OTIS: Digitalizing Tourist Information Management for the Department of Tourism, Government of Puducherry UT — An E-Governance Web Application

Mr. P.Rajapandian[1], Kaviya. M[2]

[1]*Associate Professor, Department of MCA, Sri Manakula Vinayagar Engineering College(Autonomous), Madagadipet, Puducherry, India*

[2]*Post Graduate Student, Department of MCA, Sri Manakula Vinayagar Engineering College (Autonomous), Madagadipet, Puducherry 605 107, India | Reg. No.: 24PMC062*

Abstract: This paper presents the design, development, and deployment of the Online Tourist Information System (OTIS), a full-stack e-governance web application built during a four-month internship at the National Informatics Centre (NIC), Pondicherry. Tourism management in Pondicherry has long relied on manual, paper-based processes — hotels maintaining physical visitor registers, departments compiling records by hand, and compliance reports sent to the Ministry of Home Affairs (MHA) through slow, error-prone workflows. OTIS completely replaces this fragmented system with a centralised, role-based digital platform. Built on ASP.NET Core MVC (.NET 10) with C#, dual SQL Server connections, Bootstrap 5, and QuestPDF, the system supports five user roles — Department Admin, Department Operator, Hotel Admin, Hotel Operator, and Public/MIS User — each with strictly scoped permissions. Key features include SMS OTP hotel verification, BCrypt password hashing, offline CAPTCHA, cookie-based authentication, automated daily MHA report dispatch, an AI-powered public chatbot, a feedback and rating system, and a full audit trail with IP address logging. Testing confirmed resistance to SQL injection, session hijacking, and OTP replay attacks. OTIS provides a replicable, production-ready e-governance model for tourism digitisation across Indian states and Union Territories.

Keywords: E-Governance, Online Tourist Information System, ASP.NET Core MVC, Role-Based Access Control, SQL Server, QuestPDF, OTP Verification, BCrypt, AI Chatbot, Digital Tourism, NIC Pondicherry, Automated Reporting, MHA Compliance, Audit Trail, Cookie Authentication

---

Introduction:

These days, if a tourist wants to find a hotel in Pondicherry, they just search online and see what comes up. But behind that search lies a government department still running on physical registers, manual inspections, and paper-compiled reports sent by hand to the Ministry of Home Affairs. For a heritage destination like Pondicherry — drawing both domestic and international visitors to its French colonial streets and coastal culture — this gap between tourist expectations and administrative reality is significant.

The situation isn't unique to tourism. Across India's states and Union Territories, local government departments manage visitor and hospitality data through disconnected, manual workflows. Hotels keep their own registers. Department officers periodically collect them. Reports get assembled monthly, often late, sometimes inaccurate, always resource-intensive. And nowhere in this chain is there a public portal where a tourist can browse verified hotels, read genuine reviews, or get answers to their questions.

OTIS changes this entirely. Developed as a four-month internship project at the National Informatics Centre (NIC), Pondicherry, under the guidance of Mr. S. Arulraj, Sr. Technical Director (IT), and internal guide Mr. Rajapandian, Associate Professor, Department of MCA, SMVEC, the Online Tourist Information System brings the full tourism

management workflow online — hotel registration with OTP verification, structured tourist data entry, automated government compliance reporting, and a public-facing portal with an AI chatbot.

This paper keeps it practical. Section by section, it lays out what was wrong with the existing system, what OTIS proposes, how it was built, how it performs, and where it can go next. The goal is not just a working application — it is a replicable model that any Indian state or UT can adapt for their own tourism department.

Literature Survey:

E-governance never stands still. Digital platforms for government service delivery keep evolving, and just putting a department online is never enough. You have to think through workflows, access control, data security, and reporting — because the moment one piece fails, the whole system loses trust.

ASP.NET Core MVC documentation from Microsoft (2024) provided the architectural foundation. The MVC pattern — separating Models, Views, and Controllers — keeps large applications maintainable, especially when multiple user roles interact with overlapping data. Entity Framework Core (Microsoft, 2024) guided code-first migrations for the authentication database, making schema evolution clean and traceable. For the operational tourist and hotel data, ADO.NET was preferred: direct parameterised queries offer tighter control and clearer performance characteristics for high-frequency insert and read operations.

Security literature shaped the authentication layer significantly. BCrypt's adaptive hashing (Provos & Mazières, 1999) makes brute-force attacks computationally expensive in a way that MD5 or SHA1 simply cannot. Research into session management best practices informed the cookie-based authentication design — HttpOnly, encrypted, with sliding expiration — preventing the session-fixation and CSRF vulnerabilities common in web applications.

The Clean Architecture guide by Microsoft (2024) reinforced separation of concerns across layers. QuestPDF documentation (2024) demonstrated how modern .NET applications can generate professional, print-ready PDFs without commercial licensing costs — critical in a government context where open-source or free tools are strongly preferred. Studies on Indian e-governance deployments, including NIC's own documented experiences in MeitY annual reports, validated the five-role access model and highlighted the importance of full audit trails for government accountability. Local tourism portal research confirmed that centralised, verifiable hotel listings with public ratings significantly improve tourist engagement and booking decisions.

Proposed System:

Think of OTIS as more than just a website for the Tourism Department — it is a complete digital backbone for how Pondicherry manages its hotels, tracks its tourists, and reports to the central government. The whole system is built around five roles, each with their own interface and their own slice of responsibility, so nothing overlaps and nothing gets missed.

Everything starts with hotel registration. A hotel owner visits the portal, fills in their details, and receives an OTP via SMS to verify their mobile number. No manual inspection visit. No paper forms. Once verified, the application sits with the Department Admin for review and approval — a clean, auditable workflow that eliminates the fraudulent registrations that plagued the old system. Approved hotels appear on the public portal automatically.

Once a hotel is live, the Hotel Operator logs in daily to enter tourist information — domestic visitors with their state, district, and purpose of visit; international visitors with their country, passport details, and visa type. For busy periods, a bulk import feature accepts structured Excel files, cutting data entry time dramatically. Every entry is timestamped and logged with the operator's IP address, building an unbroken audit trail.

On the department side, operators get dashboards showing tourist arrivals across all hotels — by date, by district, by nationality. QuestPDF generates professional daily and monthly PDF reports on demand, ready for printing or sharing. A scheduled background service runs every night, consolidating the day's data and automatically

dispatching the consolidated report to the Ministry of Home Affairs endpoint — no manual step, no risk of a forgotten submission.

The public portal needs no login. Tourists browse verified hotels, read star ratings and written reviews left by other visitors, and interact with an AI-powered chatbot that answers questions about hotels, attractions, and how to navigate the portal. The chatbot pulls from live OTIS data, so its answers are always current. This is what replaces the scattered, unverifiable information tourists currently piece together from unofficial sources.

System Architecture:

Figure 1 makes the structure clear — OTIS is not a single-purpose tool but a layered platform with distinct modules feeding into a common data and reporting core. It works from a full architecture split into five clear layers: Authentication & Security, Hotel Management, Tourist Data Management, Reporting & Compliance, and the Public Portal. Every user interaction passes through the authentication layer first.

[Fig 1: System Architecture]

System Architecture

The authentication layer handles all login flows — BCrypt password verification, offline CAPTCHA validation, OTP confirmation for hotel registration, failed-attempt counting, and automatic account locking. Cookie-based sessions carry role claims forward to every subsequent request, so the Hotel Management and Tourist Data modules never need to re-check credentials.

Two separate SQL Server databases sit at the data layer. The authentication database — managed by Entity Framework Core migrations on SQL Server LocalDB — holds user accounts, password hashes, login history, and lock status. The operational TouristInfo database — accessed via ADO.NET on SQL Server Express — holds all hotel details, tourist records, feedback, and master reference tables (Country, State, District, Hotel Type). Keeping these separate means authentication never competes with high-frequency tourist data writes during peak season.

The Reporting & Compliance module sits above both databases. QuestPDF pulls aggregated tourist data and formats it into print-ready PDFs — daily and monthly, hotel-level and department-level. The MHA dispatch scheduler runs independently of user sessions, ensuring government submissions happen even if no operator is logged in at midnight.

Table 1: User Roles and System Access

User Role	Access Level	Key Modules	Reporting
Department Admin	Full System Control	Hotel verification, user mgmt.	All reports
Department Operator	Department-wide	Tourist data review, dashboards	Dept. level
Hotel Admin	Hotel-wide	Facility mgmt., hotel profile	Hotel level
Hotel Operator	Data Entry	Tourist entry, bulk import	Entry logs
Public / MIS User	Read-only (no login)	Browse hotels, AI chatbot	None

Figure 2 traces the data flow from a hotel operator's daily tourist entry through to the automated MHA report dispatch — showing how each input is validated, stored, aggregated, and delivered without manual intervention.

[Fig 2: Data Flow Diagram]

Data Flow Diagram

Table 2: Development Phases and Deliverables

Phase	Module	Timeline	Outcome
Phase 1	Auth & Login	Week 1–2	Secure login, OTP verification, account locking
Phase 2	Hotel Management	Week 3–4	Hotel registration, approval, facility details
Phase 3	Tourist Data Entry	Week 5–6	Domestic/international entry, bulk import
Phase 4	Reports & MHA Dispatch	Week 7–8	QuestPDF reports, automated daily MHA email
Phase 5	AI Chatbot & Public Portal	Week 9–10	AI chatbot, hotel listings, feedback & rating
Phase 6	Testing & Deployment	Week 11–12	Unit, integration, security, UAT; IIS deployment

Want your department's tourism system to actually work? Skip the generic portals and build around what each role genuinely needs. Whether it's a hotel operator entering data on a phone or a department admin generating compliance reports at midnight, these focused approaches get real results.

Role-Specific System Approach:

1. Department Admin (Full Control)

The admin gets everything. Approve or reject hotel registrations with a single click. Lock compromised accounts, manage master reference data — hotel types, districts, countries — and pull system-wide reports. The dashboard shows live counts: hotels registered, hotels approved, tourist entries today. Every action is logged with timestamp and IP, so there is always a paper trail.

2. Department Operator (Oversight & Reporting)

The operator does not manage users, but they see all the data. Filter tourist arrivals by date, district, nationality, or hotel. Generate daily and monthly PDF reports instantly via QuestPDF. Check whether the MHA dispatch scheduler delivered last night's report. If something looks wrong in the numbers, flag it — the audit trail shows exactly who entered what.

3. Hotel Admin (Registration & Setup)

Hotel registration starts here. Enter hotel details, verify the mobile number via OTP, and wait for department approval. Once approved, update facility information — room types, amenities, contact details — so the public portal listing is always accurate. View hotel-level reports to see how many tourists checked in this month.

4. Hotel Operator (Daily Data Entry)

This is where tourist data enters the system. Log in, select the date, enter domestic visitor details — name, state, district, purpose, duration. Switch to international entry for foreign guests — country, passport number, visa type. For large groups or bus tours, use the Excel bulk import. Every row gets saved instantly and appears in the department dashboard within seconds.

5. Public User (Browse & Interact, No Login Needed)

No account required. Search the verified hotel listings, filter by type or location, read star ratings and written reviews from other tourists. Not sure what to visit or which hotel suits your budget? Ask the AI chatbot — it pulls live data from OTIS to answer tourism questions, suggest hotels, and guide you through the portal. Simple, fast, and always current.

Modular Deployment Strategy:

OTIS does not need to go live all at once. Every tourism department has a different starting point — some have partial digital records, others are starting from zero. The modular architecture supports phased deployment: begin with hotel registration and authentication, then add tourist data entry, then activate the reporting layer, and finally switch on the public portal and AI chatbot. Each phase delivers standalone value, so the department sees results from week one rather than waiting for a full rollout.

Audit and Compliance by Design:

Government systems live or die on accountability. OTIS logs every transaction — every login attempt (successful or failed), every tourist record created or edited, every report generated, every MHA dispatch — with the operator's identity, timestamp, and IP address. This is not an afterthought. It is baked into the data model from the start, so administrators always know who did what and when. This audit trail is also what makes the system defensible if data discrepancies or compliance questions arise.

Scalability Across States and UTs:

The system is designed with the NIC context in mind — IIS-based deployment, SQL Server infrastructure, and .NET on Windows servers. This means any other Indian state or UT using NIC's standard stack can adopt OTIS with minimal infrastructure changes. The master reference tables (State, District, Country, Hotel Type) are configurable, so the system adapts to any geography. Multilingual support for Tamil, Telugu, and other regional languages is architecturally straightforward to add in a future release.

Conclusion and Future Enhancement:

The way OTIS approaches tourism management helps digitise every step of the process — hotel registration, tourist data collection, government compliance reporting, and public information access — replacing the fragmented paper-based workflows that have slowed Pondicherry's tourism administration for years. The role-based architecture is a big deal: five distinct roles mean every user sees exactly what they need and nothing they don't. Security is serious — BCrypt hashing, OTP verification, offline CAPTCHA, session management, and a full audit trail make the system defensible in a government context. The QuestPDF integration fits perfectly since reports need to be print-ready and professional, without licensing costs. Automated MHA dispatch means compliance deadlines are met even when staff are unavailable. Local relevance matters too — verified hotel listings with real ratings give Pondicherry's tourists a reliable starting point.

Analytics and monitoring make a difference going forward. Tracking which hotels are most viewed on the public portal, which tourist data entries trigger validation errors most often, and how often the AI chatbot is consulted gives the department real signals for improvement — data that the old paper system could never provide. It's not

expensive to scale up as Pondicherry's tourism grows; the modular architecture handles increasing hotel counts and tourist volumes without major changes.

In the future, a mobile application for hotel operators would allow offline tourist entry with background sync — useful during connectivity outages in smaller properties. GIS-based attraction mapping integrated into the public portal would significantly improve the tourist discovery experience. Power BI or Grafana dashboards for real-time tourism analytics could give department officials live insight into visitor patterns. Aadhaar-based tourist verification would improve data accuracy for international compliance. Voice search and multilingual support — especially Tamil and French given Pondicherry's heritage — would make the public portal accessible to a far broader audience. Chatbot enhancements using retrieval-augmented generation could make tourist assistance far more sophisticated. All of this would make the system more adaptive and more valuable over time. OTIS supports not just Pondicherry's tourism growth — it provides a model that any Indian UT or state can replicate through NIC's deployment infrastructure.

References

1. Microsoft Corporation. (2024). ASP.NET Core MVC Documentation. Microsoft Docs. https://learn.microsoft.com/en-us/aspnet/core/

2. Microsoft Corporation. (2024). Entity Framework Core Documentation. Microsoft Docs. https://learn.microsoft.com/en-us/ef/core/

3. Microsoft Corporation. (2024). Common Web Application Architectures. Azure Architecture Guide. https://learn.microsoft.com/en-us/dotnet/architecture/modern-web-apps-azure/

4. QuestPDF. (2024). QuestPDF Open-Source PDF Generation Library. https://www.questpdf.com/

5. National Informatics Centre. (2024). NIC E-Governance Solutions. MeitY, Government of India. https://www.nic.in/

6. Provos, N., & Mazières, D. (1999). A Future-Adaptable Password Scheme. USENIX Annual Technical Conference. BCrypt adaptive hashing for secure password storage.

7. Department of Administrative Reforms & Public Grievances. (2023). Framework for E-Governance in India. DARPG, Government of India. https://darpg.gov.in/

8. Singh, A., & Rao, N. (2020). Technical Feasibility and Security in Government Web Applications. Journal of E-Governance, 43(2), 112–128.

9. Sharma, P., & Krishnamurthy, R. (2022). Role-Based Access Control in Multi-Tier Web Applications. International Journal of Computer Applications, 184(12), 1–8.

10. Kumar, V., & Nair, S. (2023). Automated Government Compliance Reporting Using .NET Frameworks. Indian Journal of Computer Science and Engineering, 14(3), 55–67.

11. Rajan, T., & Subramanian, K. (2022). E-Governance Tourism Portals: A Study of Indian UT Deployments. Journal of Digital Governance, 9(1), 34–49.

12. Mehta, R., & Das, P. (2023). OTP-Based Verification Systems for Digital Hotel Registration in India. Proceedings of the National Conference on E-Governance, NIC, New Delhi.