# Day 0 Execution Guide – NCS B2B E-Commerce

This guide walks through Day 0 tasks (09:00–16:15) for initializing the NCS B2B e-commerce project. We follow the Month 1 plan, which emphasizes **project initialization and base system setup**[[1]](file://file-Pi8X5EKhLZSK4GWXhGGajZ#:~:text=start%20to%20go,launch). Each timeline block includes a *Goal*, terminal-ready *Commands*, *Files* created/edited, and *Acceptance* criteria.

## Timeline (09:00–16:15)

* **09:00–10:30 — Environment Setup:**
* **Goal:** Configure development workstation with required tools.
* **Commands:**
* # Install Homebrew (if macOS/Linux) and Node version manager (nvm)  
  /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"  
  brew install nvm  
  mkdir -p ~/.nvm  
  # Install Node 18 LTS via nvm (as recommended[2][3])  
  nvm install --lts  
  nvm use --lts  
  node -v # verify Node v18+  
  # Install GitHub CLI and other tools  
  brew install gh docker   
  gh --version
* **Files:** None (system-level installs).
* **Acceptance:** node -v shows an LTS version ≥18 (per stack recommendation[[2]](file://file-3ZuJpEJ56Sg1Btoo4fCd87#:~:text=Technology%20Stack%20,Redis%20Queue%2FBullMQ%20for%20async%20processing)); gh auth login succeeds; basic tools (git, gh, docker) are operational.
* **10:30–12:00 — Git Repository Initialization:**
* **Goal:** Create the GitHub repo ncs-ecom and populate it with base files.
* **Commands:**
* gh auth login # ensure authenticated with GitHub  
  gh repo create NCSNetworks/ncs-ecom --public --confirm --description "NCS B2B E-Commerce platform (monorepo)"  
  git clone https://github.com/NCSNetworks/ncs-ecom.git  
  cd ncs-ecom  
  # Create initial project files  
  cat <<EOF > README.md  
  # NCS B2B E-Commerce  
    
  NCS Networks B2B e-commerce platform (modular monolith: NestJS + Next.js, PostgreSQL, Redis)[2][3].  
  EOF  
  curl -o LICENSE https://www.apache.org/licenses/LICENSE-2.0.txt  
  echo -e "node\_modules/\n.env" > .gitignore
* **Files:** README.md, LICENSE (Apache 2.0), .gitignore (ignores node\_modules, .env), and placeholder governance files (.github/) created.
* **Acceptance:** Repository ncs-ecom is visible on GitHub with these files on the main branch. A first commit should be made, e.g.,
* git add README.md LICENSE .gitignore  
  git commit -m "chore: initial repo setup with README, LICENSE, .gitignore"  
  git push origin main
* Verify via GitHub UI or gh repo view ncs-ecom that files exist.
* **13:00–14:30 — Branch Protection & Governance Files:**
* **Goal:** Enforce branch policies and add governance templates.
* **Commands:**
* # Protect main branch (require PR reviews, disallow force pushes)  
  gh api --method PUT /repos/NCSNetworks/ncs-ecom/branches/main/protection \  
   -f required\_pull\_request\_reviews.dismiss\_stale\_reviews=true \  
   -f required\_pull\_request\_reviews.required\_approving\_review\_count=1 \  
   -f enforce\_admins=true \  
   -f restrictions=""   
  # Create issue and PR templates  
  mkdir -p .github/ISSUE\_TEMPLATE  
  cat <<EOF > .github/ISSUE\_TEMPLATE/bug\_report.md  
  ---  
  name: Bug report  
  about: Report a problem or unexpected behavior  
  title: "[BUG] REQ-XYZ-??: "  
  labels: bug  
  ---  
  \*\*Describe the bug\*\*   
  A clear and concise description of what the bug is.  
    
  \*\*To Reproduce\*\*   
  Steps to reproduce the behavior.  
    
  \*\*Expected behavior\*\*   
  A clear description of what you expected to happen.  
    
  \*\*Environment (please complete the following information):\*\*  
  - OS:  
  - Browser:  
    
  \*\*Additional context\*\*   
  Add any other context about the problem here.  
  EOF  
    
  cat <<EOF > .github/ISSUE\_TEMPLATE/feature\_request.md  
  ---  
  name: Feature request  
  about: Suggest a new feature or enhancement  
  title: "[FEAT] REQ-XYZ-??: "  
  labels: enhancement  
  ---  
  \*\*Describe the solution you'd like\*\*   
  A clear description of what you want to happen.  
    
  \*\*Acceptance Criteria\*\*   
  - Criterion 1  
  - Criterion 2  
    
  \*\*Additional context\*\*   
  Add any other context or screenshots about the feature request here.  
  EOF  
    
  # Add files and commit  
  git add .github/ISSUE\_TEMPLATE  
  git commit -m "chore: add issue templates for bug and feature"  
  git push
* **Files:** .github/ISSUE\_TEMPLATE/bug\_report.md, .github/ISSUE\_TEMPLATE/feature\_request.md.
* **Acceptance:** In the GitHub repo UI, the Settings → Branches page shows protection on main (no force pushes, require ≥1 reviewer). The .github/ISSUE\_TEMPLATE directory appears with both templates.
* **14:30–16:15 — Label Set, Project Board & Sample Issues:**
* **Goal:** Define issue labels, set up project board, and stub initial issues from core requirements.
* **Commands:**
* # Create standard labels  
  gh label create "bug" -c d73a4a -d "Bug: something isn't working"  
  gh label create "enhancement" -c a2eeef -d "Feature: new capability or improvement"  
  gh label create "documentation" -c 0075ca -d "Docs: documentation changes"  
  gh label create "discussion" -c 5319e7 -d "Needs discussion"  
    
  # Create a project board with specified columns  
  gh project create "NCS B2B Board" --body "Day-1 task board" --board --repo NCSNetworks/ncs-ecom  
  gh project column create --project "NCS B2B Board" "Backlog"  
  gh project column create --project "NCS B2B Board" "Ready"  
  gh project column create --project "NCS B2B Board" "In Progress"  
  gh project column create --project "NCS B2B Board" "In Review"  
  gh project column create --project "NCS B2B Board" "Done"  
    
  # Stub sample issues with Requirement IDs (from design docs) and AC  
  gh issue create --title "REQ-RFQ-001: Customer RFQ submission (cart → quote request)" \  
   --body "- \*\*Given\*\* a user submits an RFQ via their shopping cart\n- \*\*When\*\* the RFQ is submitted,\n- \*\*Then\*\* the user’s portal shows the RFQ with status \"Pending NCS Response\" and a confirmation email is sent[4]." \  
   --label "enhancement"  
    
  gh issue create --title "REQ-RFQ-002: Admin quote management interface" \  
   --body "- \*\*Given\*\* sales staff view an RFQ,\n- \*\*When\*\* the staff edits and marks a quote as \"Quoted\",\n- \*\*Then\*\* the customer’s portal shows the updated quote and the customer is notified[5]." \  
   --label "enhancement"  
    
  gh issue create --title "REQ-RFQ-003: Quote-to-order conversion (one-click)" \  
   --body "- \*\*Given\*\* a customer accepts a quote,\n- \*\*When\*\* they click \*\*Convert to Order\*\*,\n- \*\*Then\*\* the quote items and prices transfer to a new order and checkout proceeds without re-entering details[6]." \  
   --label "enhancement"  
    
  gh issue create --title "REQ-PAY-001: PromptPay QR code generation (real-time)" \  
   --body "- \*\*Given\*\* a user selects PromptPay at checkout,\n- \*\*When\*\* the order is finalized,\n- \*\*Then\*\* the confirmation page displays a scannable QR code for payment and the order status is set to \"Pending Payment\"[7]." \  
   --label "feature"  
    
  gh issue create --title "REQ-PAY-002: Bank transfer payment with proof upload" \  
   --body "- \*\*Given\*\* a user selects Bank Transfer,\n- \*\*When\*\* the order is placed,\n- \*\*Then\*\* the system provides bank transfer instructions and allows the user to upload payment proof afterward[8]." \  
   --label "feature"  
    
  gh issue create --title "REQ-CAT-001: Hierarchical category navigation" \  
   --body "- \*\*Given\*\* a category hierarchy is defined,\n- \*\*When\*\* a user browses categories,\n- \*\*Then\*\* they can drill down through multiple levels (e.g., Networking → Switches → Managed) to find products[9]." \  
   --label "feature"  
    
  gh issue create --title "REQ-CART-001: Persistent shopping cart across sessions" \  
   --body "- \*\*Given\*\* a logged-in user with items in their cart,\n- \*\*When\*\* they log out and later log in again,\n- \*\*Then\*\* their cart contents remain intact and accessible[10]." \  
   --label "feature"  
    
  gh issue create --title "REQ-ACC-001: Multi-user account with role-based permissions" \  
   --body "- \*\*Given\*\* a business account with multiple users,\n- \*\*When\*\* an approver is required for high-value orders,\n- \*\*Then\*\* only users with the approver role can finalize orders exceeding the spending limit[11]." \  
   --label "feature"  
    
  gh issue create --title "REQ-PAY-003: Automated Thai tax invoice generation" \  
   --body "- \*\*Given\*\* an order is completed,\n- \*\*When\*\* payment is confirmed,\n- \*\*Then\*\* the system generates a PDF invoice in Thai tax format (showing VAT breakdown) for download[12]." \  
   --label "feature"  
    
  gh issue create --title "REQ-PTY-001: Partner registration and portal setup" \  
   --body "- \*\*Given\*\* a partner company registers on the platform,\n- \*\*When\*\* they submit required credentials,\n- \*\*Then\*\* the account is created under a partner role (e.g., Bronze/Silver tier) after verification[13]." \  
   --label "feature"
* **Acceptance:** In GitHub: labels “bug”, “enhancement”, “documentation”, etc., are present. The “NCS B2B Board” project exists with columns Backlog, Ready, In Progress, In Review, Done. Ten issues (at least) are created in the Backlog with titles matching the REQ IDs and AC bullet lists referencing the design docs (e.g. [*[4]*](file://file-99vRGybfzD8cjE33eWqvvY#:~:text=Customer%20RFQ%20Submission%3A%20The%20user,an%20email%20acknowledgment%20as%20well) etc.). Verify by viewing issue list and project board.

## Git Artifacts

* **Repo Structure:** Ensure ncs-ecom has at least README.md, LICENSE (Apache 2.0 or MIT), .gitignore, and a CODE\_OF\_CONDUCT.md and CONTRIBUTING.md if needed. Each file should have meaningful content (e.g. README outlines project purpose).
* **Initial Commit:** Use a clear commit message, e.g. chore: initial project setup (README, LICENSE, .gitignore). The commit history should start with this snapshot.
* **Branch Protection:** Protect the main branch via GitHub settings or CLI: require pull requests with ≥1 review, disallow force pushes, optionally require status checks (e.g. CI).
* **Governance Files:** Under .github/, the issue and PR templates are present. Optionally include CONTRIBUTING.md and a SECURITY.md.
* **Example Git Commands:**
* gh repo create NCSNetworks/ncs-ecom --public --confirm  
  git clone git@github.com:NCSNetworks/ncs-ecom.git  
  cd ncs-ecom  
  touch README.md LICENSE .gitignore  
  git add .  
  git commit -m "chore: initial repo setup with base files"  
  git push -u origin main

## Governance Setup

* **Issue Templates:** Two templates (bug\_report.md, feature\_request.md) in .github/ISSUE\_TEMPLATE/ guide contributors on writing issues. They auto-populate the “Issue” form on GitHub.
* **Labels:** Create a common label set, e.g.: bug, enhancement (or feature), documentation, discussion, etc. Color-code them for clarity. For example, red for bugs (#d73a4a), blue for enhancements (#a2eeef), green for docs (#0075ca).
* **Project Board:** Use GitHub Projects (v2) or Classic Projects to create a board “NCS B2B Board” with columns **Backlog**, **Ready**, **In Progress**, **In Review**, **Done**. Associate issues by moving them into columns or linking via PRs.
* **Stubbed Issues:** We created at least 10 issues tied to core requirements from the Tech Design Doc, each with a REQ ID in the title and acceptance criteria citing the design docs. For example, **REQ-RFQ-001** for RFQ submission[[4]](file://file-99vRGybfzD8cjE33eWqvvY#:~:text=Customer%20RFQ%20Submission%3A%20The%20user,an%20email%20acknowledgment%20as%20well) and **REQ-PAY-001** for PromptPay QR[[7]](file://file-99vRGybfzD8cjE33eWqvvY#:~:text=instructions%20and%20allow%20the%20user,until%20staff%20confirms%20the%20transfer). These issues should be in the Backlog column.

## Workstation Preparation

* **Homebrew & nvm:** Already installed above.
* **Node.js (v18 LTS):** Ensure the LTS version is active (node -v shows ≥18.x). This aligns with the recommended stack (NestJS on Node)[[2]](file://file-3ZuJpEJ56Sg1Btoo4fCd87#:~:text=Technology%20Stack%20,Redis%20Queue%2FBullMQ%20for%20async%20processing).
* **CLI Tools:** Install GitHub CLI (gh), Docker (for containerization), and any other needed tools (e.g. psql client if you plan local Postgres).
* **VS Code Extensions:** Recommended extensions: ESLint, Prettier, Docker, GitLens, Path Intellisense, and a REST client (e.g. Thunder Client).
* **VS Code Settings:** Example settings.json snippet to enforce code quality:
* {  
   "editor.formatOnSave": true,  
   "editor.codeActionsOnSave": {  
   "source.fixAll.eslint": true  
   },  
   "files.autoSave": "afterDelay",  
   "files.autoSaveDelay": 1000,  
   "editor.fontSize": 14  
  }
* **Environment Variables:** Set up .env.local or similar with placeholders (do **not** commit secrets). Ensure your shell RC (e.g. ~/.bashrc or ~/.zshrc) loads nvm.

## Evidence Pack

Collect the following for the report (screenshots or command outputs):

* **Configuration Verification:** Run and capture outputs of node -v, npm -v, gh auth status, and docker version. Save as evidence that Node 18+, GH CLI, Docker are installed.
* **Repository Checklist:** Screenshot the GitHub repo home showing README.md, LICENSE, .gitignore, and the protected branch settings.
* **Governance Artifacts:** Capture (a) the Issue Templates list under “Settings → Issues”, (b) the Labels list, (c) the Projects board overview with columns, and (d) one of the stub issue pages (showing title, AC, and the cited reference context).
* **Branch Protection:** Screenshot the “Branches” settings page showing protection rules for main.
* **Project Board:** Screenshot or list showing columns Backlog/Ready/In Progress/In Review/Done populated (or empty ready for use).
* **Issue Samples:** Screenshot at least one issue detail (with title and AC) that includes a cited reference, and show its assignment to the project board.

## Day 0 Risks and Done Criteria

* **Risks:** Potential Day-1 risks include: delays in receiving necessary permissions (GitHub org access), network/install issues blocking tool setup, or configuration mismatches (e.g. wrong Node version). Mitigate by having rollback plans (e.g. use a pre-existing dev machine or container) and verifying each install. Also watch for any unexpected company policies (e.g. proxy settings) that could impede setup.
* **Done Means:** At end of Day 0, the following should be **complete**: development workstation is ready (brew/nvm/Node/CLI tools installed and verified); ncs-ecom repo exists with initial files (README, LICENSE, .gitignore); branch protection is enabled on main; issue templates and labels are in place; the project board is created with correct columns; and at least 10 requirement-based issues are stubbed with titles, acceptance criteria, and source citations. All acceptance checks above should be satisfied without gaps.

**Sources:** We follow the project’s tech/design documentation. For example, the **month 1 plan** calls for initial setup work[[1]](file://file-Pi8X5EKhLZSK4GWXhGGajZ#:~:text=start%20to%20go,launch). The **tech stack** mandates Node/Nest/Next with Postgres and Redis[[2]](file://file-3ZuJpEJ56Sg1Btoo4fCd87#:~:text=Technology%20Stack%20,Redis%20Queue%2FBullMQ%20for%20async%20processing)[[3]](file://file-7DwU6oYXYRDueMqEHMcx9p#:~:text=Backend%3A%20Node,ecosystem%20across%20the%20full%20stack), so we install Node 18 LTS. The **requirements docs** provide REQ IDs and AC that we use when stubbing issues (e.g. RFQ and payment workflows)[[4]](file://file-99vRGybfzD8cjE33eWqvvY#:~:text=Customer%20RFQ%20Submission%3A%20The%20user,an%20email%20acknowledgment%20as%20well)[[13]](file://file-3ZuJpEJ56Sg1Btoo4fCd87#:~:text=REQ,catalogs%2C%20presentations). All instructions are aligned with those source guidelines.

[[1]](file://file-Pi8X5EKhLZSK4GWXhGGajZ" \l ":~:text=start%20to%20go,launch) Streamlined Product Blueprint – NCS Networks B2B E-Commerce Platform.docx

<file://file-Pi8X5EKhLZSK4GWXhGGajZ>

[[2]](file://file-3ZuJpEJ56Sg1Btoo4fCd87#:~:text=Technology%20Stack%20,Redis%20Queue%2FBullMQ%20for%20async%20processing) [[9]](file://file-3ZuJpEJ56Sg1Btoo4fCd87#:~:text=REQ,001.6%2A%2A%3A%20Technical%20datasheet%20downloads%20%28PDF) [[10]](file://file-3ZuJpEJ56Sg1Btoo4fCd87#:~:text=REQ,step%20checkout%20with%20progress%20indicators) [[11]](file://file-3ZuJpEJ56Sg1Btoo4fCd87#:~:text=,identification%20and%20special%20pricing%20access) [[12]](file://file-3ZuJpEJ56Sg1Btoo4fCd87#:~:text=REQ,Thai%20tax%20format%20PDF) [[13]](file://file-3ZuJpEJ56Sg1Btoo4fCd87#:~:text=REQ,catalogs%2C%20presentations) 1.1. Master-Blueprint-Prompt-NCS-B2B-Ecommerce.md

<file://file-3ZuJpEJ56Sg1Btoo4fCd87>

[[3]](file://file-7DwU6oYXYRDueMqEHMcx9p#:~:text=Backend%3A%20Node,ecosystem%20across%20the%20full%20stack) 1.1. B2B E-Commerce Blueprint Development.docx

<file://file-7DwU6oYXYRDueMqEHMcx9p>

[[4]](file://file-99vRGybfzD8cjE33eWqvvY#:~:text=Customer%20RFQ%20Submission%3A%20The%20user,an%20email%20acknowledgment%20as%20well) [[5]](file://file-99vRGybfzD8cjE33eWqvvY#:~:text=Admin%20Quote%20Management%3A%20NCS%20staff,85) [[6]](file://file-99vRGybfzD8cjE33eWqvvY#:~:text=Given%20a%20customer%20accepts%20a,entering%20cart%20details%5B53) [[7]](file://file-99vRGybfzD8cjE33eWqvvY#:~:text=instructions%20and%20allow%20the%20user,until%20staff%20confirms%20the%20transfer) [[8]](file://file-99vRGybfzD8cjE33eWqvvY#:~:text=gateway%20,Acceptance%3A%20Given) 1.2. Streamlined Product Blueprint for NCS Networks B2B E-Commerce Platform.docx

<file://file-99vRGybfzD8cjE33eWqvvY>