**תקציר במבט־על**

אתה מקים ריפו  GitHub, יוצר לוח Projects לניהול משימות, בונה שלד  React ו־Node, עוטף ב‑Docker Compose, מוסיף צינור CI/CD ב‑GitHub Actions, מחבר PostgreSQL, מגבה JWT‑Auth, ומפרסם ב‑Railway. כל ממונח ופעולה מלווים במדריך רשמי, כך שלא דרוש ידע קודם.

**מילון מונחים בסיסי**

| **מונח** | **הסבר** | **מדריך** |
| --- | --- | --- |
| **Repository (ריפו)** | תיקיית‑קוד בענן | [git-scm.com](https://git-scm.com/book/en/v2/Getting-Started-Installing-Git?utm_source=chatgpt.com) |
| **Commit / Branch / PR** | צילום‑קוד, קו‑פיתוח, בקשת מיזוג | [git-scm.com](https://git-scm.com/book/en/v2/Getting-Started-Installing-Git?utm_source=chatgpt.com) |
| **Issue / Epic / Story / Task** | כרטיסי עבודה בהיררכיה | [אטלסיאן](https://www.atlassian.com/work-management/project-management/acceptance-criteria?utm_source=chatgpt.com) |
| **Acceptance Criteria** | תנאי‑גמר ברורים לכל Story | [אטלסיאן](https://www.atlassian.com/work-management/project-management/acceptance-criteria?utm_source=chatgpt.com) |
| **Story Points** | מדד מאמץ (1‑8) | [GitHub Docs](https://docs.github.com/en/issues/planning-and-tracking-with-projects/learning-about-projects/quickstart-for-projects?utm_source=chatgpt.com) |
| **Conventional Commits** | פורמט קבוע להודעות Commit | [conventionalcommits.org](https://www.conventionalcommits.org/en/v1.0.0/?utm_source=chatgpt.com) |
| **Time‑Blocking** | חסימת‑זמנים ביומן | [WIRED](https://www.wired.com/story/block-scheduling-calendar-workflow-productivity?utm_source=chatgpt.com) |
| **Pomodoro** | 25 דק עבודה + 5 דק הפסקה | [Reddit](https://www.reddit.com/r/learnprogramming/comments/vcmpqu/using_pomodoro_as_a_developer/?utm_source=chatgpt.com) |

**שלב 0 — התקנות חד‑פעמיות (יום 0, 17:00–18:30)**

1. התקנת **Git** ‏→ בדוק git --version. [git-scm.com](https://git-scm.com/book/en/v2/Getting-Started-Installing-Git?utm_source=chatgpt.com)
2. התקנת **Node LTS** ‏→ בדוק node -v. [Node.js — Run JavaScript Everywhere](https://nodejs.org/en/download?utm_source=chatgpt.com)
3. התקנת **VS Code** + תוספי ESLint / Prettier

**שגרת‑ערב מומלצת**

| **זמן** | **פעולה** | **מדריך** |
| --- | --- | --- |
| 17:00‑17:05 | Stand‑up עצמי: לגרור Story ל‑In Progress | [GitHub Docs](https://docs.github.com/en/issues/planning-and-tracking-with-projects/learning-about-projects/quickstart-for-projects?utm_source=chatgpt.com) |
| 17:05‑18:35 | Deep‑Work #1 (3 פומודורו) | [Reddit](https://www.reddit.com/r/learnprogramming/comments/vcmpqu/using_pomodoro_as_a_developer/?utm_source=chatgpt.com) |
| 18:35‑19:00 | הפסקה | — |
| 19:00‑19:30 | למידה: קרא את מדריך השלב | (קישור בטבלה הבאה) |
| 19:30‑21:00 | Deep‑Work #2 | — |
| 21:00‑21:30 | PR + Docs (Commit בפורמט מוסכם) | [conventionalcommits.org](https://www.conventionalcommits.org/en/v1.0.0/?utm_source=chatgpt.com) |

**מפת‑דרכים (12 שלבים ≈ 6 שבועות)**

| **#** | **ערבים** | **תוצאה** | **מדריך** |
| --- | --- | --- | --- |
| **1. ריפו + Kanban** | 1 | Repo + לוח Projects | [GitHub Docs](https://docs.github.com/en/issues/planning-and-tracking-with-projects/learning-about-projects/quickstart-for-projects?utm_source=chatgpt.com) |
| **2. Epic + Stories** | ½ | Epic “MVP” + 2 Stories | [אטלסיאן](https://www.atlassian.com/work-management/project-management/acceptance-criteria?utm_source=chatgpt.com) |
| **3. שלד React** | 2 | create-react-app client | [React](https://react.dev/learn/creating-a-react-app?utm_source=chatgpt.com) |
| **4. API Express** | 2 | /api/health שרץ | [expressjs.com](https://expressjs.com/en/starter/hello-world.html?utm_source=chatgpt.com) |
| **5. חיבור Proxy** | 1 | React ↔ Express | (מדריך Express) |
| **6. Docker Compose** | 1 | Dockerfile ×2 + docker-compose.yml + Postgres | [Docker Documentation](https://docs.docker.com/compose/gettingstarted/?utm_source=chatgpt.com) |
| **7. GitHub Actions** | 2 | צינור Test → Build | [GitHub Docs](https://docs.github.com/en/actions/writing-workflows/quickstart?utm_source=chatgpt.com) |
| **8. PostgreSQL** | 2 | טבלת users + Migration | [Neon](https://www.postgresqltutorial.com/?utm_source=chatgpt.com) |
| **9. JWT Auth + CRUD** | 6 | מסלולי /auth/\*, /tasks | (Express + JWT docs) |
| **10. Deploy Railway** | 2 | גרסה חיה | [docs.railway.app](https://docs.railway.app/deploy/dockerfiles?utm_source=chatgpt.com) |
| **11. Grafana Loki** | 1 | לוגים בענן | (Grafana Cloud guide) |
| **12. MkDocs Site** | 1 | אתר תיעוד GitHub Pages |  |

**זרימת עבודה בלוח GitHub**

1. **Backlog** – כל Story חדשה.
2. **In Progress** – Story פעילה.
3. **Code Review** – PR פתוח; ‏CI חייב להצליח. [GitHub Docs](https://docs.github.com/en/actions/writing-workflows/quickstart?utm_source=chatgpt.com)
4. **Done** – PR מוזג, Issue נסגר.

**הערב הראשון (היום 17:00)**

1. השלם התקנות Git ו‑Node. [git-scm.com](https://git-scm.com/book/en/v2/Getting-Started-Installing-Git?utm_source=chatgpt.com)[Node.js — Run JavaScript Everywhere](https://nodejs.org/en/download?utm_source=chatgpt.com)
2. צור ריפו task‑manager‑fullstack + README.
3. פתח לוח Projects והוסף עמודות. [GitHub Docs](https://docs.github.com/en/issues/planning-and-tracking-with-projects/learning-about-projects/quickstart-for-projects?utm_source=chatgpt.com)
4. צור Epic “MVP” + Stories (“register/login”, “create task”) עם Acceptance Criteria. [אטלסיאן](https://www.atlassian.com/work-management/project-management/acceptance-criteria?utm_source=chatgpt.com)
5. חסום את בלוקי ה‑Time‑Blocking ביומן לפי המדריך. [Verywell Mind](https://www.verywellmind.com/how-to-use-time-blocking-to-manage-your-day-4797509?utm_source=chatgpt.com)

**רק אחרי** שהצעדים האלה נעשו – עבור לשלב React (שלב 3).

**יש לך עכשיו**

* רשימת מושגים עם מדריכי־מקור.
* שגרת‑ערב ברורה וטבלה מדורגת ל‑12 שלבים.
* קישורים רשמיים לכל כלי/טכנולוגיה.

צעד‑אחר‑צעד, בלי דילוגים, תגיע לאפליקציה חיה, מתועדת ומוכנה להראות למעסיק הבא. בהצלחה!

| **יום** |  |  |  |  | **בלוק 17:00‑18:35** |  |  |  | **למידה 19:00‑19:30** | **בלוק 19:30‑21:00** | **21:00‑21:30** | **תוצר יומי** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **א׳** |  |  |  |  | *Setup* – התקנות Git + Node, פתיחת ריפו, יצירת **Project Board** (עמודות + Story Points). |  |  |  | מדריך “GitHub Projects” | יצירת **Epic MVP** + 2 Stories (“Register/Login”, “Create Task”). | Commit ״chore: project init״ + PR ל־main | Repo & Board מוכנים; Epic & Stories בלוח Backlog. |
| **ב׳** |  |  |  |  | שלד **React** (npx create-react-app client). |  |  |  | מדריך Create‑React‑App | חדירת קבצי בסיס (Router, דף Login, CSS reset). | Commit ״feat(client): bootstrap React app״ | Front‑end רץ ב‑localhost 3000. |
| **ג׳** |  |  |  |  | שלד **Express API** (server/index.js + GET /api/health). |  |  |  | מדריך Express Hello World | קביעת proxy ב‑client; בדיקת fetch /api/health. | PR ״feat(api): basic health route״ | Back‑end מאזין ב‑3001; React מתקשר. |
| **ד׳** |  |  |  |  | **User Registration** – טבלת users, ספריית bcrypt. |  |  |  | מדריך PostgreSQL + bcrypt | Route POST /auth/register, test Jest יחיד. | PR ״feat(api): register user״ | משתמש חדש נוצר במסד. |
| **ה׳** |  |  |  |  | **User Login** – JWT issue (jsonwebtoken). |  |  |  | JWT quick start | Route POST /auth/login; שמירת token ב‑localStorage. | PR ״feat(api): login & token storage״ | התחברות עובדת, token מתקבל. |
| **ו׳** |  |  |  |  | **Create Task** – Route POST /tasks (Auth middleware). |  |  |  | מדריך Express middleware | טבלה tasks, Form React “Add Task”, עדכון state. | PR ״feat(tasks): create task CRUD base״ | משימות נוספות בבסיס‑נתונים. |
| **שבת** |  |  |  |  | בדיקות E2E (Playwright) – רישום→כניסה→יצירת משימה. |  |  |  | מדריך Playwright intro | Dockerfile + Compose (client, server, db). | **Retro + Review**: Story Points שנסגרו, Burn‑down, תכנון Sprint הבא. | גרסה מקומית מלאה רצה עם Docker; Stories “Register/Login” ו‑“Create Task” == Done. |

* **Acceptance Criteria** לכל Story (פורמט Given/When/Then) נכתבים בתיאור ה‑Issue ומתוחזקים.
* בכל Commit השתמש בפורמט Conventional Commits (feat|chore|test:).
* קבצי Docker ו‑Compose נכנסים לענף feature/devops-docker; מוזגים רק כשהבדיקות ירוקות.
* בסוף כל ערב:  
  \* עדכן Story Points “Done” בשדה בלוח.  
  \* פתח PR אם סיימת משימה; אם לא – השאר Todo‑list בתיאור ה‑PR להמשך מחר.

**קישורים מהירים (ללמוד בחלון 19:00‑19:30)**

1. GitHub Projects Board → <https://docs.github.com/en/issues/planning-and-tracking-with-projects/quickstart>
2. Create React App → https://react.dev/learn/start-a-new-react-project
3. Express Hello World → https://expressjs.com/en/starter/hello-world.html
4. PostgreSQL Getting Started → https://www.postgresqltutorial.com/postgresql-getting-started/
5. bcrypt npm → <https://www.npmjs.com/package/bcrypt>
6. JSON Web Token → https://jwt.io/introduction
7. Docker Compose Quick‑Start → https://docs.docker.com/compose/gettingstarted/
8. Playwright Test → https://playwright.dev/docs/intro

כך תסיים את השבוע עם MVP עובד (רישום, כניסה, יצירת משימה) ואפיק MVP סגור, מוכנים להתחיל Sprint חדש. בהצלחה!

| **צעד** | **פקודה או פעולה** | **מה בדיוק מתרחש** |
| --- | --- | --- |
| **1. יצירת ענף חדש** | bash\ngit checkout -b feature/react-skeleton\n | • Git יוצר עותק עבודה נפרד בשם feature/react-skeleton. • אתה כבר לא על main, ולכן אינך פוגע בקוד היציב. |
| **2. כתיבת קוד ו‑Commit** | bash\ngit add .\ngit commit -m "feat(client): bootstrap React"\n | • git add . מכניס את כל הקבצים ששינית ל‑“תמונה” Staging. • git commit … יוצר צילום קוד קבוע עם הודעה שמתארת את השינוי. |
| **3. Push לענן (GitHub)** | bash\ngit push -u origin feature/react-skeleton\n | • השינויים עולים ל‑GitHub בענף החדֵש. • הדגל -u קובע מעקב (ל־push בעתיד מספיק git push). |
| **4. פתיחת Pull Request (PR)** | GitHub מראה כפתור **Compare & pull request** | • לחץ על הכפתור. • מלא **כותרת** (למשל feat: React skeleton), ו‑**תיאור** קצר על מה שנעשה. |
| **5. קישור ל‑Story בלוח** | בלשונית **Linked issues** ב‑PR | • בחר את הכרטיס US: React Skeleton. • הכרטיס עובר אוטומטית לעמודת **Code Review** בלוח Projects. |
| **6. בדיקות CI** | GitHub Actions רץ לבד | • אם הלוגים ירוקים – אפשר למזג. • אם אדום – תקן קוד → git add → git commit → git push (ה‑PR מתעדכן). |
| **7. מיזוג (Merge)** | כפתור **Merge pull request** → **Confirm** | • הקוד של הענף מתווסף ל‑main. • אם בכותרת ה‑PR כתבת Closes #3, ה‑Issue #3 (ה‑Story) נסגר אוטומטית ועובר לעמודת **Done**. |
| **8. מחיקת הענף** | כפתור **Delete branch** (מופיע אחרי המיזוג) | • מנקה את רשימת הענפים ב‑GitHub. • אין צורך לשמור ענפים שכבר מוזגו. |

| **סוג ענף** | **מתי לפתוח** | **דוגמת שם** | **מה הוא מכיל** | **מה קורה בסוף** |
| --- | --- | --- | --- | --- |
| **פיצ’ר** | כל פעם שמתחיל פיצ’ר קוד חדש | feature/react-skeleton | קבצי קוד, טסטים, תיעוד שקשור לאותו פיצ’ר | Merge → main |
| **תיק‑באג** | תיקון נקודתי | fix/login-bug | תיקון קוד קטן + טסט | Merge → main |
| **דוקומנטציה‑מתמשכת** | קובץ יומן שבועי או שינוי גדול בתיעוד | docs/journal-w17-2025 | Markdown יומן, קבצי README, ADR | Merge → main |
| **hotfix** (נדיר) | תיקון חם בפרודקשן | hotfix/critical-db | שינוי דחוף שצריך לפרודקשן מיד | Merge → main + Deploy |