

“Implementation of VSARP: An AI-Powered Student Academic Resource and Placement Readiness Platform”

Sanjyot Khartade *1, Hrishikesh Kunjir *2, Anirudh Kondhalkar *3, Vinayak Lambate *4

Computer Engineering Department, Savitribai Phule Pune University Pune, India

ABSTRACT

VSARP (Virtual Student Academic Resource Portal) is a multi-role academic workflow platform that connects student evidence management, faculty verification, employability analytics, and placement readiness into a single web application. The present implementation combines a React and Vite interface, role-based dashboards, deterministic audit logging, public verification through integrity hashes, seeded demonstration data, and a Supabase-ready persistence model with a localStorage fallback for low-friction deployment and demos. The system is designed to reduce fragmentation in institutional records, improve traceability of approved student achievements, and convert verified academic and co-curricular records into reusable readiness signals. This paper presents the VSARP implementation in compact journal form, documents what is fully implemented in the repository, distinguishes heuristic from production-ready modules, and outlines how the current architecture supports students, faculty reviewers, HOD-level monitoring, placement operations, and administrative governance.

Keywords: student portfolio verification, placement readiness, audit logging, academic activity tracking, Supabase, React, employability analytics, institutional workflow.

I. INTRODUCTION

Higher-education institutions often manage student achievements, semester outcomes, placement preparation artifacts, and faculty approvals across disconnected files, spreadsheets, and informal communication channels. This weakens traceability, slows verification, and prevents approved records from being reused for institutional analytics or placement readiness decisions. VSARP addresses this fragmentation by treating validated academic and co-curricular evidence as structured institutional data rather than isolated submissions.

The current repository demonstrates a working system with role-based routes for students, faculty, HODs, placement-cell users, and administrators. It includes activity submission, faculty approval flows, semester-result verification, public verification pages, audit-log capture, placement-drive matching, aptitude support, and an AI copilot route. Some career-analysis functionality remains heuristic, but the broader workflow backbone is coherent and implementation-ready.

The implementation is therefore positioned as a practical institutional platform rather than a narrow proof-of-concept. Its main contribution lies in connecting approval workflows to reusable downstream outcomes such as public verification, placement filtering, student readiness indicators, and administrative reporting.

II. RELATED WORK

A. Student Record Management

Conventional student record systems capture attendance, grades, or registration status, but they rarely unify verified co-curricular evidence, public verification, and placement intelligence within one reusable workflow. This leaves institutions with partial visibility into student readiness and forces placement teams to rely on manual validation.

B. Placement and Employability Systems

Placement portals commonly store openings and applications, yet many do not compute readiness from verified academic signals, approved activities, aptitude attempts, and extracted skill tags together. VSARP improves this by connecting eligibility logic directly to institutional records rather than student self-declaration alone.

C. Gap Analysis

Feature	Typical Portal	VSARP
Activity verification	Basic submission tracking	Faculty approval workflow with auditable status changes
Public authenticity check	Often unavailable	Hash-based verification pages for approved records
Placement matching	Drive listing only	CGPA, skills, deadlines, and department-aware matching
Deployment mode	Backend-dependent	Supabase-ready plus local fallback demo mode

Table 1: Gap Analysis Between Typical Student Portals and VSARP

III. METHODOLOGY

A. Role-Driven Workflow Model

VSARP follows a role-driven methodology in which students create records, faculty validate them, the system seals approved states through hashes and logs, and downstream institutional modules reuse the validated data. This design keeps every major workflow anchored to a consistent data lifecycle rather than independent feature silos.

B. Verification and Audit Model

Approved activities and verified semester results can be exposed through public verification routes. The repository also includes audit-log recording for critical actions, allowing administrators to inspect who changed what and when. This methodology turns routine approval activity into a traceable governance layer.

C. Readiness Analytics Strategy

Employability scoring is computed from approved activities using weighted categories, caps, recency effects, internship-type differentiation, and aptitude percentile bonuses. Placement matching then combines this evidence with verified CGPA, profile skills, approved skill tags, eligible departments, and application deadlines. The methodology therefore favors defensible readiness signals over simple activity counts.

D. Demonstration Data Method

The system includes a deterministic seed-data strategy for presentation and testing. Demo users are generated across student, faculty, HOD, placement-cell, and admin roles. Approved and pending activities, research papers, placement drives, semester results, courses, and aptitude attempts are pre-populated so that every major route can be demonstrated without dependency on a live backend instance. This method improves repeatability and allows feature validation in classroom, review, and project-evaluation settings.

E. Constraints and Assumptions

The methodology assumes that faculty approvals and verified result imports remain the trust anchor for institutional correctness. It also assumes a dual-mode deployment model in which localStorage is acceptable for demos while Supabase-backed persistence is available for production use. Current career recommendation and skill-gap inference are acknowledged as heuristic and should not be interpreted as mature machine learning outputs.

Dataset area	Repository-backed coverage
Users	Students, faculty, HOD, placement-cell, and admin identities are seeded for realistic walkthroughs
Activities	Approved and pending activity records support review, analytics, and portfolio screens
Academic results	Verified and pending result records support CGPA and verification flows
Placement data	Placement drives, applications, and notifications support placement hub interactions
Aptitude analytics	Attempt data supports percentile logic and readiness bonuses

Table 2: Demonstration Data Method Used by the Current Implementation

IV. ARCHITECTURE

A. Architecture Overview

The architectural structure is organized into presentation layers, application services, and a dual-mode persistence backbone. Student, faculty, and institutional interfaces operate on shared service logic for scoring, verification, and placement workflows. Persistence switches between a Supabase-aligned schema and local browser storage depending on environment configuration.

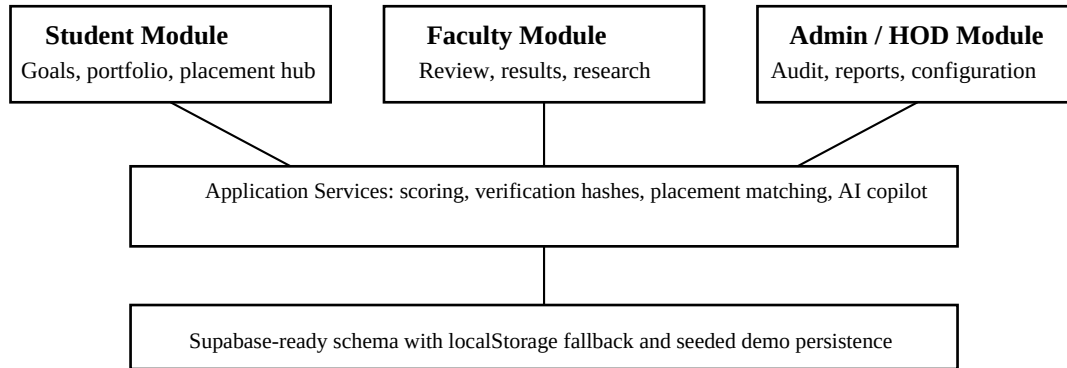


Figure 1: High-Level VSARP Architecture

B. Technology Stack

Layer	Technology	Implementation Role
Frontend	React + Vite	Role-based UI, routing, and interaction flows
Styling	Tailwind CSS	Responsive compact dashboard interface
Persistence	Supabase / localStorage	Production-ready schema path and demo fallback
Analytics	Local JS utility modules	Employability score, CGPA, skill extraction, drive fit
AI Route	NVIDIA chat completion API	Context-aware VSARP Copilot responses

Table 2: Technology Stack Overview

C. System Implementation and User Interface

The implementation includes student pages for dashboard access, activity submission, resume building, career-goal definition, career navigation, placement hub access, and academic result handling. Faculty modules support review queues, result verification, and research publication. Administrative and HOD modules extend the system into audit monitoring, reporting, student filtering, and configuration. The user interface is optimized for quick scanning and operational actions rather than long-form ERP complexity.

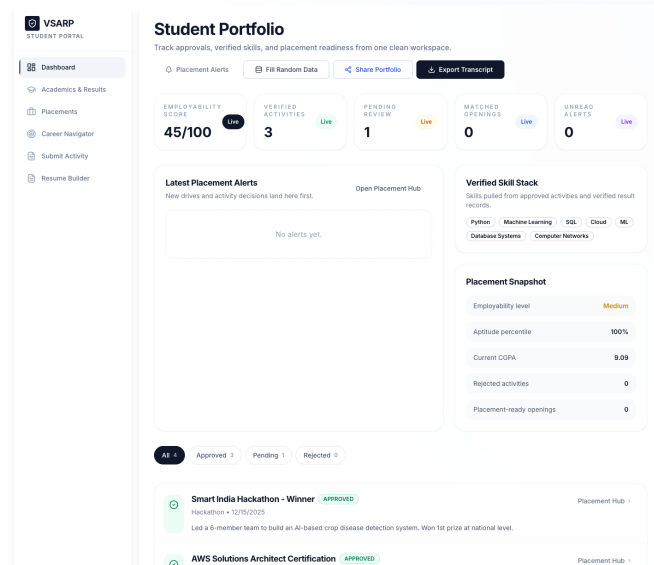


Figure 2: Student Dashboard Implementation Screenshot

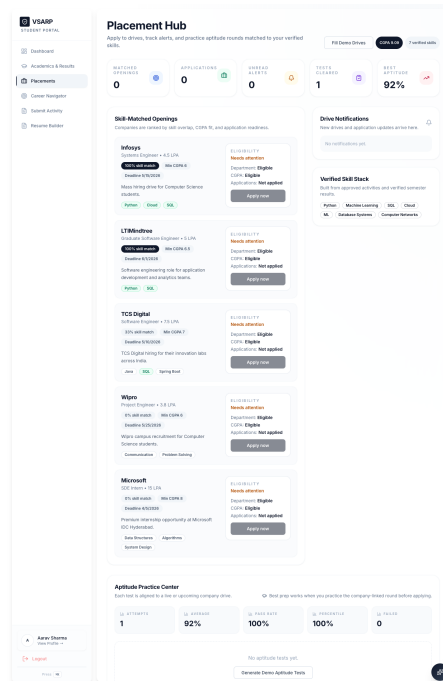


Figure 3: Student Placement Hub Implementation Screenshot

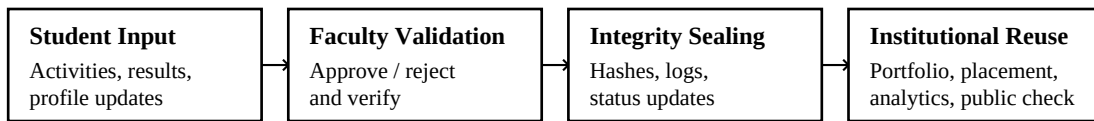


Figure 4: Core Submission-to-Reuse Workflow in VSARP

D. Data Model Overview

At schema level, VSARP is organized around users, activities, research papers, courses, semester results, placement drives, placement applications, notifications, aptitude tests, and audit logs. These entities are deliberately linked through role ownership, status transitions, and derived readiness logic. This schema is broad enough to support both individual student views and institution-wide oversight without duplicating the same information across separate subsystems.

Entity	Purpose in VSARP	Operational value
Users / profiles	Role identity, department, skills, and account state	Supports routing, authorization, and student filtering
Activities	Student co-curricular submissions and evidence	Feeds approval, scoring, portfolio, and verification flows
Semester results	Verified academic outcomes and hashes	Supports CGPA, authenticity, and readiness signals
Placement drives	Company opportunities, deadlines, criteria, and openings	Enables drive ranking and student fit evaluation
Audit logs	Critical event history for administrative review	Strengthens traceability and governance confidence

Table 3: Core Data Model Elements in VSARP

E. Verification and Security Flow

The verification route checks integrity hashes for approved activities and verification hashes for semester results. In mock mode it resolves records from localStorage; in production mode it uses structured database queries. This split design keeps the verification experience identical for the end user while allowing infrastructure differences beneath the interface. In addition, audit-log capture records high-value

actions such as approvals, rejections, and administrative operations.

F. Interface Composition

The student interface emphasizes dashboard summaries, activity submission, placement opportunities, and resume generation. Faculty screens are optimized for queue review, result verification, and publication workflows. Administrative screens prioritize audit views, reporting, and institutional monitoring. This division of concerns improves usability because each role is exposed primarily to the actions that matter to it, while all roles still operate on the same institutional dataset.

V. RESULTS AND DISCUSSION

The implemented system already demonstrates several concrete outcomes at repository level. First, it unifies activity evidence, academic verification, and placement readiness in one data pipeline. Second, it operationalizes trust through audit logs and public verification routes. Third, it converts approved evidence into reusable readiness indicators through scoring and placement matching. These outcomes make the implementation more institutionally useful than a standard CRUD portal.

Observed implementation area	Discussion
Faculty verification workflow	Pending activity and result queues are implemented and can transition records through approval or rejection actions
Placement intelligence	Drive ranking uses verified academic and skill signals rather than generic listing behavior
Demo readiness	Seeded users, activities, placement drives, research papers, and aptitude attempts allow full walkthroughs without infrastructure setup
AI maturity	Copilot routing is implemented, while some career recommendation features remain heuristic or mock-driven

Table 4: Results and Discussion Summary

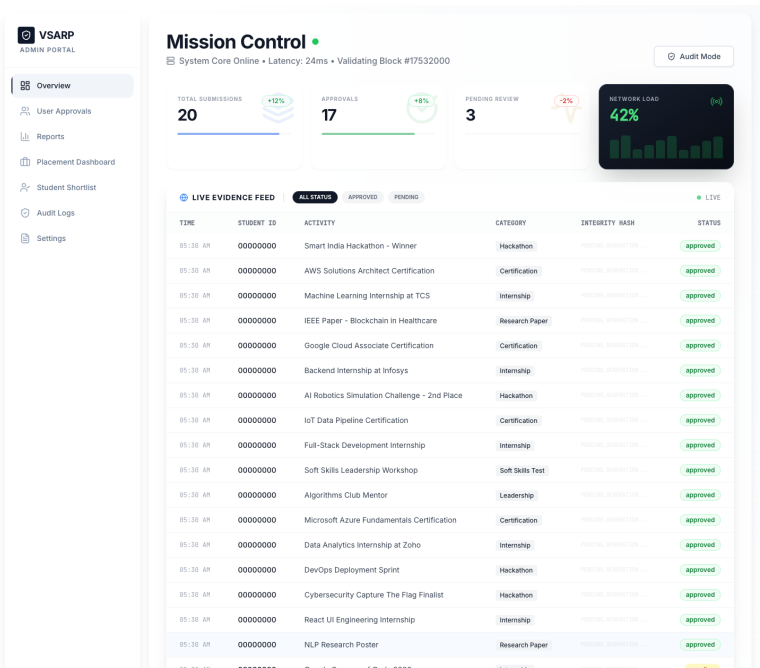


Figure 5: Administrative Overview Implementation Screenshot

A. Functional Maturity Review

Module	Status	Remarks
Activity submission and review	Implemented	Supports pending, approved, and rejected flows
Result verification	Implemented	Pending result queues and verified result paths are available
Public verification page	Implemented	Approved records can be resolved through verification hashes
Placement hub	Implemented	Uses skills, CGPA, deadlines, and applications
Career recommendation layer	Partial / heuristic	Current logic is mock-driven and rule-based

AI copilot	Implemented	Depends on configured external API credentials
------------	-------------	--

Table 5: Functional Maturity Across Major VSARP Modules

B. Readiness Interpretation

From a project-evaluation perspective, VSARP is strongest where workflow integration matters: verification, data reuse, and role-based operations. It is moderate in areas where intelligence depth still depends on future enhancement, especially recommendation quality and long-term production telemetry. This distinction is useful because it prevents the implementation from being overclaimed while still demonstrating strong technical breadth.

Readiness dimension	Assessment	Interpretation
Demo readiness	High	Seeded users and data allow immediate project walkthroughs
Institutional workflow readiness	High	Approval, verification, and placement workflows are coherent
Production backend readiness	Moderate	Schema exists but live deployment depends on configuration and operational hardening
AI decision maturity	Moderate	Copilot exists, but career inference remains heuristic

Table 6: Implementation Readiness Interpretation

C. Discussion of Practical Use

In practical use, VSARP can reduce duplicate verification effort because one approved activity record can later serve portfolio display, employability scoring, placement matching, and authenticity validation simultaneously. Similarly, verified semester results can support academic transparency, CGPA calculation, and placement filtering without repeated manual data entry. This reuse-driven design is a major strength of the implementation.

The main limitation is not the workflow shell but the maturity of some intelligence layers. Career suggestion and skill-gap logic are currently simple and should not be overstated as production machine learning. Similarly, local fallback mode is excellent for demos but should not be treated as authoritative multi-user persistence. Even so, the implementation already supports a strong institutional demonstration and a credible transition path to production deployment.

D. Limitations

Constraint	Current effect
Heuristic career module	Career guidance is functional but not yet backed by advanced inference or live market data
Mock-mode persistence	Appropriate for demos, but not for authoritative multi-user institutional deployment
Operational telemetry	Latency, adoption, and long-term outcome analytics are not yet tracked in production form
Proof-document lifecycle	Evidence references exist, but stronger document storage and anti-fraud controls remain future work

Table 7: Present Limitations of the Current Implementation

VI. FUTURE ENHANCEMENTS

Enhancement	Expected benefit
Production-grade career recommendation engine	Move beyond heuristic suggestions toward richer skill-gap and career-path inference
Storage-backed proof management	Strengthen document authenticity and lifecycle control for uploaded evidence
Institutional outcome reporting	Support accreditation, NAAC-style summaries, and department-level readiness exports
Recruiter-facing verification surfaces	Allow external stakeholders to validate records and shortlist readiness signals faster
Expanded analytics telemetry	Track adoption, review latency, placement conversion, and operational bottlenecks

Table 8: Future Enhancement Roadmap

Additional work can also connect institution-wide accreditation and recruiter reporting to the same validated data foundation. Because the core model already links verified evidence to analytics and placement logic, future layers can be added incrementally rather than through full architectural redesign.

VII. CONCLUSION

VSARP presents a compact but technically meaningful implementation of a student academic resource and placement-readiness platform. The system's core strength lies in reusing approved institutional data across verification, analytics, and placement workflows. The repository already contains a coherent verification model, role-based interfaces, placement intelligence, seeded demo coverage, and a production-aligned schema path. With stronger recommendation logic and storage hardening, the implementation can evolve from a presentation-ready institutional system into a deeper deployment-grade academic operations platform.

VIII. REFERENCES

- [1] VSARP Repository README, project workspace documentation, 2026.
- [2] VSARP Product Requirements Document (PRD), internal repository artifact, 2026.
- [3] React Documentation, Meta Open Source, referenced for component-based application design.
- [4] Vite Documentation, referenced for frontend build and routing configuration context.
- [5] Supabase Documentation, referenced for schema and persistence integration context.
- [6] Tailwind CSS Documentation, referenced for responsive UI implementation patterns.
- [7] NVIDIA Chat Completions API documentation, referenced for VSARP Copilot route design.
- [8] Repository modules for employability scoring, placement matching, and verification utilities, inspected June 2026.