# **Quest of Code – Building a CS Portfolio (AP CSA 2025)**

## **Requirement Brief and Learning Goals**

**Course Details**

* Grade Level: 9 – 12
* Content Area: AP Computer Science A / Software Engineering Pathway
* Frameworks: College Board AP CSA (2024), CSTA 3A Standards (2017)
* Project Structure: Quest-based, team-driven, modular development of a professional CS portfolio

Students will collaboratively design and implement a mini-quest (a series of modules) contributing to the mega-quest: Building a CS Portfolio.

Each team’s mini-quest emphasizes a core domain (front-end, back-end, AI, data, or career development) and culminates in deployable portfolio artifacts that highlight teamwork, coding proficiency, and design thinking.

### **1a. Elaboration on Goals**

The Building a CS Portfolio Quest transforms classroom learning into an authentic engineering experience.

Students assume professional roles such as developer, designer, analyst, and curator to craft real-world applicable artifacts that showcase technical and career readiness.

Each mini-quest functions as a chapter in a larger project: learning to create, connect, analyze, and present one’s work online through platforms such as LinkedIn.

* Reinforces object-oriented design, version control, and documentation.
* Simulates an Agile Scrum environment with Kanban tracking and peer review.
* Builds a personal portfolio integrating front-end, back-end, and AI components.

### **1b. Prerequisites**

* Prior exposure to Java, Python (Flask), HTML/CSS/JS, and GitHub basics.
* Familiarity with Git workflows (branches, commits, pull requests).
* Access to the Open Coding Society organization and GitHub Projects Kanban Board: [OCS Projects Board](https://github.com/orgs/Open-Coding-Society/projects/2/views/1)

### **1c. Background Context**

This quest scales that learning into a team-based full-stack journey where each group builds a unique module contributing to the schoolwide CS Portfolio Site. Deliverables include code, documentation, and presentations during the “Night at the Museum” showcase.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## **Mini-Quest Overview**

| **Mini-Quest** | **Team** | **Focus Area** | **Example Submodules / Lessons** |
| --- | --- | --- | --- |
| **Frontend Development** | **Creators** | UI Design & Interactivity | Purpose of Frontend • Markdown to HTML • CSS Styling • JS Event Handling |
| **Backend Development** | **Encrypters** | Data Management & APIs | Database Schemas • Flask Routes • Spring Boot Controllers |
| **Data Visualization** | **Applicators** | Analytics & Machine Learning | Collecting Data • Visual Representations • Intro to ML |
| **AI Usage** | **Thinkers** | Prompt Engineering & Integration | Responsible Prompting • API Calls • Automation |
| **CS Writing / Resume Building** | **Grinders** | Professional Communication | Technical Blogging • Resume Writing • LinkedIn Profiles |
| **Analytics / Admin** | **Curators** | Tracking and Evaluation | Metrics Dashboards • Admin Portal • Feedback Analysis |
| **User Flow / Integration** | **Innovators** | UX Design & System Flow | Navigation Design • Cross-Module Integration • Testing UX Consistency |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## **Learning Phases**

### **Phase 1 – Ideation**

Teams brainstorm the purpose and narrative of their mini-quest.  
They submit a concept brief describing objectives, target audience, and expected portfolio artifact.  
Deliverable: Google Doc Concept Proposal

### **Phase 2 – Storyboarding & Prototyping**

Visual planning through wireframes and lesson flowcharts.  
Front-end and UX teams sketch designs while back-end teams define data flows.  
Deliverable: Storyboard + Technical Plan

### **Phase 3 – Development**

Teams collaborate through GitHub Projects.  
Each module follows the SDLC process: design → build → test → review.  
Deliverables: Functional prototype with working code and documentation

### **Phase 4 – Integration & Testing**

Modules connect to form the mega-quest portal.  
User Flow and Curator teams coordinate data connections and analytics.

### **Phase 5 – Showcase & Reflection**

Live demonstration during the school showcase.  
Teams gather feedback and reflect on collaboration and technical growth.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## **Standards Alignment**

**AP CSA Objectives**

* Program design and algorithm development.
* Use and implementation of classes and objects.
* Code organization and documentation practices.

**CSTA 3A Standards**

* 3A-AP-13: Decompose complex problems for design and review.
* 3A-AP-17: Systematically design programs for broad audiences.
* 3A-AP-22: Collaborate using version control and team tools.

**ISTE Digital Citizenship Links**

* Empowered Learner – Independent and ethical use of AI and code assistants
* Knowledge Constructor – Data handling and source evaluation
* Innovative Designer – Iterative prototyping and testing

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## **Technology Resources**

| **Domain** | **Primary Tools** | **Purpose** |
| --- | --- | --- |
| Front-End | VS Code, GitHub Pages, Chrome DevTools | Build and deploy UI components |
| Back-End | Python Flask / Spring Boot, SQLite | Handle data and server-side logic |
| AI Integration | ChatGPT / Gemini / OpenAI APIs | Demonstrate ethical AI usage |
| Data Visualization | Python Pandas / Matplotlib / Recharts | Represent data clearly and securely |
| Collaboration | GitHub Projects, Google Docs, Slack | Agile workflow and documentation |
| Deployment | GitHub Actions | Automate build and deployment |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## **Assessment Plan**

| **Checkpoint** | **Description** | **Evidence of Completion** |
| --- | --- | --- |
| 1. Ideation Approval | Teams submit concept doc and story alignment | Approved proposal on Google Docs |
| 2. Prototype Review | Initial front-end and back-end demo | Deployed link + GitHub commit history |
| 3. Integration Review | Cross-module testing and UX evaluation | Working data flow across modules |
| 4. Showcase | Public demo and presentation | Event participation feedback form |
| 5. Reflection | Individual analysis of learning and teamwork | Blog entry or README summary |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## **Digital Citizenship and Ethics**

* Use of open-source code requires proper licensing and attribution.
* AI-assisted coding must include human verification and source citation.
* Personal data must not be stored without consent.
* Communication within GitHub and Google Docs should model respectful, inclusive collaboration.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## **Implementation Plan**

### **1. Online Workflow**

* Environment: GitHub organization + VS Code Live Share
* Tracking: Kanban Board with milestones for each lesson phase
* Communication: Slack for daily syncs
* Documentation: Google Docs + GitHub Wiki for lesson content

### **2. Module Publishing**

* Each team repository deploys via GitHub Pages.
* The Curators aggregate modules into a central “Mega-Quest” Portal.
* Innovators manage user navigation and progress tracking.

### **3. Showcase Preparation**

* Teams finalize README files and record short demo videos.
* Presentations highlight technical and ethical considerations.
* Feedback is logged using Google Forms and GitHub Issues.