

Task 1: Ideation & Innovation

Q.1: Innovation:

Innovation is Everywhere

From prehistoric times to the twenty-first century, innovation has consistently impacted how humans live and society evolve. It has enabled people to generate food and energy, manage social structures, and create cultural meaning. Innovation is an important component of how the world evolves, not just via major advances, but also through continuous improvement and new ways of tackling issues. Importantly, creativity isn't limited to skilled individuals; everyone has the ability to innovate because humans are naturally good at spotting patterns and adapting them to solve problems. Innovation has an impact on everyone's lives, both positively and negatively, but it ultimately promotes continuous advancement. Even if a "perfect" world is unachievable, innovation enables civilisations to continuously improve welfare, systems, and daily life throughout time.

The Onward March of Technology

The evolution of technology has happened in multiple stages, reflecting how innovation reshapes civilisation over time. Klaus Schwab describes history as four industrial revolutions, each defined by significant technological advances that altered how people worked, lived, and interacted. These stages demonstrate that technological change isn't random, but rather comes in waves, with fresh ideas driving periods of rapid growth before becoming normalised. Technology is increasingly intertwined with all aspects of life, from manufacturing and communication to health and social structures. As innovation increases, its effect grows, transforming economies and societies at a faster rate. This continuous march of technology demonstrates how progress is continuous, with each advancement building on the one that came before.

Creativity and You

Creativity is a fundamental human trait, as everything in our world contains some type of creative input. Despite this, many people believe they are not creative because creativity is frequently connected with specific fields such as art, music, or performance. This has given birth to the idea that creativity is a fixed attribute that people are either born with or lack. In actuality, creativity is essential to all forms of innovation since new ideas and solutions cannot exist without it. Everyone possesses a natural talent to be creative, however this skill is frequently diminished over time. According to Ken Robinson, education and society can impede creativity by focusing too much on avoiding mistakes, ignoring the fact that experimenting and failure are necessary parts of the creative process. Creativity generally emerges from experimentation, learning, and adaptation.

Q.2: Ideation:

Project Title:

Design and development of a secure digital career management platform to support aspiring actors with productivity and cyber safety.

Aim:

To design, develop and evaluate a secure web-based platform that improves aspiring actors' productivity while reducing common online security risks.

Project Objectives:

- Research the regular digital workflow requirements of aspiring actors, as well as the typical cyber dangers (phishing, frauds, and impersonation) that affect creatives online.
- Design system requirements, user journeys, and wireframes for a secure productivity platform, taking into account authentication and data processing.
- Create a functioning prototype with key productivity elements (audition tracker, task scheduler, document store) and integrated cyber-safety guidance.

- Implement security features such as secure login, input validation, basic privacy settings, and safe reporting/flagging of suspicious contacts or messages.
- Test and assess usability and security performance using user feedback and basic security checks, then improve the platform depending on the results.

Project Description:

This project will design and build a secure web-based platform to assist aspiring actors in managing their careers safely and efficiently. The site will include essential productivity tools including an audition tracker, work planning, and organised storage for scripts or self-tape notes. In addition, it will have integrated cybersecurity awareness elements aimed at dangers common to creatives, such as fraudulent casting calls, phishing emails, and impersonation. The project will prioritise secure design, including authentication, safe data processing, and fundamental protection controls. The end result will be a tested prototype that shows how productivity and cybersecurity may be integrated to help creative people in a modern industry.

Q.3: Research:

1. <https://owasp.org/www-project-top-ten/>
2. <https://learn.microsoft.com/en-us/training/paths/secure-your-web-app/>
3. <https://www.netacad.com/courses/cybersecurity/introduction-cybersecurity>
4. <https://www.linkedin.com/learning/programming-foundations-web-security-22680062/the-importance-of-security?u=77966930>
5. <https://www.linkedin.com/learning/programming-foundations-secure-coding-2018/implement-secure-code-with-your-team?u=77966930>

Task 2: project management

Q.1: Document the **Project scope** including:

1. Proposed solution

A secure web-based platform for aspiring actors that includes:

- Productivity tools include a role/application tracker, an audition calendar, to-do lists, notes, and file storage (headshots, self-tapes, scripts) with reminders.
- Security features include secure login, strong password rules, optional two-factor authentication, encrypted storage (where applicable), safe file upload, access controls, security logging, and basic security guidance prompts.
- Goal: Assist actors in staying organised while decreasing risks such as weak passwords, account takeover, unsafe file sharing, and poor data management.

1. Required resources / technologies (what you need to implement it)

Core build tools:

- Laptop + internet
- Code editor (VS Code)
- Version control (Git + GitHub)

Web app (tech stack):

- Front-end: HTML/CSS/JavaScript
- Back-end: one of (decide later)
- Python (Flask) or Node.js (Express) or .NET (ASP.NET)
- Database: SQLite
- Security requirements
- Authentication (login + sessions)
- Password hashing (never store plain passwords)
- Access control (users only see their own data)
- Input validation (protect against injection/XSS)
- HTTPS (deployment stage / hosting)
- Logging (basic security/audit logs)

Testing & evaluation:

- User testing with a few students/actors (feedback form)
- Security checklist based on OWASP Top 10
- Screenshots/evidence from your learning sources (appendix)

Optional

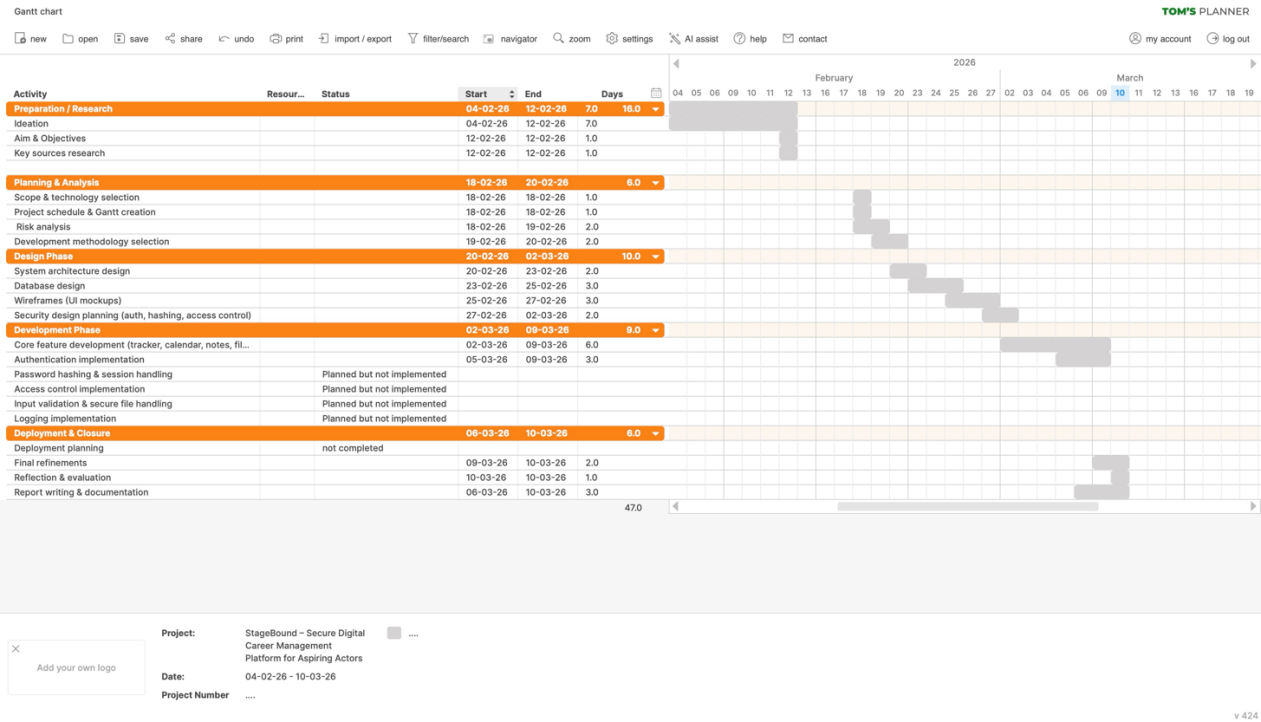
- 2FA, password manager tips, simple “security score”, email alerts for suspicious logins.

Project start 4/2/2026
Project end 10/03/2026

Task	Start Date & End Date
Ideation	4/2/2026 - 12/2/2026
Aim & Objectives	12/2/2026 - 12/2/2026
Key sources research	12/2/2026 - 12/2/2026
Scope & technology selection	18/2/2026 - 19/2/2026
Planning & Gantt	18/2/2026 - 20/2/2026
Risk analysis	18/2/2026 - 21/2/2026
Methodology selection	22/2/2026 – 25/2/2026
Design phase (wireframes, architecture)	22/2/2026 – 1/3/2026
Core feature development	1/3/2026 – 9/3/2026
Security implementation	Planned but not implemented in this stage
User testing	6/3/2026 - 6/3/2026
Security evaluation (OWASP-based)	Planned for future iteration
Final improvements	9/3/2026 – 10/3/2026

Deployment planning	
Reflection & report writing	8/3/2026 – 10/3/2026

Q.2: Using a software tool of your choice, create a Gantt chart:



Q.3: Identify some **risks/challenges** your project could face and document what you can do to reduce/remove them:

Risk No	Risk	Impact	Contingency
1	Hardware failure / Data loss	High	Apply the rule of 3: back up code to GitHub, use cloud storage (OneDrive/Google Drive), and maintain local copies. Enable version control from the outset.
2	Lack of technical skill in security implementation (authentication, hashing, sessions)	High	Complete LinkedIn Learning and Microsoft Learn security modules before development. Use OWASP Top 10 as implementation checklist. Start security features early rather than at the end.
3	Time management / Running out of time	High	Use structured Gantt chart planning. Dedicate fixed weekly hours. Prioritise Must-have requirements first. Leave optional features (2FA, security score) until core system is complete.
4	Underestimating development time (especially authentication & database logic)	High	Break development into small milestones. Test features incrementally. Avoid implementing multiple complex features simultaneously.
5	Scope creep (adding unnecessary features)	Medium	Strictly follow prioritised Requirements Specification. Focus only on productivity + core security features. Add enhancements only if time permits.
6	Security vulnerabilities in implementation (e.g., improper password storage, missing validation)	High	Implement password hashing from day one. Apply input validation and access control early. Use OWASP checklist during testing phase.
7	Technical stack decision risk (choosing unfamiliar framework)	Medium	Select a simple, well-documented stack (e.g., Flask or Node). Research before committing. Avoid switching stacks mid-project.
8	Database design errors requiring restructuring	Medium	Plan basic data model before coding. Keep schema simple. Test with sample data early to identify design flaws.

9	Deployment / Hosting issues	Medium	Research hosting platform early (Render / Railway / Azure). Test deployment on small prototype before final build. Keep local working version as fallback.
10	Usability issues identified late	Medium	Conduct small user testing sessions mid-development. Collect structured feedback and refine interface before final submission.
11	Personal motivation drop / inconsistent work pattern	Medium	Follow weekly milestone targets from Gantt chart. Use structured phase approach (SDLC). Review progress every week.
12	Competing time commitments (other modules, acting, gym, etc.)	High	Allocate fixed project time blocks per week. Treat project work as scheduled commitment. Adjust optional feature scope if workload increases.

Task 3: Project Development method & Design work

Q.1: Describe a suitable Development method that will help you plan and complete your project.

A hybrid development approach combining the Systems Development Life Cycle (SDLC) and Scrum-Lite will be used to plan and complete this project. The SDLC provides a clear structured framework consisting of Analysis, Design, Development, Testing, and Evaluation stages (Pressman & Maxim, 2020). This structured model suits the project as it ensures careful planning of requirements, security considerations, system architecture, and testing before deployment. Given that the project focuses on building a secure web-based platform, structured documentation and stage control are important to reduce risks and maintain quality.

To introduce flexibility, Scrum-Lite principles will be incorporated during the Development and Testing phases. Scrum-Lite supports iterative feature implementation through short time-boxed sprints, allowing continuous improvement

and adaptation (Schwaber & Sutherland, 2020). This is particularly useful for implementing security features such as authentication, password hashing, and access control, which may require refinement based on testing results.

This hybrid approach balances stability and adaptability, enabling structured planning while allowing incremental improvements. It supports effective time management, risk reduction, and continuous evaluation, ensuring the project is completed within the defined timeframe and meets both productivity and security objectives. Rewrite

Q.2: Document and describe some project ‘Design work’. 2 pieces required, e.g. depending on the nature of your project example design work could include: a requirements specification, sketches/drawings, storyboard, wireframes, flow chart/UML diagram, ERD data structure, network structure diagram, network security architecture/diagram.

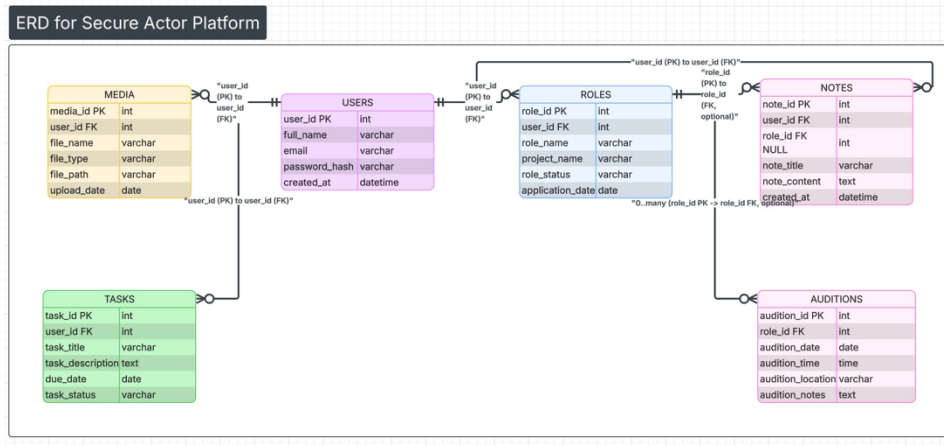
Design Work 1 — Requirements Specification

ID	Requirement Title	Priority	Description
FR01	User Registration	Must	The system must allow aspiring actors to create a personal user account securely.
FR02	User Login	Must	The system must allow registered users to log in securely using their account credentials.
FR03	Role/Application Tracker	Must	The system must allow users to add, edit, view and delete acting role applications.
FR04	Audition Calendar	Must	The system must provide a calendar feature for users to record and manage auditions.
FR05	Task Management	Must	The system must allow users to create, manage and track tasks related to auditions and acting work.

FR06	Notes Section	Should	The system should allow users to store personal notes linked to roles, auditions or tasks.
FR07	Media / File Upload	Should	The system should allow users to upload files such as headshots, scripts and self-tapes.
FR08	Access Control	Must	The system must ensure users can only access and manage their own personal data.
NFR09	Password Security	Must	The system must securely store passwords using hashing and enforce basic password security rules.
NFR10	Input Validation	Must	The system must validate and sanitise user input to reduce risks such as injection and XSS.

A requirements specification was created to specify the key features and system expectations for the planned secure web-based platform for aspiring actors. Functional requirements include role tracking, audition scheduling, and task management, whereas non functional requirements include password security and input validation. Each demand was assigned a unique identification, a priority level, and a short description. This clarifies the project scope, allows for the prioritisation of the most critical features, and ensures that development remains focused on the minimal viable product.

Design Work 2 — ERD / Database Design



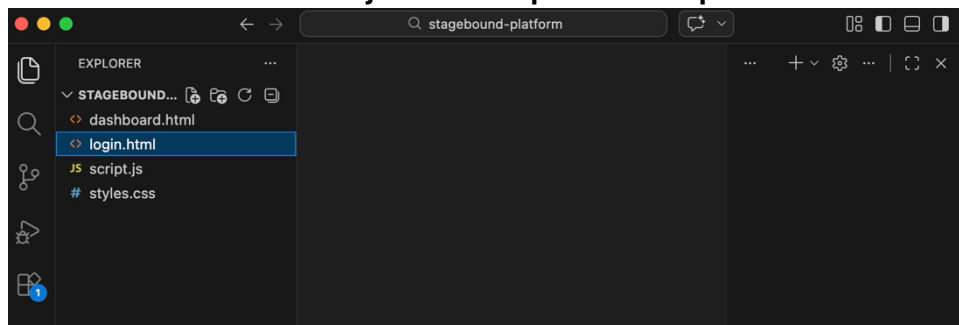
An Entity Relationship Diagram (ERD) was constructed to plan the data structure for the secure web-based platform for aspiring actors. The ERD defines fundamental entities such as Users, Roles, Auditions, Tasks, Media Files, and Notes, as well as the relationships between them.

The User's table is the key entity that maintains essential account information. Each user can have several linked records, including jobs applied for, tasks to fulfil, media materials uploaded, and personal notes. The Roles object contains information about acting roles and is related to Auditions, which allows the system to track audition dates, places, and notes.

Primary keys and foreign key associations are used to ensure data integrity and that each record is correctly associated with the relevant user or role. This ERD serves as a structured blueprint for database implementation and helps to support the system architecture's data layer.

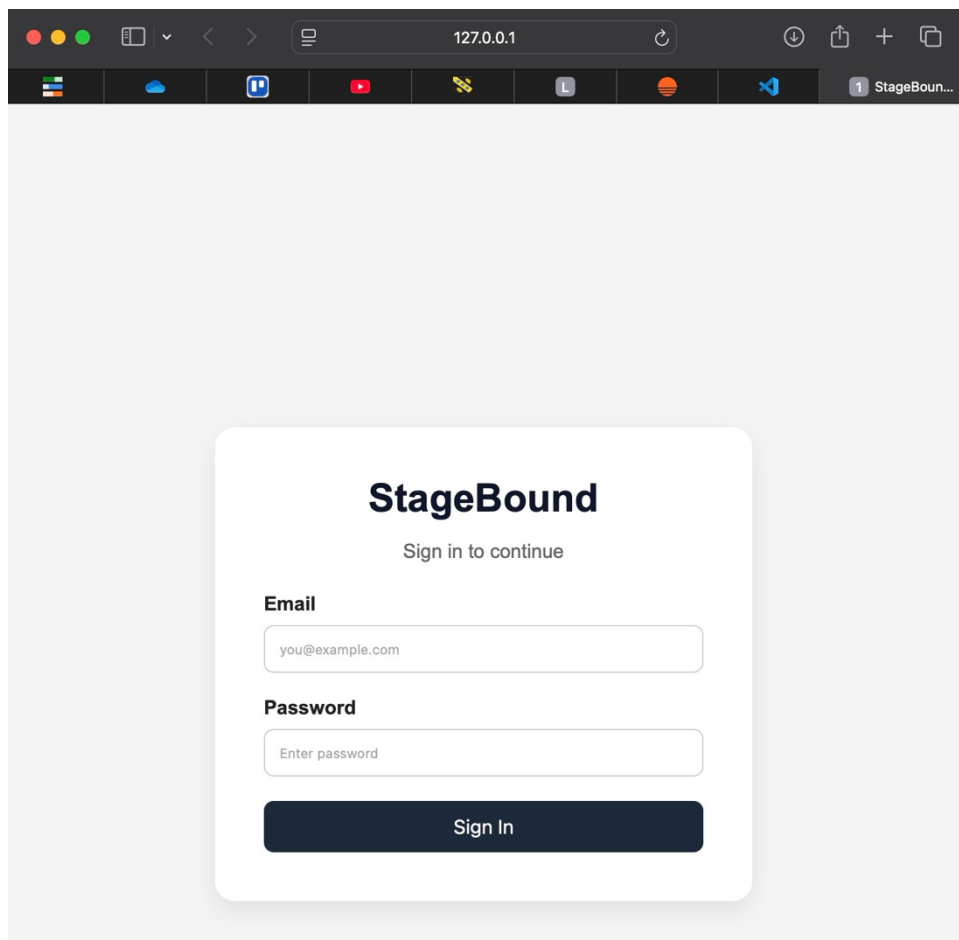
Q.3: Use several screenshots with brief description to document some initial practical implementation work for your project.

Screenshot 1 – Initial Project Development Setup



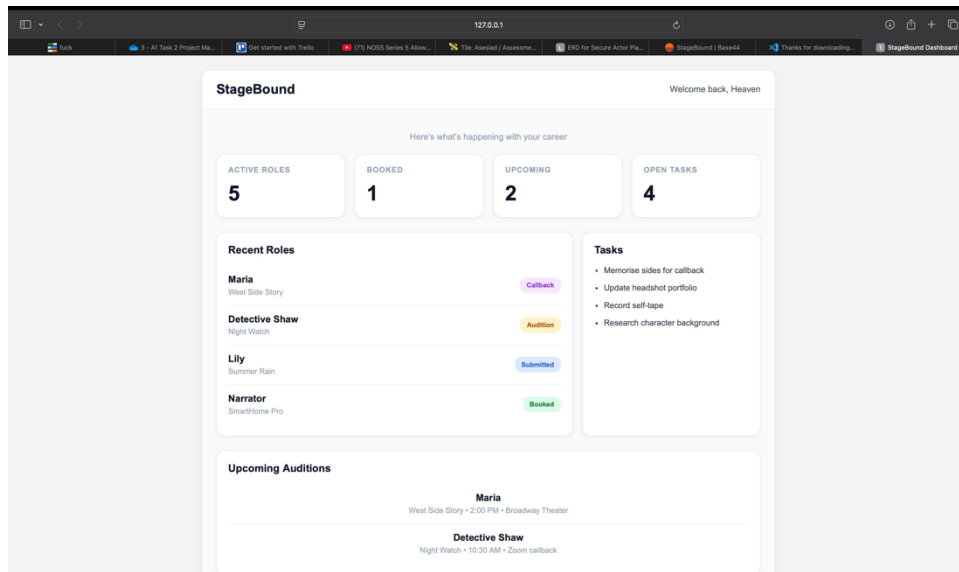
This screenshot illustrates the Stage Bound platform's initial development setup. A project folder called stage bound platform was created to contain the essential files needed to start development, which included HTML files for the login and dashboard interfaces, a CSS file for styling, and a JavaScript file for future functionality. This framework marks the beginning of the practical implementation phase and serves as the foundation for constructing the web-based platform.

Screenshot 2 — Login Interface Implementation



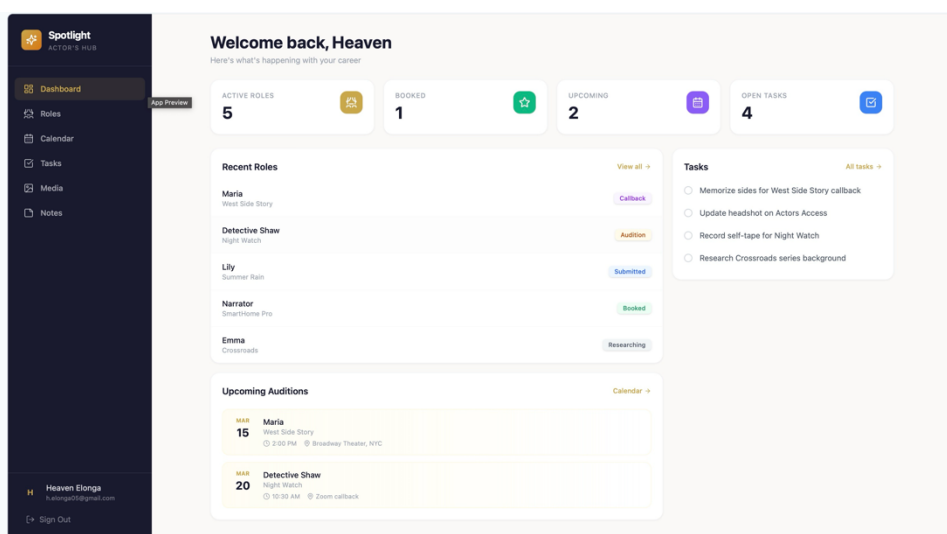
This screenshot shows the Stage Bound platform's login interface in its first implementation. The page features email and password input fields as well as a sign-in button. This feature meets the system's requirement for user authentication and serves as the platform's entry point.

Screenshot 3 — Dashboard Layout



A basic dashboard interface was created to display important information to the user. The layout was influenced by the Base44 prototype design. The dashboard displays summary cards for active parts, booked roles, future auditions, and open assignments to give the user a fast overview of their current schedules.

Screenshot 4 — Base44 Prototype Reference



This screenshot depicts the previous Base44 prototype, which served as a visual reference for the dashboard design. Base44 was used in the early stages of the project to quickly produce a layout and experiment with how the platform interface could be set up. The prototype helped in the design of the final dashboard, particularly the usage of summary cards and divided sections for positions, tasks, and auditions. During the initial development stage, elements from this prototype were reproduced in HTML and CSS to begin developing the platform's functioning version.

These implementation examples highlight the platform's early development and how the productivity tools for managing roles, auditions, and tasks may be used in practice. While cybersecurity elements such as fraud detection and protection methods were discovered during the project design phase, they were not fully integrated at this time and would be expanded further in future platform versions.