Deployment and Operations

1. Introduction

The Infomate chatbot system is deployed as a web application accessible to students, faculty, new admissions, and guests. Deployment focuses on making the system reliable, scalable, and easily maintainable, while operations ensure that the chatbot remains up-to-date with the ICT Department’s latest information.

2. Deployment Environment

- Frontend (React): Deployed on a cloud hosting platform such as Vercel, which provides free SSL, CDN-based delivery, and automatic scaling for static web applications.
- Backend (Node.js): Deployed on Render Render offers easy setup, containerized deployments, and auto-restart in case of failure.
- Gemini API: Accessed via API key (environment variable). Requires secure storage of the key in backend environment configurations.
- Knowledge Base (PDF): Stored in the backend’s file storage. Updating departmental information only requires uploading a new PDF file.

3. Deployment Process

Step 1: Backend Deployment (Node.js)

1. Push backend code to GitHub.
2. Connect repository to Render.
3. Configure environment variables:
   - PORT → Backend server port
   - GEMINI_API_KEY → Secure Gemini API key
4. Deploy server and test /chat endpoint.

Step 2: Frontend Deployment (React)

1. Push frontend code to GitHub.
2. Deploy using Vercel by connecting the repo.
3. Configure backend API URL in environment variables.
4. Test chatbot UI against deployed backend.

4. Operations and Maintenance

1. PDF Updates (Knowledge Base):
   - Whenever the department updates faculty, placements, or curriculum, a new PDF is uploaded to the backend.
   - No code changes are required; chatbot responses automatically adapt.
2. Monitoring:
   - Use Render/Vercel dashboards to track uptime, request logs, and error rates.
   - Manual monitoring during events like admission season or placement drives to ensure performance.
3. Error Handling:
   - Backend logs all failed requests.
   - If API rate limits are hit, a fallback message is displayed to the user.
4. Scaling:
   - Frontend: Auto-scaled by Vercel CDN.
   - Backend: Can be scaled horizontally by adding more instances on Render.
   - API: If Gemini API quotas are exceeded, optimization or premium plan can be considered.
5. Security:
   - API keys stored securely in backend environment variables (not exposed to frontend).
   - Only departmental PDFs are used to prevent misinformation.

5. User Access and Operations

- Students / Admissions / Guests: Access Infomate via a browser link (e.g., https://infomateict.vercel.app). No installation required.
- Faculty/Staff (Operators): Responsible for updating PDFs and verifying chatbot accuracy.
- ICT Department Admin: Oversees performance, approves updates, and ensures chatbot information matches official data.

6. Conclusion

The deployment of Infomate leverages cloud-hosted infrastructure for reliability, accessibility, and scalability. Operations remain lightweight, with the only maintenance task being the periodic upload of updated departmental PDFs. This ensures the chatbot stays relevant while minimizing technical overhead, making Infomate a sustainable and user-friendly solution for the ICT Department.