task	Owner	Status	Due Date	Story Points
Refactor UI for Home page	Front-End	Completed	12/05/25	4
Refactor UI for Milestone retrieval	Front-End	Completed	12/05/25	3
Design UI for Admin pages	Front-End	Completed	13/05/25	4
Implement Ongoing Projects Page	Team	Completed	13/05/25	5
Implement Contracts & Tasks	Team	Completed	13/05/25	6
Refactor Recent Activities Page	Front-End	Completed	13/05/25	3
Implement About Page	Team	Completed	16/05/25	2
Display chosen unique Icon in	Team	Completed	16/05/25	2

header				
Implement Reports/Quick Stats Page	Team	Completed	14/05/25	5
Implement Ongoing Projects	Team	Completed	14/05/25	5
Implement Reports/Quick Stats Page	Team	Completed	13/05/25	5
Implement Freelancer Payments functionality	Back-end	Completed	14/05/25	6
Implement Client Payments functionality	Back-end	Completed	14/05/25	6
Implement Export Functionality	Team	Completed	16/05/25	4
Backend endpoint for payments	Back-End	Completed	15/05/25	5
Store Payments in DB	Back-End	Completed	16/05/25	3
Fetch Payments in DB	Back-End	Completed	16/05/25	3
Implement Logic for “Payments”	Team	Completed	16/05/25	5
Creation of UML diagrams	Back-End	Completed	17/05/25	2
Implement Logic for Admin reports	Team	Completed	18/05/25	6
Implement Logic for Admin	Team	Completed	17/05/25	5

jobs				
UATs in correct format, code coverage tool integrated on GitHub	Back-End	Completed	16/05/25	5
60%+ Code coverage	Back-End	Completed	19/05/25	8
Deploy on Azure	Front-End	Completed	19/05/25	7
Implement Logic for milestones.	Team	Completed	16/05/25	5

📱 Sprint User Stories

User Story 1: Create Milestones as Client with a due date

Who : Client

What: Add milestone to a listing with a due date .

Why : Ensures freelancers have knowledge of tasks end dates

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 that has an assigned due date.

User Story 2: Mark a milestone as ‘in progress’ as a Freelancer 🔐

Who : Freelancer

What : Can mark a milestone as ‘in progress’ to job listings

Why : Ensures Client knows their milestone is currently being worked on

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 ‘mark as in progress’ to apply to the job

User Story 3: Mark a milestone as ‘done’ as a Freelancer 🔐

Who : Freelancer

What : Can mark a milestone as ‘done’ to job listings

Why : Ensures Client knows a milestone is done

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 ‘mark as done’ to apply to the job

---

User Story 4: Pay for a milestone as a Client 👨‍💼

Who : Client

What: Pay for a milestone that has been done

Why ： Ensures a client can pay a Freelancer

Acceptance Criteria

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

User Story 5: View reports as Admin 👨‍💼

Who : Admin

What: View all clients and freelancers on the platform

Why : Ensures an admin can see all clients and freelancers on the platform

Acceptance Criteria

- ✓ **Given:** the admin is on the reports page. **When:** view the page . **Then:** they can see all clients and freelancers on the platform.

User Story 6: View contracts and tasks as a Admin 👨‍💼

Who : Admin

What: View all clients and freelancers contracts and tasks

Why : Ensures an admin can see all clients and freelancers contracts

Acceptance Criteria

- ✓ **Given:** the admin is on the contracts and tasks page. **When:** view the page . **Then:** they can see all clients and freelancers contracts along with freelancer milestones

User Story 7: Input Profile details as a Client on Sign Up🙋‍♂️

Who : Client

What: I want to input profile details on the about page

Why : So I can store a bio, choose an icon, etc

Acceptance Criteria

- ✓**Given:** a client signs up .**When:** routed to About page. **Then:** they can input their non-personal details

User Story 8: Input Profile details as a Freelancer on Sign Up🙋‍♂️

Who : Freelancer

What: I want to input profile details on the about page

Why : So I can store a bio, choose an icon, etc

Acceptance Criteria

- ✓**Given:** a freelancer signs up .**When:** routed to About page. **Then:** they can input their non-personal details

User Story 9: View quick stats/reports as Client👨‍💼

Who : Client

What: View different metrics as a Client

Why : Ensures a Freelancer can see 4 different metrics

Acceptance Criteria

- ✓ **Given:** the Client is on the reports page. **When:** view the page . **Then:** they can see 4 various metrics

User Story 10: View quick stats/reports as Freelancer 👨‍💼

Who : Freelancer

What: View different metrics as a Freelancer

Why : Ensures a Freelancer can see 4 different metrics

Acceptance Criteria

- ✓ **Given:** the Freelancer is on the reports page. **When:** view the page . **Then:** they can see 4 various metrics

User Story 11: View contracts and tasks as a Client 👨‍💼

Who : Client

What: View all previous and active jobs

Why : Ensures a client can view their contracts

Acceptance Criteria

- ✓ **Given:** the client is on the contracts and tasks page. **When:** view the page . **Then:** they can see all their contracts

User Story 12: View contracts and tasks as a Freelancer 👨‍💼

Who : Freelancer

What: View all previous and active jobs

Why : Ensures a freelancer can view their contracts

Acceptance Criteria

- ✓ **Given:** the freelancer is on the contracts and tasks page. **When:** view the page . **Then:** they can see all their contracts

User Story 13: Export a CSV for payments as Freelancer 👨‍💼

Who : Freelancer

What: Export time specific CSV

Why : Ensures a freelancer can export a CSV for any filtered date

Acceptance Criteria

- ✓ **Given:** the freelancer is on the payments. **When:** view the page . **Then:** they can export a CSV from specific dates

User Story 14: Export a CSV for payments as Client 👨‍💼

Who : Client

What: Export time specific CSV

Why : Ensures a Client can export a CSV for any filtered date

Acceptance Criteria

- ✓ **Given:** the Client is on the payments. **When:** view the page . **Then:** they can export a CSV from specific dates

User Story 15: View ongoing projects Client 👨‍💼

Who : Client

What: View ongoing projects

Why : Ensures a client can view ongoing projects

Acceptance Criteria

- ✓ **Given:** the client is on the ongoing page. **When:** view the page . **Then:** they can see their ongoing projects

User Story 16: View ongoing projects as a Freelancer 👨‍💼

Who : Freelancer

What: View ongoing projects

Why : Ensures a freelancer can view ongoing projects

Acceptance Criteria

- ✓ **Given:** the freelancer is on the ongoing projects page. **When:** view the page . **Then:** they can see all their ongoing projects

Requirements Gathering

Jobs & Milestones (Client & Freelancer):

- What milestone details are required besides name and due date (e.g., budget, description)?
- Are milestone modifications allowed after job posting, and when?
- Can Freelancers update milestone status without formal approval?
- Should milestones support dependencies?
- Does a milestone require Client approval to be marked complete?

- Should Freelancers be able to revert milestone status changes?

Payments (Client & Freelancer):

- How are milestone payment amounts calculated?
- Should payments data be exportable, and in what format?

Profiles (Client & Freelancer):

- What *non-personal* profile information (bio, skills, portfolio, etc.) should be mandatory at sign up

Reporting & Admin Views:

- What *specific* metrics should appear in quick stats reports for Clients and Freelancers?
- What *specific* user information should Admins be able to view?
- What *specific* details about jobs, contracts, and milestones should Admins see?
- What level of control should Admins have over jobs and user accounts?
- What are the consequences and procedures when an Admin removes a 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 milestones for my jobs.	High	6	Completed
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