Below, provided is the rough time estimates for simple implementation of each feature, assuming that the current backend infrastructure (NodeJS), frontend (ReactJS), and database (MongoDB) are all in good shape and dockerized using CapRover. Please note that these estimates are tentative and subject to change based on potential unforeseen complexities or challenges:

1. **Display production line name: 2 days**. This requires creating an input field, updating the database schema, and allowing for real-time updates.
2. **Product Name: 2 days**. Similar to feature 1, this also requires real-time updates.
3. **Display running (green) Line Stopped (red): 3-4 days**. This involves reading sensor data in real-time and updating the display based on predefined conditions.
4. **Shift Start Time: 2 days**. The system needs to recognize when a shift starts, based on sensor data or user input.
5. **Shift Duration timer: 2-3 days**. This involves real-time tracking of shift duration.
6. **Shift Downtime: 3-4 days**. Implementing this feature involves keeping track of sensor inactivity and calculating downtime as a percentage of total shift time.
7. **Units Per Minute: 3-4 days**. The system must track production units in real-time and calculate a rate per minute.
8. **Target Today: 2 days**. This feature requires an input field for target setting and updating the database schema.
9. **Progress Chart: 4-5 days**. This feature requires generating and updating a real-time progress chart based on current progress against target.
10. **OEE Chart: 4-5 days**. The system must calculate OEE (Overall Equipment Efficiency) parameters and display them in a dynamic chart.
11. **Line Performance Each Day: 4-5 days**. This feature involves storing daily data and generating bar charts.
12. **Client can print report/PDF: 4-5 days**. The system must compile data into a report and generate a PDF that the user can print.

Total Development Time Required: 36 - 43 days

These timeframes are rough estimates and assume an uninterrupted workflow. Testing, bug fixing, and additional adjustments could potentially extend these estimates further. Each feature will require careful design, implementation, and testing to ensure its seamless integration with the existing system. If the list of requirements increases or the specifics of these features become more complex, the timeline would also need to be adjusted accordingly.

Please keep in mind that any significant alterations to the backend might require additional time, especially if they impact the underlying database schema or the data handling processes.