**Job Sequencing Visualization – Report**

**1. Problem Statement**

The Job Sequencing Problem is a classic **Greedy Algorithm** problem. The goal is to schedule jobs (or in this case, startup pitches) within given deadlines to maximize total profit. Each job has a deadline and profit associated with it.

**Objective:** Schedule the maximum profit jobs within their deadlines.

**Input:** A list of jobs with deadlines and profits, for example:

| **Startup** | **Deadline** | **Profit** |
| --- | --- | --- |
| S1 | 2 | 100 |
| S2 | 1 | 19 |
| S3 | 2 | 27 |

**Output:** Scheduled jobs and total profit.

**2. Algorithm Used**

**Greedy Algorithm – Job Sequencing:**

1. Sort all jobs in **descending order of profit**.
2. Find the maximum deadline among all jobs.
3. Create an array slot[] to track free slots.
4. Iterate through jobs:
   * For each job, check from its deadline backwards to find a free slot.
   * Assign the job to the first free slot found.
5. Sum the profits of scheduled jobs.

**Time Complexity:**

* Sorting: O(n log n)
* Scheduling: O(n \* maxDeadline)  
  **Overall Complexity:** O(n log n + n \* maxDeadline)

**Space Complexity:** O(maxDeadline + n)

**3. Visualization Implementation**

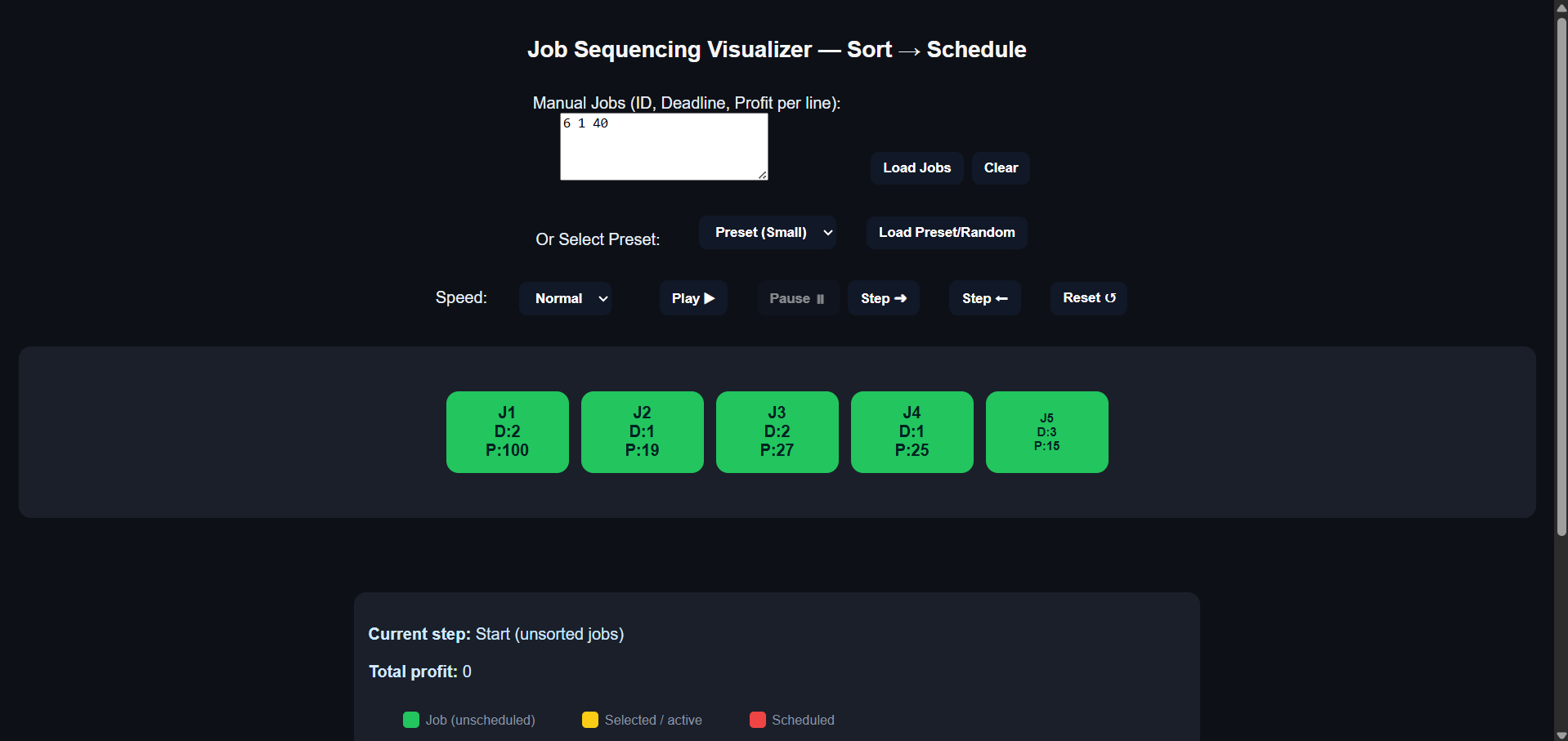
* **Languages:** HTML, CSS, JavaScript
* **Features:**
  + Stepwise visualization of the algorithm:
    - Jobs being sorted by profit
    - Slots being checked and filled
  + Controls: Play, Pause, Step Forward, Step Backward, Reset
  + Input options:
    - User-defined
    - Random jobs
    - Preset jobs (best/worst cases)
  + Display of key variables: Jobs array, Slots array, Total Profit

**Example of Visualization Steps:**

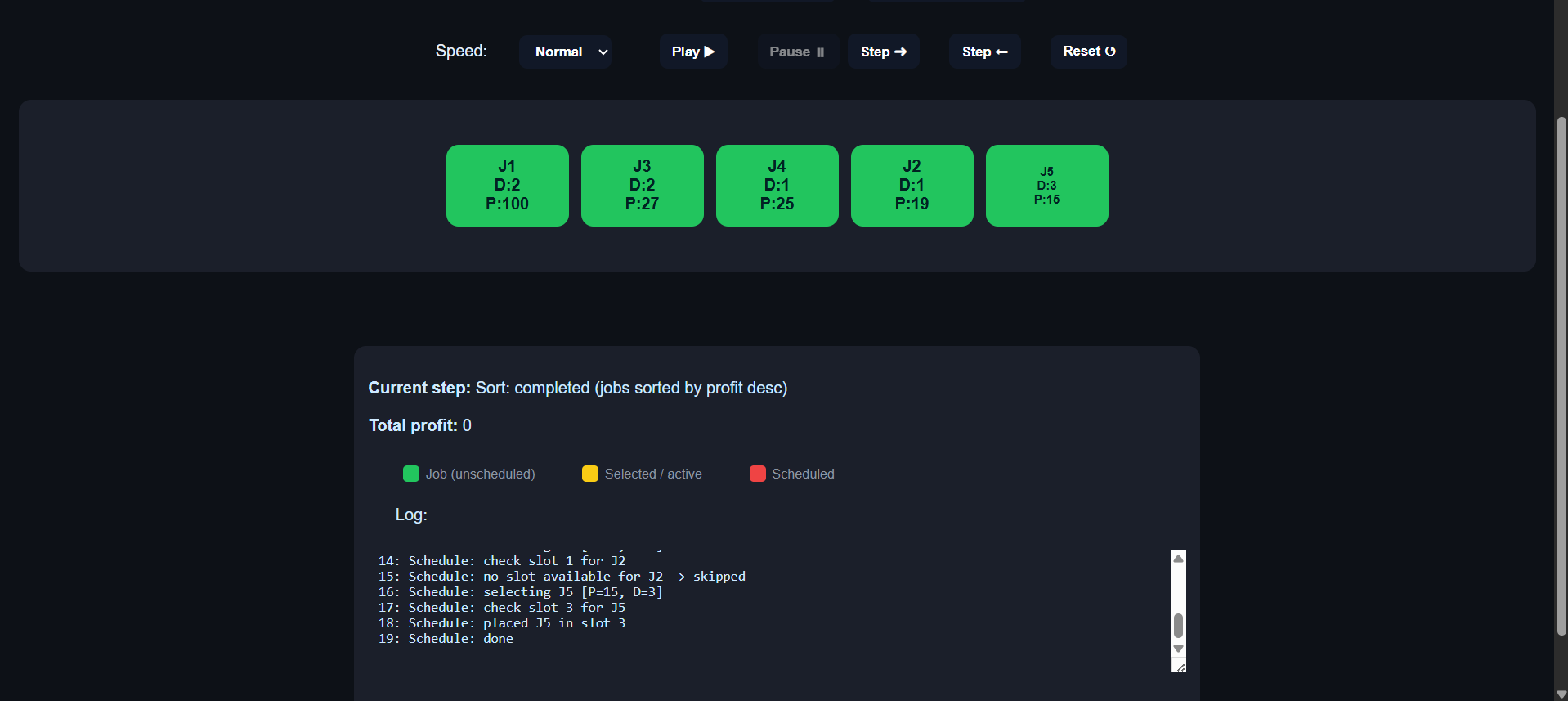
1. Jobs sorted by profit: [S1, S3, S2]
2. Slot array initialized: [-1, -1, -1]
3. Schedule S1 → Slot 2
4. Schedule S3 → Slot 1
5. S2 cannot be scheduled
6. Total Profit = 100 + 27 = 127

**4. Screenshots**

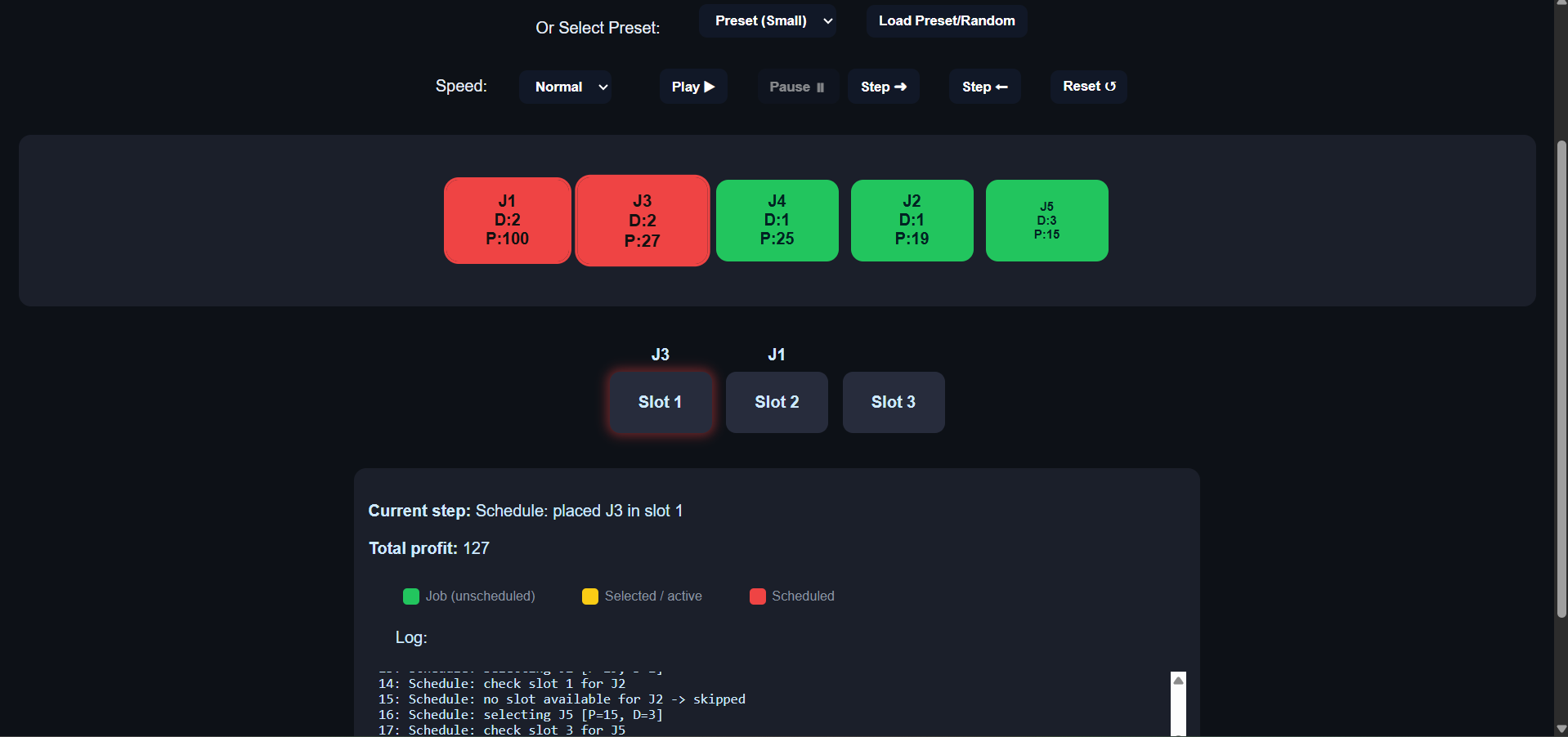
* **Step 1:** Input form for job deadlines and profits



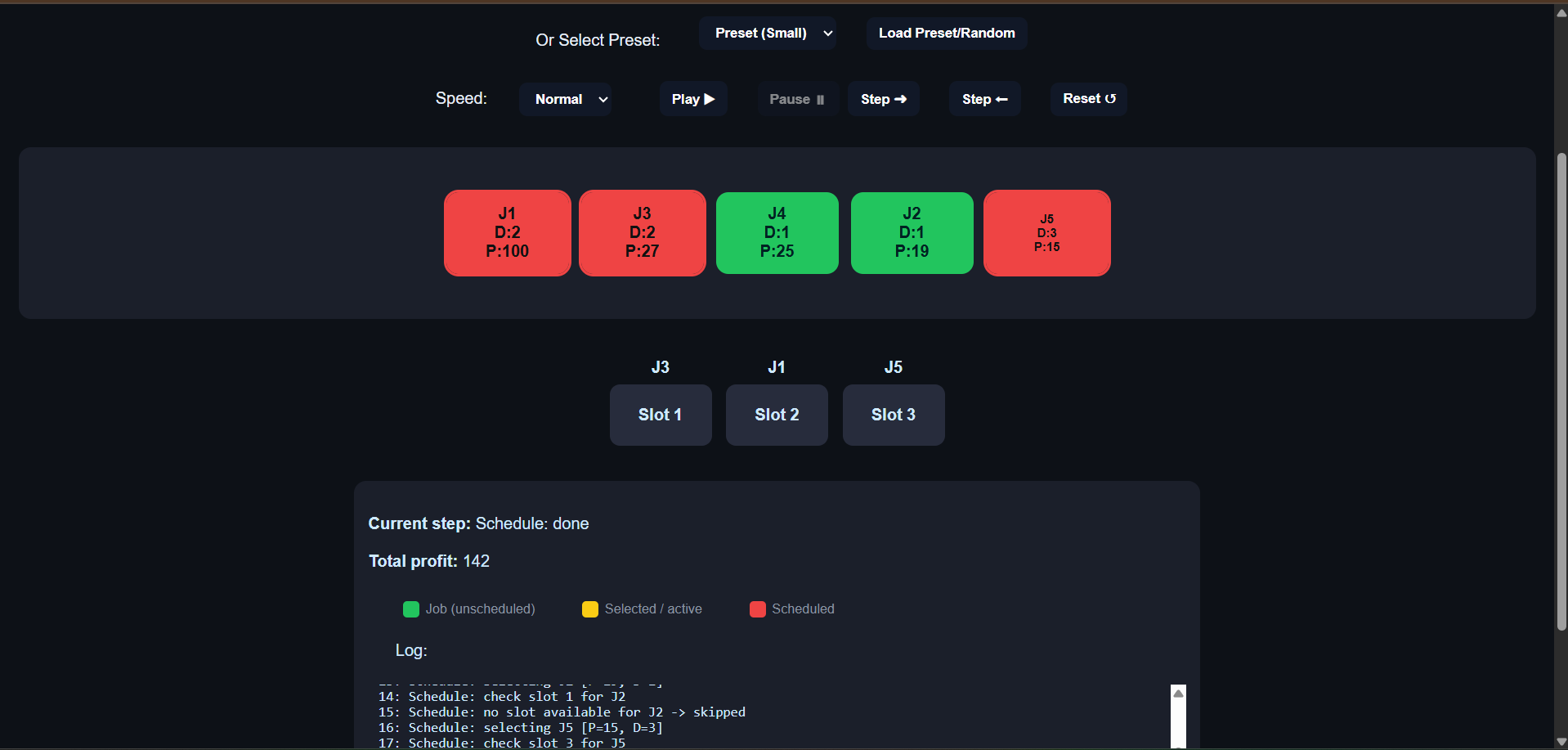
* **Step 2:** Jobs sorted by profit



* **Step 3:** Stepwise scheduling process



* **Step 4:** Final scheduled slots and total profit



**5. Run Instructions**

1. Open index.html in a browser.
2. Enter number of startups/jobs.
3. Fill in deadlines and profits or choose random/preset input.
4. Use play/pause or step buttons to visualize the scheduling.
5. Observe the scheduled jobs and total profit.

**6. Conclusion**

* Successfully implemented **Job Sequencing Algorithm** using Greedy approach.
* Stepwise visualization allows understanding of key operations (sorting, slot allocation).
* Time & space complexity analysis included.
* Interactive visualization enhances learning of algorithms.