

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8" />
5   <title>EngFlow- Productivity App</title>
6   <meta name="viewport" content="width=device-width, initial-scale=1.0" />
7   <style>
8     :root {
9       --bg: #020617;
10      --panel: #050816;
11      --card: #0b1120;
12      --accent: #3b82f6;
13      --text: #e5e7eb;
14      --muted: #9ca3af;
15      --border: #1f2937;
16      --radius: 10px;
17    }
18
19    *,
20    *::before,
21    *::after {
22      box-sizing: border-box;
23    }
24
25    body {
26      margin: 0;
27      min-height: 100vh;
28      font-family: system-ui, -apple-system, BlinkMacSystemFont, "Segoe UI",
29        sans-serif;
30      background: radial-gradient(
31        circle at top,
32        #111827 0,
33        #020617 40%,
34        #000 100%
35      );
36      color: var(--text);
37    }
38
39    .top-bar {
40      display: flex;
41      align-items: center;
42      justify-content: space-between;
43      padding: 0.7rem 1rem;
44      border-bottom: 1px solid var(--border);
45      background: rgba(15, 23, 42, 0.9);
46      backdrop-filter: blur(10px);
47    }
48
49    .top-bar h1 {
50      margin: 0;
51      font-size: 1rem;
52      letter-spacing: 0.08em;
53    }
54
55    .top-bar-right {
56      font-size: 0.8rem;
57      color: var(--muted);
58    }
59
60    .nav-tabs {
```

```
61     display: flex;
62     gap: 0.25rem;
63 }
64
65 .nav-tab {
66     border: none;
67     border-radius: 999px;
68     padding: 0.25rem 0.7rem;
69     font-size: 0.75rem;
70     background: transparent;
71     color: var(--muted);
72     cursor: pointer;
73 }
74
75 .nav-tab.active {
76     background: var(--accent);
77     color: white;
78 }
79
80 .main {
81     padding: 1rem;
82 }
83
84 .screen {
85     display: none;
86     background: var(--panel);
87     border-radius: var(--radius);
88     border: 1px solid var(--border);
89     padding: 0.9rem 1rem;
90 }
91
92 .active-screen {
93     display: block;
94 }
95
96 h2 {
97     margin-top: 0;
98     margin-bottom: 0.5rem;
99 }
100
101 .subtext {
102     margin: 0 0 0.7rem;
103     font-size: 0.8rem;
104     color: var(--muted);
105 }
106
107 /* Dashboard */
108 .dashboard-grid {
109     display: grid;
110     grid-template-columns: repeat(4, minmax(0, 1fr));
111     gap: 0.6rem;
112 }
113
114 .card {
115     background: var(--card);
116     border-radius: var(--radius);
117     padding: 0.6rem 0.7rem;
118     border: 1px solid var(--border);
119 }
120
```

```
121 .card h3 {
122     margin: 0;
123     font-size: 0.75rem;
124     color: var(--muted);
125     text-transform: uppercase;
126     letter-spacing: 0.12em;
127 }
128
129 .card-number {
130     margin: 0.2rem 0 0;
131     font-size: 1.3rem;
132 }
133
134 /* Subjects */
135 .subjects-list {
136     list-style: none;
137     margin: 0;
138     padding: 0;
139 }
140
141 .subject-item {
142     display: flex;
143     justify-content: space-between;
144     align-items: center;
145     padding: 0.5rem 0.6rem;
146     margin-bottom: 0.35rem;
147     border-radius: var(--radius);
148     background: rgba(15, 23, 42, 0.9);
149     border: 1px solid var(--border);
150     cursor: pointer;
151 }
152
153 .subject-item:hover {
154     border-color: var(--accent);
155 }
156
157 .subject-main {
158     display: flex;
159     flex-direction: column;
160 }
161
162 .subject-name {
163     font-size: 0.95rem;
164 }
165
166 .subject-extra {
167     font-size: 0.75rem;
168     color: var(--muted);
169 }
170
171 .subject-stats {
172     text-align: right;
173     font-size: 0.75rem;
174     color: var(--muted);
175 }
176
177 /* Subject Detail */
178 .subject-header {
179     display: flex;
180     justify-content: space-between;
```

```
181     align-items: center;
182     margin-bottom: 1rem;
183 }
184
185 .subject-meta {
186     font-size: 0.8rem;
187     color: var(--muted);
188 }
189
190 .summary-grid {
191     display: grid;
192     grid-template-columns: repeat(4, minmax(0, 1fr));
193     gap: 0.5rem;
194     margin-bottom: 1rem;
195 }
196
197 .summary-card {
198     background: var(--card);
199     border-radius: var(--radius);
200     border: 1px solid var(--border);
201     padding: 0.4rem 0.5rem;
202     text-align: center;
203 }
204
205 .summary-label {
206     display: block;
207     font-size: 0.7rem;
208     color: var(--muted);
209 }
210
211 .summary-value {
212     font-size: 1rem;
213     font-weight: 500;
214 }
215
216 /* Task lists */
217 .task-list,
218 .simple-task-list {
219     list-style: none;
220     margin: 0;
221     padding: 0;
222     border-radius: var(--radius);
223     border: 1px solid var(--border);
224     overflow: hidden;
225 }
226
227 .task-item {
228     display: flex;
229     justify-content: space-between;
230     align-items: center;
231     padding: 0.45rem 0.6rem;
232     font-size: 0.85rem;
233     border-bottom: 1px solid var(--border);
234 }
235
236 .task-item:last-child {
237     border-bottom: none;
238 }
239
240 .task-title {
```

```
241     font-weight: 500;
242 }
243
244 .task-meta {
245     font-size: 0.75rem;
246     color: var(--muted);
247 }
248
249 .chip {
250     display: inline-block;
251     padding: 0.1rem 0.45rem;
252     border-radius: 999px;
253     font-size: 0.7rem;
254     border: 1px solid var(--border);
255     margin-right: 0.3rem;
256 }
257
258 /* Buttons */
259 .primary-btn,
260 .secondary-btn {
261     border: none;
262     cursor: pointer;
263     font: inherit;
264 }
265
266 .primary-btn {
267     padding: 0.35rem 0.8rem;
268     border-radius: 999px;
269     background: var(--accent);
270     color: white;
271     font-size: 0.8rem;
272 }
273
274 .primary-btn.small {
275     padding: 0.25rem 0.6rem;
276     font-size: 0.75rem;
277 }
278
279 .secondary-btn {
280     padding: 0.35rem 0.8rem;
281     border-radius: 999px;
282     background: transparent;
283     color: var(--muted);
284     border: 1px solid var(--border);
285     font-size: 0.8rem;
286 }
287
288 /* Form */
289 .task-form {
290     max-width: 360px;
291 }
292
293 .form-row {
294     display: flex;
295     flex-direction: column;
296     margin-bottom: 0.5rem;
297     font-size: 0.8rem;
298 }
299
300 .form-row label {
```

```
301     margin-bottom: 0.15rem;
302     color: var(--muted);
303 }
304
305 .form-row input,
306 .form-row select {
307     padding: 0.35rem 0.4rem;
308     border-radius: 0.4rem;
309     border: 1px solid var(--border);
310     background: rgba(15, 23, 42, 0.9);
311     color: var(--text);
312     font: inherit;
313 }
314
315 .form-actions {
316     margin-top: 0.6rem;
317     display: flex;
318     gap: 0.4rem;
319 }
320
321 .section-title {
322     margin-top: 1rem;
323     margin-bottom: 0.4rem;
324     font-size: 0.9rem;
325 }
326
327 @media (max-width: 768px) {
328     .dashboard-grid,
329     .summary-grid {
330         grid-template-columns: repeat(2, minmax(0, 1fr));
331     }
332
333     .subject-header {
334         flex-direction: column;
335         align-items: flex-start;
336         gap: 0.5rem;
337     }
338
339     .subject-item {
340         flex-direction: column;
341         align-items: flex-start;
342     }
343
344     .subject-stats {
345         margin-top: 0.25rem;
346         text-align: left;
347     }
348 }
349 </style>
350 </head>
351 <body>
352     <header class="top-bar">
353         <h1>EngFlow</h1>
354         <nav class="nav-tabs">
355             <button class="nav-tab active" data-screen="dashboard-
screen">Dashboard</button>
356             <button class="nav-tab" data-screen="subjects-screen">Subjects</button>
357             <button class="nav-tab" data-screen="subject-detail-screen">Subject
Detail</button>
358             <button class="nav-tab" data-screen="add-task-screen">Add Task</button>
```

```
359     </nav>
360     <div class="top-bar-right">
361         <span id="today-date"></span>
362     </div>
363 </header>
364
365 <main class="main">
366     <!-- Dashboard -->
367     <section id="dashboard-screen" class="screen active-screen">
368         <h2>Dashboard</h2>
369         <div class="dashboard-grid">
370             <div class="card">
371                 <h3>Total Tasks</h3>
372                 <p id="dash-total" class="card-number">0</p>
373             </div>
374             <div class="card">
375                 <h3>Due Today</h3>
376                 <p id="dash-today" class="card-number">0</p>
377             </div>
378             <div class="card">
379                 <h3>Next 7 Days</h3>
380                 <p id="dash-week" class="card-number">0</p>
381             </div>
382             <div class="card">
383                 <h3>Completion</h3>
384                 <p id="dash-completion" class="card-number">0%</p>
385             </div>
386         </div>
387         <h3 class="section-title">Upcoming Tasks</h3>
388         <ul id="dashboard-task-list" class="task-list"></ul>
389     </section>
390
391     <!-- Subjects -->
392     <section id="subjects-screen" class="screen">
393         <h2>Subjects</h2>
394         <p class="subtext">Select a subject to view its tasks and progress.</p>
395         <ul id="subject-list" class="subjects-list"></ul>
396     </section>
397
398     <!-- Subject Detail -->
399     <section id="subject-detail-screen" class="screen">
400         <header class="subject-header">
401             <div>
402                 <h2 id="subject-detail-title">Engineering Physics</h2>
403                 <p id="subject-detail-meta" class="subject-meta"></p>
404             </div>
405             <button id="subject-add-task-btn" class="primary-btn small">
406                 + Add Task
407             </button>
408         </header>
409
410         <div class="summary-grid">
411             <div class="summary-card">
412                 <span class="summary-label">Total</span>
413                 <span id="sub-total" class="summary-value">0</span>
414             </div>
415             <div class="summary-card">
416                 <span class="summary-label">Pending</span>
417                 <span id="sub-pending" class="summary-value">0</span>
418             </div>
419         </div>
420     </section>
421 </main>
422 </body>
423 </html>
```

```
419     <div class="summary-card">
420         <span class="summary-label">Done</span>
421         <span id="sub-done" class="summary-value">0</span>
422     </div>
423     <div class="summary-card">
424         <span class="summary-label">Completion</span>
425         <span id="sub-completion" class="summary-value">0%</span>
426     </div>
427 </div>
428
429 <h3 class="section-title">Tasks for this subject</h3>
430 <ul id="subject-task-list" class="task-list"></ul>
431 </section>
432
433 <!-- Add Task -->
434 <section id="add-task-screen" class="screen">
435     <h2 id="add-task-title">Add Task</h2>
436     <p id="add-task-context" class="subtext"></p>
437
438     <form id="task-form" class="task-form">
439         <div class="form-row">
440             <label for="task-title">Title</label>
441             <input id="task-title" type="text" required />
442         </div>
443
444         <div class="form-row">
445             <label for="task-type">Type</label>
446             <select id="task-type" required>
447                 <option value="assignment">Assignment</option>
448                 <option value="lab">Lab</option>
449                 <option value="exam">Exam</option>
450                 <option value="project">Project</option>
451             </select>
452         </div>
453
454         <div class="form-row">
455             <label for="task-subject">Subject</label>
456             <input
457                 id="task-subject"
458                 type="text"
459                 placeholder="e.g. Engineering Physics"
460                 required
461             />
462         </div>
463
464         <div class="form-row">
465             <label for="task-due">Due Date</label>
466             <input id="task-due" type="date" required />
467         </div>
468
469         <div class="form-row">
470             <label for="task-status">Status</label>
471             <select id="task-status" required>
472                 <option value="pending">Pending</option>
473                 <option value="in-progress">In Progress</option>
474                 <option value="done">Done</option>
475             </select>
476         </div>
477
478         <div class="form-actions">
```



```
479         <button type="button" id="cancel-add" class="secondary-btn">
480             Cancel
481         </button>
482         <button type="submit" class="primary-btn">
483             Save Task
484         </button>
485     </div>
486 </form>
487 </section>
488 </main>
489
490 <script>
491     // Mock data (frontend-only)
492     const tasks = [
493         {
494             id: 1,
495             title: "Physics Assignment 1",
496             type: "assignment",
497             subject: "Engineering Physics",
498             due: "2026-01-20",
499             status: "pending"
500         },
501         {
502             id: 2,
503             title: "C Lab 3",
504             type: "lab",
505             subject: "Programming in C",
506             due: "2026-01-18",
507             status: "in-progress"
508         },
509         {
510             id: 3,
511             title: "Midterm Exam",
512             type: "exam",
513             subject: "Engineering Mathematics",
514             due: "2026-01-25",
515             status: "pending"
516         },
517         {
518             id: 4,
519             title: "Mini Project Proposal",
520             type: "project",
521             subject: "Basic Electronics Engineering",
522             due: "2026-01-22",
523             status: "done"
524         },
525         {
526             id: 5,
527             title: "Physics Assignment 2",
528             type: "assignment",
529             subject: "Engineering Physics",
530             due: "2026-01-28",
531             status: "done"
532         }
533     ];
534
535     let nextId = 6;
536     let currentSubject = "Engineering Physics";
537
538     // Elements
```

```
539 const todayDateEl = document.getElementById("today-date");
540 const navTabs = document.querySelectorAll(".nav-tab");
541 const screens = document.querySelectorAll(".screen");
542
543 const dashTotalEl = document.getElementById("dash-total");
544 const dashTodayEl = document.getElementById("dash-today");
545 const dashWeekEl = document.getElementById("dash-week");
546 const dashCompletionEl = document.getElementById("dash-completion");
547 const dashTaskListEl = document.getElementById("dashboard-task-list");
548
549 const subjectListEl = document.getElementById("subject-list");
550
551 const subjectDetailTitleEl = document.getElementById("subject-detail-title");
552 const subjectDetailMetaEl = document.getElementById("subject-detail-meta");
553 const subjectTaskListEl = document.getElementById("subject-task-list");
554 const subTotalEl = document.getElementById("sub-total");
555 const subPendingEl = document.getElementById("sub-pending");
556 const subDoneEl = document.getElementById("sub-done");
557 const subCompletionEl = document.getElementById("sub-completion");
558 const subjectAddTaskBtn = document.getElementById("subject-add-task-btn");
559
560 const addTaskTitleEl = document.getElementById("add-task-title");
561 const addTaskContextEl = document.getElementById("add-task-context");
562 const taskForm = document.getElementById("task-form");
563 const titleInput = document.getElementById("task-title");
564 const typeInput = document.getElementById("task-type");
565 const subjectInput = document.getElementById("task-subject");
566 const dueInput = document.getElementById("task-due");
567 const statusInput = document.getElementById("task-status");
568 const cancelAddBtn = document.getElementById("cancel-add");
569
570 // Utils
571 function switchScreen(id) {
572   screens.forEach((s) => s.classList.remove("active-screen"));
573   document.getElementById(id).classList.add("active-screen");
574
575   navTabs.forEach((btn) => {
576     btn.classList.toggle("active", btn.dataset.screen === id);
577   });
578 }
579
580 function formatDateShort(dateStr) {
581   const d = new Date(dateStr);
582   return d.toLocaleDateString("en-IN", {
583     day: "2-digit",
584     month: "short"
585   });
586 }
587
588 function getTodayISO() {
589   return new Date().toISOString().slice(0, 10);
590 }
591
592 // Dashboard
593 function renderDashboard() {
594   const total = tasks.length;
595   dashTotalEl.textContent = total;
596
597   const todayISO = getTodayISO();
598   const todayCount = tasks.filter((t) => t.due === todayISO).length;
```

```
599 dashTodayEl.textContent = todayCount;
600
601 const today = new Date(todayISO);
602 const weekAhead = new Date(today);
603 weekAhead.setDate(today.getDate() + 7);
604 const weekCount = tasks.filter((t) => {
605     const d = new Date(t.due);
606     return d > today && d <= weekAhead;
607 }).length;
608 dashWeekEl.textContent = weekCount;
609
610 const doneCount = tasks.filter((t) => t.status === "done").length;
611 const completion = total === 0 ? 0 : Math.round((doneCount / total) * 100);
612 dashCompletionEl.textContent = `${completion}%`;
613
614 dashTaskListEl.innerHTML = "";
615 const upcomingSorted = [...tasks].sort((a, b) =>
616     a.due.localeCompare(b.due)
617 );
618
619 upcomingSorted.forEach((task) => {
620     const li = document.createElement("li");
621     li.className = "task-item";
622
623     const left = document.createElement("div");
624     left.textContent = task.title;
625
626     const right = document.createElement("div");
627     right.className = "task-meta";
628     right.textContent = `${task.subject} • ${formatDateShort(task.due)}`;
629
630     li.appendChild(left);
631     li.appendChild(right);
632     dashTaskListEl.appendChild(li);
633 });
634 }
635
636 // Subjects
637 function renderSubjects() {
638     const subjects = [...new Set(tasks.map((t) => t.subject))];
639     subjectListEl.innerHTML = "";
640
641     subjects.forEach((subj) => {
642         const subjTasks = tasks.filter((t) => t.subject === subj);
643         const total = subjTasks.length;
644         const done = subjTasks.filter((t) => t.status === "done").length;
645         const pending = subjTasks.filter((t) => t.status !== "done").length;
646         const completion = total === 0 ? 0 : Math.round((done / total) * 100);
647
648         const li = document.createElement("li");
649         li.className = "subject-item";
650
651         const mainDiv = document.createElement("div");
652         mainDiv.className = "subject-main";
653
654         const nameDiv = document.createElement("div");
655         nameDiv.className = "subject-name";
656         nameDiv.textContent = subj;
657
658         const extraDiv = document.createElement("div");
```

```
659     extraDiv.className = "subject-extra";
660     extraDiv.textContent = `${total} tasks • ${pending} pending • ${done} done`;
661
662     mainDiv.appendChild(nameDiv);
663     mainDiv.appendChild(extraDiv);
664
665     const statsDiv = document.createElement("div");
666     statsDiv.className = "subject-stats";
667     statsDiv.textContent = `${completion}% done`;
668
669     li.appendChild(mainDiv);
670     li.appendChild(statsDiv);
671
672     li.addEventListener("click", () => {
673         currentSubject = subj;
674         switchScreen("subject-detail-screen");
675         renderSubjectDetail();
676     });
677
678     subjectListEl.appendChild(li);
679 });
680 }
681
682 // Subject Detail
683 function renderSubjectDetail() {
684     if (!currentSubject) return;
685
686     const subjTasks = tasks.filter((t) => t.subject === currentSubject);
687     const total = subjTasks.length;
688     const pending = subjTasks.filter((t) => t.status !== "done").length;
689     const done = subjTasks.filter((t) => t.status === "done").length;
690     const completion = total === 0 ? 0 : Math.round((done / total) * 100);
691
692     subjectDetailTitleEl.textContent = currentSubject;
693     subjectDetailMetaEl.textContent = `${total} tasks • ${completion}% done`;
694
695     subTotalEl.textContent = total;
696     subPendingEl.textContent = pending;
697     subDoneEl.textContent = done;
698     subCompletionEl.textContent = `${completion}%`;
699
700     subjectTaskListEl.innerHTML = "";
701     subjTasks
702     .sort((a, b) => a.due.localeCompare(b.due))
703     .forEach((task) => {
704         const li = document.createElement("li");
705         li.className = "task-item";
706
707         const left = document.createElement("div");
708         const titleDiv = document.createElement("div");
709         titleDiv.className = "task-title";
710         titleDiv.textContent = task.title;
711
712         const typeChip = document.createElement("span");
713         typeChip.className = "chip";
714         typeChip.textContent = task.type;
715         left.appendChild(titleDiv);
716         left.appendChild(typeChip);
717
718         const right = document.createElement("div");
```

```
719     right.className = "task-meta";
720     right.textContent = `${formatDateShort(task.due)} • ${task.status}`;
721
722     li.appendChild(left);
723     li.appendChild(right);
724
725     li.addEventListener("click", () => {
726       alert("Edit task: " + task.title);
727     });
728
729     subjectTaskListEl.appendChild(li);
730   });
731 }
732
733 // Add Task
734 function openAddTask(subjectPref = null) {
735   switchScreen("add-task-screen");
736
737   addTaskTitleEl.textContent = "Add Task";
738   if (subjectPref) {
739     addTaskContextEl.textContent = `New task for ${subjectPref}`;
740   } else {
741     addTaskContextEl.textContent = "";
742   }
743
744   taskForm.reset();
745   dueInput.value = getTodayISO();
746   subjectInput.value = subjectPref || "";
747 }
748
749 taskForm.addEventListener("submit", (e) => {
750   e.preventDefault();
751
752   const data = {
753     title: titleInput.value.trim(),
754     type: typeInput.value,
755     subject: subjectInput.value.trim(),
756     due: dueInput.value,
757     status: statusInput.value
758   };
759
760   if (!data.title || !data.subject || !data.due) return;
761
762   tasks.push({
763     id: nextId++,
764     ...data
765   });
766
767   renderDashboard();
768   renderSubjects();
769   if (currentSubject && currentSubject === data.subject) {
770     renderSubjectDetail();
771   }
772
773   if (currentSubject && currentSubject === data.subject) {
774     switchScreen("subject-detail-screen");
775   } else {
776     switchScreen("dashboard-screen");
777   }
778 });
```

```
779
780 cancelAddBtn.addEventListener("click", () => {
781     switchScreen("dashboard-screen");
782 });
783
784 // Navigation
785 navTabs.forEach(btn => {
786     btn.addEventListener("click", () => {
787         const target = btn.dataset.screen;
788         switchScreen(target);
789         if (target === "dashboard-screen") renderDashboard();
790         if (target === "subjects-screen") renderSubjects();
791         if (target === "subject-detail-screen") renderSubjectDetail();
792         if (target === "add-task-screen") openAddTask();
793     });
794 });
795
796 subjectAddTaskBtn.addEventListener("click", () => {
797     if (!currentSubject) return;
798     openAddTask(currentSubject);
799 });
800
801 // Init
802 function init() {
803     const today = new Date();
804     todayDateEl.textContent = today.toLocaleDateString("en-IN", {
805         weekday: "short",
806         day: "2-digit",
807         month: "short",
808         year: "numeric"
809     });
810
811     dueInput.value = getTodayISO();
812     renderDashboard();
813     renderSubjects();
814     renderSubjectDetail(); // Set default subject
815 }
816
817 init();
818 </script>
819 </body>
820 </html>
821
```