Full-Stack Dating App Prototype

IEEE-Style Project Documentation and Case Study

Muhammad Haider Abbas | MSc Computer Science & Technology | Ulster University Birmingham Campus



Figure 1. Project visual identity used in the portfolio website.

Abstract

Full-Stack Dating App Prototype is documented as a professional portfolio case study by Muhammad Haider Abbas. The project demonstrates practical capability in Web Systems through a structured problem-solution design, implementation direction, testing plan and future development roadmap. The report explains the business or academic motivation, system requirements, architecture, workflow, quality considerations and expected professional value of the project. It is prepared for use as downloadable portfolio evidence for recruiters, clients and academic reviewers.

Keywords: Web Systems, Prototype, Full-Stack Developer, React, Node, Prisma, PostgreSQL

I. Project Profile

Field	Details
Project Title	Full-Stack Dating App Prototype
Category	Web Systems
Year	2026
Role	Full-Stack Developer
Status	Prototype
Tools	React, Node.js, Prisma, PostgreSQL
Portfolio Slug	full-stack-dating-app-prototype

II. Introduction and Problem Context

This document presents the technical and professional documentation for Full-Stack Dating App Prototype. The project is positioned as a prototype in the Web Systems category. The work highlights the author's ability to combine technical implementation, documentation discipline and outcome-focused project thinking. The project summary is: A full-stack app prototype with authentication direction, profile flow, matching concept and database schema planning.

A. Problem Statement

Modern dating apps require secure authentication, clean data modelling and smooth profile/matching flows.

B. Proposed Solution

The prototype plans frontend, backend, Prisma schema and PostgreSQL database structure.

III. Aim and Objectives

To design a modern full-stack dating app prototype with authentication, profile management, matching concepts and relational database planning.

- Build user registration and login direction.
- Design profile, interest and preference data models.
- Plan matching and discovery workflow.
- Prepare messaging and moderation direction.
- Document privacy and safety controls.

IV. Requirements Analysis

ID	Requirement	Type	Priority
FR-01	React frontend direction	Functional / project requirement	Required
FR-02	Node backend API	Functional / project requirement	Required
FR-03	Prisma schema	Functional / project requirement	Required
FR-04	PostgreSQL database	Functional / project requirement	Required
FR-05	Authentication flow	Functional / project requirement	Required
FR-06	Profiles and preferences	Functional / project requirement	Required
FR-07	Matching/messaging design	Functional / project requirement	Required
NFR-01	Usability and clear portfolio presentation	Non-functional	Required
NFR-02	Maintainability and future extensibility	Non-functional	Required
NFR-03	Professional documentation and evidence quality	Non-functional	Required

V. System Architecture

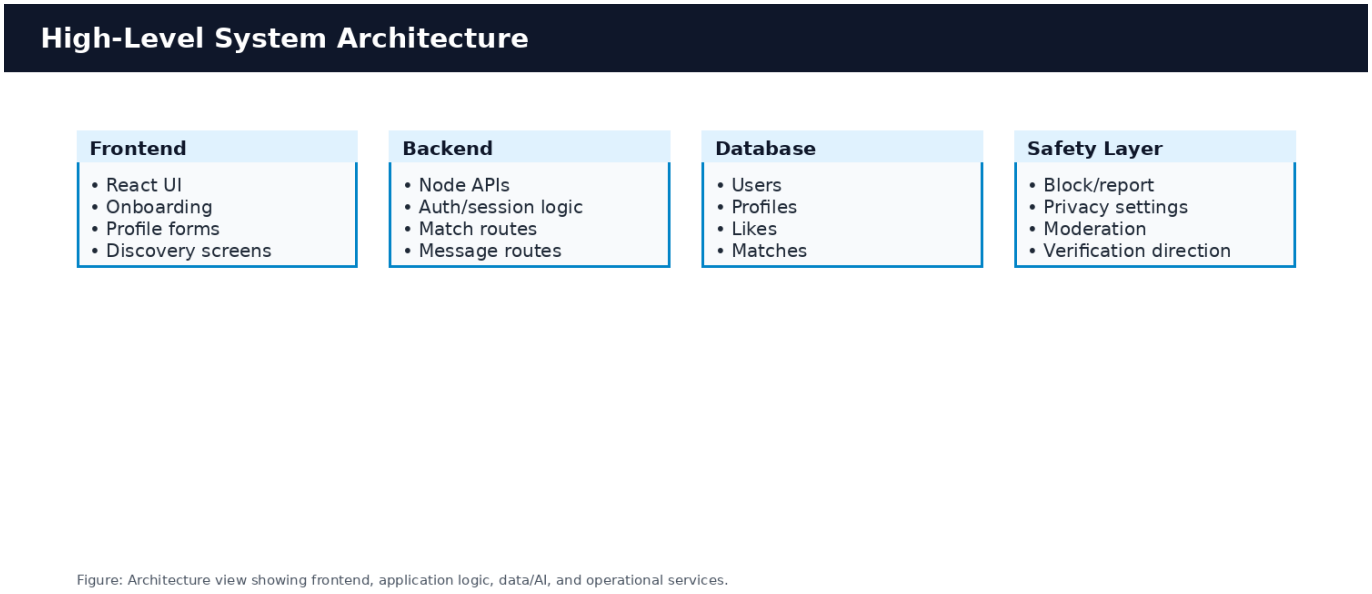


Figure 2. High-level architecture or production pipeline for the project.

The architecture separates user-facing presentation, project logic, data or model processing, and operational output. This structure makes the project easier to explain, test and extend inside a professional portfolio environment.

VI. Workflow Design

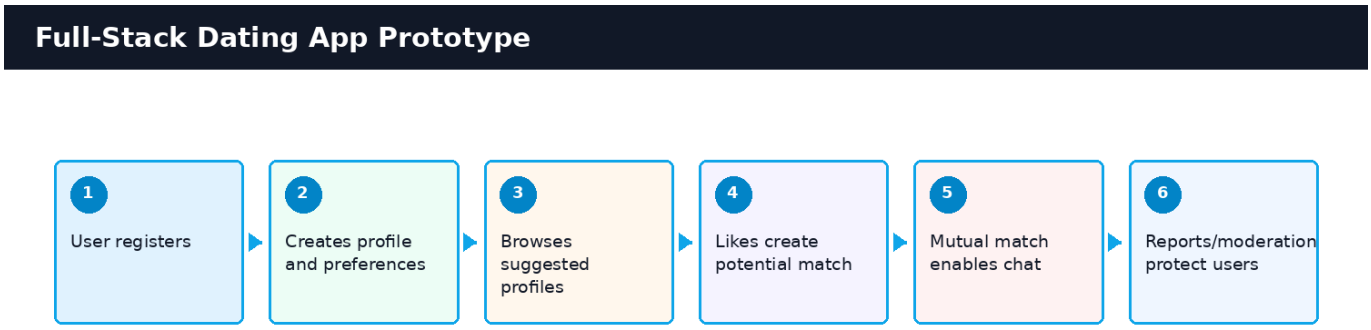


Figure: Generated project workflow / architecture artefact for portfolio documentation.

Figure 3. Project workflow from input or user action to final output.

VII. Data, Components and Implementation View

Component	Layer	Responsibilities	Notes
C-01	Frontend	React UI, Onboarding, Profile forms, Discovery screens	Core component group
C-02	Backend	Node APIs, Auth/session logic, Match routes, Message routes	Core component group
C-03	Database	Users, Profiles, Likes, Matches, Messages	Core component group
C-04	Safety Layer	Block/report, Privacy settings, Moderation, Verification direction	Core component group

Implementation should be documented through screenshots, code snippets, database schema extracts, test outputs or demonstration media when available. For deployed or live systems, screenshots and logs should be attached as appendices to increase evidence strength.

VIII. Testing and Validation Strategy

Test Area	Test Method	Expected Result
Registration	Create test account	User record created
Profile form	Save profile fields	Data persists
Matching logic	Mutual like simulation	Match record created
Privacy controls	Block/report action	Interaction restricted

The testing strategy focuses on validating the main user or system journey, confirming that core requirements behave as expected, and recording limitations clearly. This supports honest portfolio evidence and avoids overclaiming features that are still prototypes.

IX. Security, Ethics and Professional Considerations

- Validate user input and uploads.
- Use secure payment flows and avoid storing card data directly.
- Follow privacy principles and minimise personal data collection.

X. Results, Outcome and Future Work

A. Outcome

A useful full-stack learning and product design project.

B. Future Enhancements

Add working auth, chat, profile verification and recommendations.

Roadmap Stage	Enhancement Direction
Short term	Add missing screenshots, demo links and downloadable supporting files.
Medium term	Connect live/source URLs and replace placeholder evidence with real artefacts.
Long term	Use analytics, user feedback or formal evaluation results to strengthen the project case study.

XI. References

[1] React, "React documentation," https://react.dev/

[2] OpenJS Foundation, "About Node.js," https://nodejs.org/en/about

[3] Prisma, "What is Prisma ORM?" https://www.prisma.io/docs/orm/overview/introduction/what-is-prisma

[4] PostgreSQL Global Development Group, "What is PostgreSQL?" https://www.postgresql.org/docs/current/intro-whatis.html

Appendix A. Portfolio Download Mapping

Item	Value
Recommended PDF path	docs/full-stack-dating-app-prototype-project-documentation.pdf
Recommended Word path	docs/full-stack-dating-app-prototype-project-documentation.docx
Admin field: Documentation	docs/full-stack-dating-app-prototype-project-documentation.pdf
Admin field: Extra Project File	docs/full-stack-dating-app-prototype-project-documentation.docx