# Chapter 1: Introduction

## 1.1 Background:

The InternQuest project was developed to address a specific and significant challenge faced by 12th-grade students in India: the extended period of inactivity after their final examinations. This gap, which can last for approximately four months, is a result of the time required for exam paper checking, result publication, and the subsequent college admission processes.

In today's highly competitive job market, companies increasingly seek candidates with practical skills and industry knowledge. Internships have become a crucial way for individuals to gain this experience and a valuable certificate. However, many 12th-grade students lack a fixed career goal and the necessary guidance to find such opportunities.

This project serves as a solution by providing a dedicated platform that connects these students with companies offering internships. The platform facilitates the acquisition of new skills and provides a valuable understanding of how companies and corporate systems function. By offering a structured path to gain experience during this transitional period, InternQuest empowers students to make a more informed decision about their future career and gives them a competitive advantage.

## 1.2 Objective:

The core objective of the InternQuest project is to develop a comprehensive platform that addresses the key problems identified in the background, namely, the lack of structured opportunities and guidance for 12th-pass students. This is achieved through a multi-faceted approach that serves the needs of both students and companies. The platform's primary goal is to transform the idle period after high school into a productive phase of skill acquisition and career exploration.

**The specific, measurable objectives of the InternQuest platform are:**

- **Student Empowerment**: To provide students with a centralized platform to find and apply for internships in various fields, helping them gain valuable practical experience and industry knowledge that can be converted into a valuable certificate.

- **Industry Collaboration:** To create a seamless interface for companies to post and manage internship opportunities and select suitable candidates from a pool of motivated students.

- **Streamlined Process:** To automate the internship application process through a user-friendly dashboard, including features for aptitude testing, application tracking, and document generation (such as offer letters, payment receipts, and certificates).

- **Skill Assessment:** To implement an aptitude test that evaluates a student's readiness and helps them identify suitable career paths, ensuring that students meet a minimum standard for internship eligibility.

- **Revenue Generation:** To implement a payment system for a nominal fee (□149) that students must pay to receive their official offer letter after being selected, thus ensuring a committed applicant base and supporting the platform's operations.

- **Administrative Oversight:** To establish an administrative dashboard for full control over the platform, enabling the management of users, internships, and payments, as well as the ability to update the website's content.

## 1.3 **Purpose**:

The fundamental purpose of the InternQuest project is to act as a crucial bridge, connecting aspiring students with the professional world. It is designed to solve the problem of a significant four-month period of inactivity that students face after completing their 12th-grade examinations.

The platform's primary goal is to **leverage this downtime** by providing a structured and productive way for students to learn new skills and gain practical experience. In a competitive job market where companies seek candidates with real-world knowledge, this early exposure is invaluable.

The key purposes of InternQuest are:

- **Career Exploration**: To assist students who may not have a fixed career goal by offering them the opportunity to explore different fields through a 15-day hands-on internship.
- **Skill Enhancement**: To provide a platform for students to acquire new skills and get industry-specific knowledge that is critical for surviving in today's job environment.
- **Experience & Validation**: To offer students a path to gain valuable experience and earn an official certificate, which is highly valued by companies during job applications.
- **Facilitating the System**: To provide a comprehensive, organized system for both students and companies to manage the entire internship process, from application to certification.

## 1.4 Scope:

The scope of the InternQuest project defines the boundaries and key functionalities of the system. It clarifies what features are included in the initial development and which are deliberately excluded for a later phase. This focused approach ensures the project's objectives are met efficiently and that the final product is both functional and stable.

---

**In-Scope Functionalities:**

The InternQuest platform is built as a complete, multi-user system with the following core features:

- **User Management**: The system supports three distinct user roles: Student, Company, and Admin. Each role has a dedicated dashboard with specific functionalities tailored to their needs.
- **Student Dashboard**: This includes features for profile creation and management, a mandatory aptitude test to assess eligibility, and the ability to browse and apply for available internships.
- **Company Dashboard**: Companies can manage their profiles and post new internship opportunities. They have a dedicated section to view, approve, or reject student applications, as well as mark an internship as completed.
- **Admin Dashboard**: The administrator has full control over the platform, including managing user accounts (students and companies), overseeing applications, adding/deleting aptitude test questions, and updating website content via a Content Management System (CMS).
- **Application & Document Flow**: The system automates the application process, from initial submission to company review. It also includes a simulated payment process of ▪149 for students who are selected for an internship. Upon payment, the platform generates a unique offer letter, and a certificate can be downloaded upon internship completion.

---

**Out-of-Scope Functionalities (Future Work):**

The current version of InternQuest focuses on core functionalities. The following features are considered outside the initial scope but are planned for future development to enhance the platform:

- **Integrated Payment Gateway**: The current payment process is a simulation. Future iterations will integrate with a real payment gateway (e.g., Stripe, Razorpay) to handle live transactions.
- **Advanced Analytics**: While the admin can see payments and user lists, the platform currently lacks advanced analytics dashboards with detailed metrics on user engagement, internship success rates, and other key performance indicators.
- **Communication Features**: Direct messaging or a chat function between students and companies is not included. Communication is handled outside the platform.
- **Recommendation Engine**: The platform does not currently recommend internships to students based on their profile or aptitude test scores. This would be a future enhancement to improve the user experience.
- **Automated Skill Assessment**: The aptitude test is a static set of questions. Future work could include dynamic or specialized skill tests tailored to specific fields.

# Chapter 2: System Analysis

## 2.1 Existing System:

The current process for 12th-pass students to find internships is largely unstructured and inefficient. Without a dedicated platform like InternQuest, students typically rely on traditional and often limited methods. Many turn to personal contacts and networking to find opportunities, which can be challenging and restrictive for young individuals who are just beginning to build their professional networks. Others may resort to searching on generic job portals or company websites, which often cater to experienced professionals rather than students seeking short-term, entry-level experience. This often leads to a time-consuming and frustrating experience, with students needing to manually search for suitable roles and apply individually to each company.

From a company's perspective, the process is equally cumbersome. Without a centralized hub, companies must rely on multiple channels to attract student interns. This can involve setting up manual application processes, sorting through a large volume of unqualified resumes from general job boards and dedicating significant human resources to a time-consuming screening process. The lack of a dedicated platform means there's no pre-screening or qualification process, making it difficult to efficiently identify suitable candidates from the outset. In essence, the existing system is characterized by its manual nature, lack of centralization, and inefficiency for both students and companies.

## 2.2 Proposed System:

The proposed InternQuest system is a comprehensive, web-based platform designed to address the shortcomings of existing, manual internship search and management processes. This system provides a centralized hub that efficiently connects 12th-pass students with companies seeking interns, transforming the unstructured search process into a streamlined, automated workflow. The platform is structured around three distinct user roles: Student, Company, and Admin, each with a dedicated dashboard to manage their specific tasks.

The system's core functionality is to facilitate a clear and guided journey for students from application to certification. After registering and completing a profile, students must pass an aptitude test before they can view or apply for internships. This pre-screening step ensures that only qualified and serious candidates can proceed, saving time for both students and companies. Once a student is selected by a company, they are required to pay a nominal fee of ▪149 to receive their official offer letter, a mechanism that helps filter for committed applicants and contributes to the platform's operational sustainability. The system automates the generation of crucial documents like the offer letter, payment receipt, and a certificate of completion upon the successful conclusion of the internship.

For companies, the proposed system simplifies the hiring and management of interns. Instead of sifting through countless resumes from various sources, companies can post their internship opportunities directly on the platform and manage all applications from a single dashboard. They can easily approve or reject applications and issue offer letters with just a few clicks. The platform also provides a transparent view of all applicants and their progress, allowing companies to track their current and completed interns efficiently.

The administrative module provides an oversight and control layer for the entire system. The admin can manage all user accounts, approve new companies, and oversee the internship and application pipeline. Furthermore, the admin can manage the aptitude test by adding or deleting questions and can update the website's content and FAQs through a built-in Content Management System (CMS). This centralized control ensures the platform's integrity, security, and relevance.

## 2.3 Requirement Analysis:

The InternQuest system's design is based on a detailed analysis of its functional and non-functional requirements to ensure it effectively serves its target users.

---

**Functional Requirements**

Functional requirements define what the system must do to fulfill its purpose.

- **User Registration and Authentication**: The system must allow users to register and log in with unique credentials, supporting three distinct roles: Student, Company, and Admin.
- **Profile Management**:
  - **Student**: A student must be able to create a profile with personal details, academic information (12th-grade stream and marks), and bank details for future payments.
  - **Company**: A company must be able to create and manage its profile, including a description of its business.
- **Aptitude Test**: The system must provide a mandatory aptitude test for students. It must be timed (10 minutes for 15 questions), require a passing score of 12 or more, and limit students to a maximum of three attempts.
- **Internship Management**:
  - **Student**: Students who pass the aptitude test must be able to view a list of available internships and apply with a single click.
  - **Company**: Companies must be able to post new internship opportunities with a title and description.
- **Application Workflow**: The system must track the status of applications, which transitions through various stages: Applied, Pending Company Approval, Selected, Payment Complete, Offer Letter Issued, Ongoing, and Completed.

- **Document Generation**: The platform must automatically generate and allow users to download a formal offer letter, a payment receipt, and a certificate of completion as PDF documents.
- **Payment Simulation**: The system must have a simulated payment process where a student pays a □149 fee to get their official offer letter.
- **Admin Control**: The administrator must have a dashboard to manage all user accounts, approve new companies, oversee all applications, and manage the content of the aptitude test and the website's front page.

---

**Non-Functional Requirements**

Non-functional requirements specify the criteria used to judge the system's operation, rather than specific behaviors.

- **Usability**: The user interface for all three dashboards (Student, Company, Admin) must be intuitive and easy to navigate.
- **Security**: The platform must ensure the security of user data, including login credentials and personal information, using Firebase Authentication and Firestore's security rules to prevent unauthorized access.
- **Reliability**: The system must be stable and available 24/7 with minimal downtime.
- **Performance**: The platform should respond quickly to user interactions, with fast page loading and real-time updates for application statuses.
- **Scalability**: The architecture must be scalable to handle a growing number of students, companies, and internship applications without a decline in performance.
- **Maintainability**: The codebase should be well-structured and documented to allow for easy maintenance, bug fixes, and the addition of new features.

---

## 2.4 Hardware and Software Requirements:

Developing and deploying the InternQuest platform requires a specific set of hardware and software tools. The project's client-side, web-based nature and its reliance on a serverless backend simplify these requirements significantly.

---

### Hardware Requirements

The system's core is a client-side web application, so the hardware requirements are minimal. The platform does not require a dedicated server, as Firebase handles all backend infrastructure.

- **Development Environment:** A standard personal computer or laptop is sufficient. A system with an Intel i3/i5 processor (or equivalent), at least 8 GB of RAM, and a Solid-State Drive (SSD) of 256 GB or more is recommended for a smooth development experience.
- **User Environment:** End-users (students and companies) can access the platform from any device (desktop, laptop, tablet, or smartphone) with a modern web browser and a stable internet connection.

---

### Software Requirements

The software stack for InternQuest is based on a modern web development approach, utilizing a combination of front-end technologies and a cloud-based backend.

- **Front-End Development:**
    - **Operating System:** Windows 10/11, macOS, or any Linux distribution.
    - **Code Editor:** A source-code editor like Visual Studio Code (VS Code) is ideal for writing and managing the project's code.
    - **Web Technologies:** The entire front-end is built using HTML5, CSS3, and JavaScript (ES6+) for structure, styling, and interactivity.

  - **Libraries:** The **jsPDF** library is used on the client side to generate PDF documents, such as offer letters, receipts, and certificates.

- **Back-End and Database:**
  - **Platform: Firebase** is the primary backend platform, providing multiple services that are crucial for the system's functionality.
  - **Authentication: Firebase Authentication** is used for secure user login and registration.
  - **Database: Firebase Firestore** serves as the NoSQL database to store all application data, including user profiles, internships, applications, and payments.
  - **Hosting: Firebase Hosting** is used for deploying the static web files (.html, .css, .js) of the application.

## 2.5 Justification of Selection of Technology:

The technology stack for the InternQuest project was chosen based on a careful analysis of the system requirements, focusing on efficiency, scalability, and ease of development. The selected stack—consisting of HTML, CSS, JavaScript, and Firebase—provides a robust and modern foundation for a serverless web application.

---

**Frontend Technologies: HTML, CSS, and JavaScript**

HTML, CSS, and JavaScript were selected as the core front-end technologies because they are the foundational languages of the web, ensuring maximum compatibility and accessibility across all major web browsers and devices.

- **HTML5** provides the semantic structure for the web pages, from user dashboards to application forms.
- **CSS3** is used for responsive and modern styling, ensuring a clean and user-friendly interface as demonstrated by the style.css file. It allows the platform to be easily accessible and visually appealing on desktops, tablets, and mobile phones.
- **JavaScript (ES6+)** is the primary programming language for all client-side logic and interactivity. It handles dynamic content updates, form validation, and the core functionality of the aptitude test and application process. The use of JavaScript libraries like jsPDF for on-the-fly PDF generation is a key reason for its selection, as it eliminates the need for a server-side document generation process.

---

**Backend as a Service (BaaS): Firebase**

Firebase, a platform from Google, was chosen as the backend for its serverless architecture and powerful, integrated services. This choice significantly reduced development complexity and time compared to building a custom backend from scratch.

- **Firebase Authentication** provides a secure and straightforward way to handle user registration and login. It supports email and password-based authentication out of the box, which was a key functional requirement.

- **Firebase Firestore** serves as the NoSQL database. It offers real-time data synchronization, a crucial feature for a platform where application statuses need to be updated instantly across multiple user dashboards. Its document-based model is highly flexible, making it ideal for storing structured data for users, internships, and applications.

- **Firebase Hosting** was selected for its simplicity and reliability in deploying the static web application files. It provides a fast, secure, and global content delivery network (CDN), ensuring low-latency access for all users.

- **Cost-Effectiveness and Scalability:** The serverless nature of Firebase means the platform can scale automatically to handle a growing user base without manual intervention. This pay-as-you-go model is highly cost-effective, as resources are only consumed when the application is actively in use.

# Chapter 3: System Design

## 3.1 Modulo Division:

Modulo division, in the context of system design, refers to breaking down a complex system into smaller, more manageable, and independent modules. This approach simplifies development, makes debugging easier, and improves the overall organization of the project. The InternQuest platform is divided into six key modules, each handling a specific set of functionalities and interacting with the others as needed.

- **Authentication Module:** This is the gateway to the system. Its sole responsibility is to manage user identity and access. It handles the logic for user registration and login for all three user types: student, company, and admin. It also ensures that a user's session is properly managed, allowing them to access their respective dashboards after successful authentication.

- **Student Module:** This module contains all the features accessible to a student user. It manages the student's personal and academic profile, handles the aptitude test logic, and facilitates the browsing and application for internships. Additionally, this module includes the payment process for offer letters and the functionality to download generated documents like offer letters, certificates, and payment receipts.

- **Company Module:** This module is designed for companies to manage their presence on the platform. It allows them to create and update their company profile, post and manage internship listings, and oversee all received applications. The module also provides the tools to approve or reject student applications and to issue offer letters to selected candidates.

- **Admin Module:** This module provides the administrator with full oversight and control of the entire platform. It contains functionalities for managing all registered users (students and companies), managing the aptitude test questions, and updating the website's content via a built-in CMS. The admin can also manage payments and manually oversee the application process.

- **Database Module:** This is the core data management unit of the system. It handles all Create, Read, Update, and Delete (CRUD) operations for the Firebase Firestore database. This module ensures data consistency and integrity by managing how all other modules interact with the database.

- **Document Generation Module:** This module is responsible for creating and formatting the PDF documents. It takes data from the database (e.g., student name, company details, internship title) and uses the jsPDF library to generate offer letters, certificates, and payment receipts, which can then be downloaded by users.

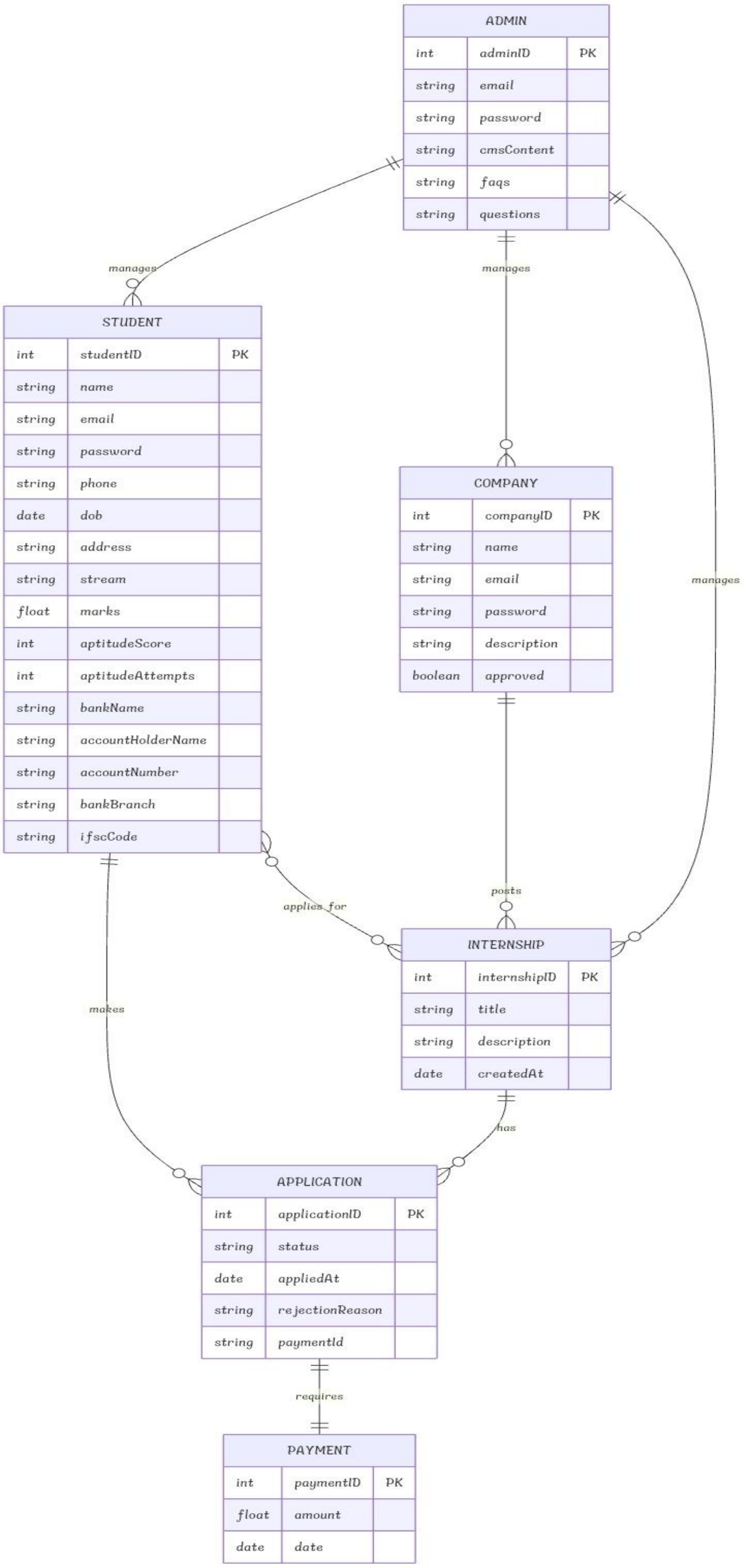
## 3.2 ER Diagram:

An ER Diagram is a visual representation of the logical structure of a database. It shows the entities (data objects), their attributes (properties), and the relationships between them. For the InternQuest system, the key entities are Student, Company, Admin, Internship, and Application.

### Entities and Their Attributes

- Student (Entity): Represents a student user.
- Attributes: studentID (Primary Key), name, email, password, phone, dob (Date of Birth), address, stream, marks, aptitudeScore, aptitudeAttempts, bankName, accountHolderName, accountNumber, bankBranch, ifscCode.
- Company (Entity): Represents a company user.
- Attributes: companyID (Primary Key), name, email, password, description, approved.
- Admin (Entity): Represents the platform administrator.
- Attributes: adminID (Primary Key), email, password, cmsContent, faqs, questions.
- Internship (Entity): Represents an internship listing posted by a company.
- Attributes: internshipID (Primary Key), title, description, createdAt.
- Application (Entity): Represents a student's application for a specific internship.
- Attributes: applicationID (Primary Key), status, appliedAt, rejectionReason, paymentId.
- Payment (Entity): Represents a payment made by a student.
- Attributes: paymentID (Primary Key), amount, date.
- Relationships
- The relationships define how entities are connected.
- Student – Makes – Application: A one-to-many relationship. One student can make many applications.
- Company – Posts – Internship: A one-to-many relationship. One company can post many internships.
- Internship – Has – Application: A one-to-many relationship. One internship can receive many applications.
- Student – Applies for – Internship: A many-to-many relationship, which is resolved by the Application entity. The Application links a specific studentID to a specific internshipID.
- Admin – Manages – Student: A one-to-many relationship. The admin can manage many students.
- Admin – Manages – Company: A one-to-many relationship. The admin can manage many companies.
- Admin – Manages – Internship: A one-to-many relationship. The admin can manage many internships.
- Application – Requires – Payment: A one-to-one relationship. Each completed payment is linked to a single application.

The ER diagram visually depicts this structure, with entities shown as rectangles, attributes as ovals, and relationships as diamonds. The connecting lines indicate the cardinality of the relationships (one-to-one, one-to-many, etc.).



| ADMIN | | |
|---|---|---|
| int | adminID | PK |
| string | email | |
| string | password | |
| string | cmsContent | |
| string | faqs | |
| string | questions | |

| STUDENT | | |
|---|---|---|
| int | studentID | PK |
| string | name | |
| string | email | |
| string | password | |
| string | phone | |
| date | dob | |
| string | address | |
| string | stream | |
| float | marks | |
| int | aptitudeScore | |
| int | aptitudeAttempts | |
| string | bankName | |
| string | accountHolderName | |
| string | accountNumber | |
| string | bankBranch | |
| string | ifscCode | |

| COMPANY | | |
|---|---|---|
| int | companyID | PK |
| string | name | |
| string | email | |
| string | password | |
| string | description | |
| boolean | approved | |

| INTERNSHIP | | |
|---|---|---|
| int | internshipID | PK |
| string | title | |
| string | description | |
| date | createdAt | |

| APPLICATION | | |
|---|---|---|
| int | applicationID | PK |
| string | status | |
| date | appliedAt | |
| string | rejectionReason | |
| string | paymentId | |

| PAYMENT | | |
|---|---|---|
| int | paymentID | PK |
| float | amount | |
| date | date | |

## 3.3 Data Flow Diagram (DFD):

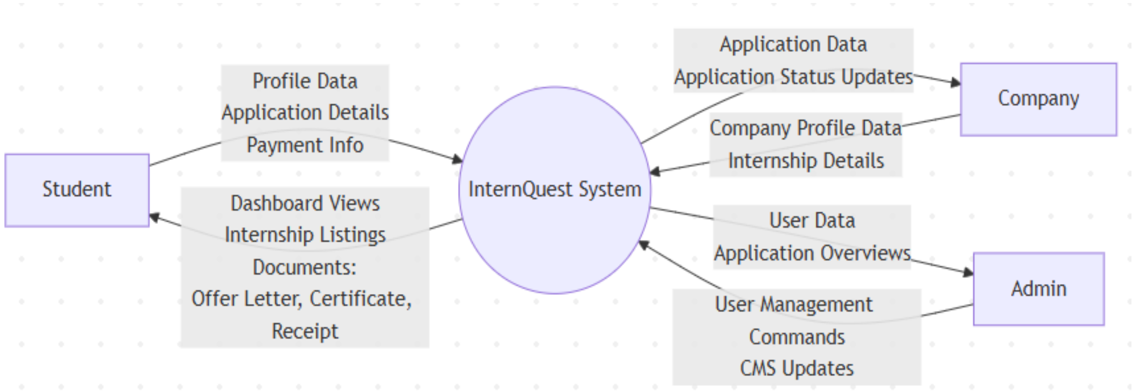
A Data Flow Diagram (DFD) visually represents the flow of information through a system, showing how data is processed and stored. It helps in understanding the system's processes and the relationships between its components. The InternQuest system can be broken down into three levels to illustrate data flow, starting from a high-level overview to more detailed processes.

---

**Level 0: Context Diagram:**

The Level 0 DFD provides a high-level view of the entire InternQuest system as a single process. It shows the main external entities that interact with the system and the data that flows between them.

- **Entities**: The main actors are the **Student**, the **Company**, and the **Admin**.
- **Data Flows**:
  - **Student** sends **Profile Data**, **Application Details**, and **Payment Information** to the system. The system returns **Dashboard Views**, **Internship Listings**, and **Generated Documents** (Offer Letter, Certificate, Receipt).
  - **Company** sends **Company Profile Data** and **Internship Details** to the system. It receives **Application Data** and manages **Application Status Updates**.
  - **Admin** sends **User Management Commands** and **CMS Updates** to the system. The system provides **User Data** and **Application Overviews**.

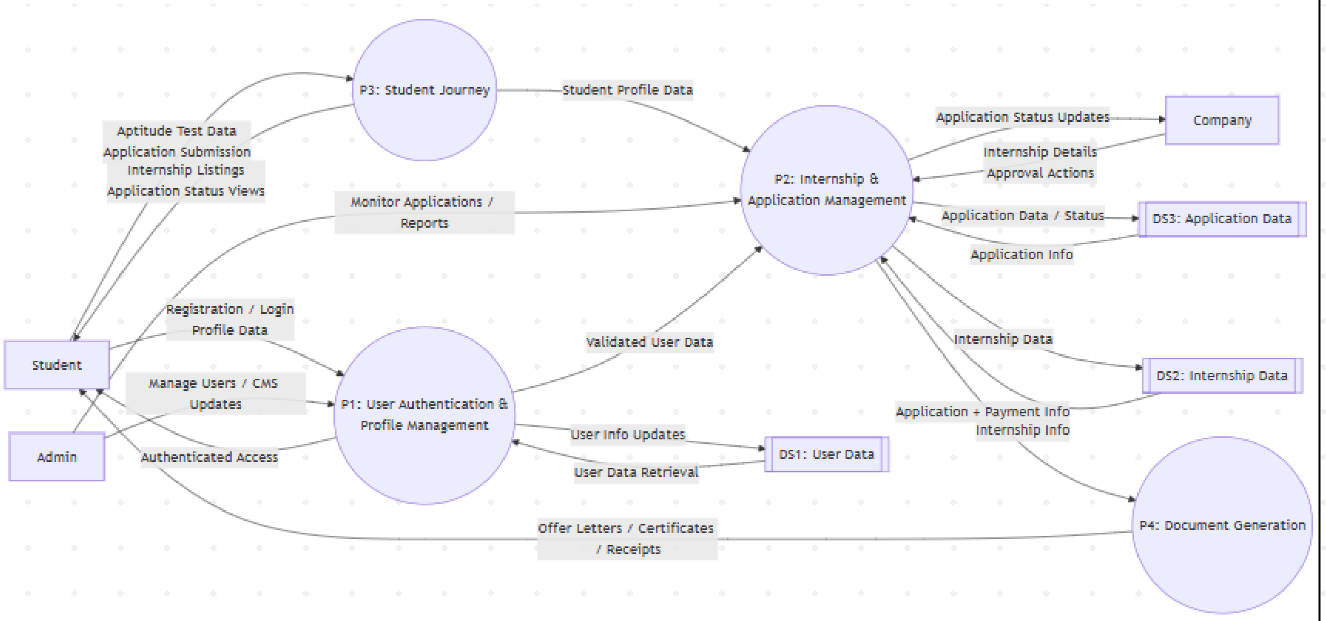


---

**Level 1: Main Processes:**

The Level 1 DFD breaks down the single system process into its major functional modules. This diagram shows how data moves between these modules and the system's data stores.

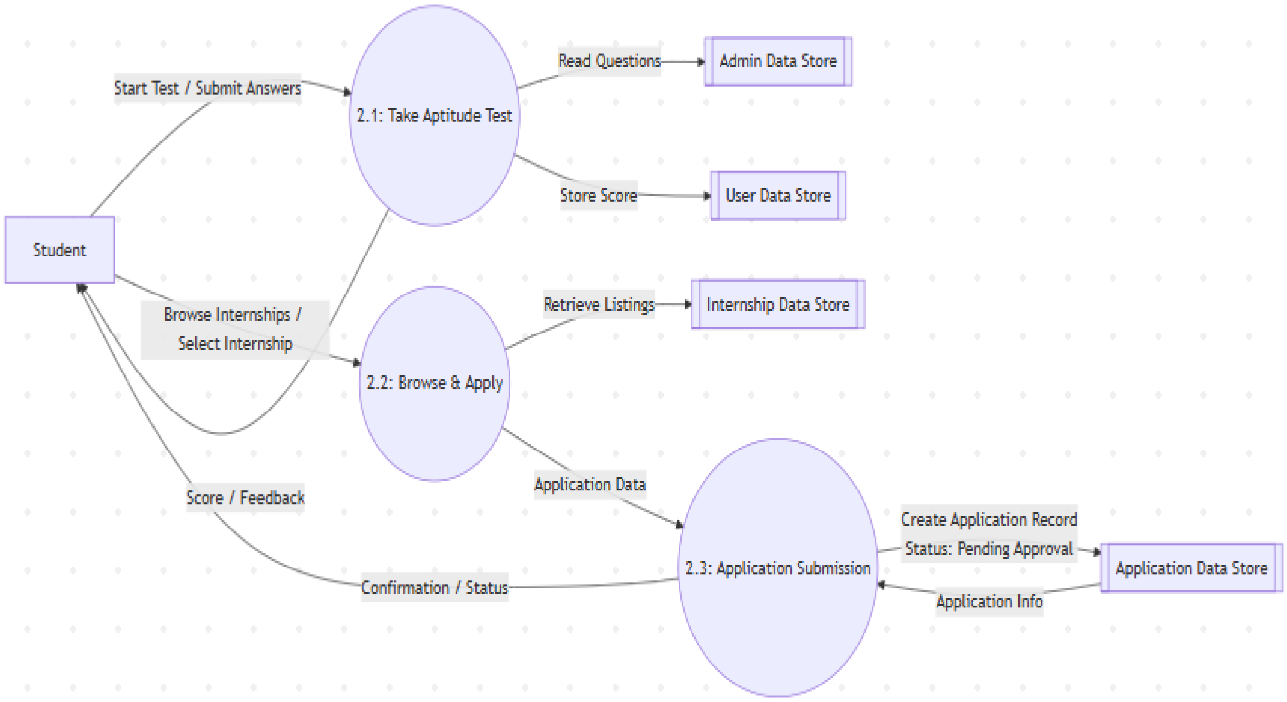
- **Processes**:
  - **P1: User Authentication & Profile Management**: This process handles user registration and login for all roles. It reads from and writes to the **User Data Store**.
  - **P2: Internship & Application Management**: This central process manages the entire internship lifecycle. It receives **Applications** from students, and companies send **Internship Details** and **Approval Actions**. This process interacts with the **Internship Data Store** and the **Application Data Store**.
  - **P3: Student Journey**: This process is specific to students. It takes **Student Profile Data** to enable the **Aptitude Test** and then provides access to **Internship Listings** and **Application Status** views.
  - **P4: Document Generation**: This process takes **Application Data** and **Payment Information** and produces **Offer Letters**, **Certificates**, and **Receipts**, which are then sent back to the student.
- **Data Stores**:
  - **DS1: User Data**: Stores all user information (profiles, credentials, roles).
  - **DS2: Internship Data**: Stores details of all internship listings.
  - **DS3: Application Data**: Stores all student applications and their current status.



**Level 2: Student Application Flow:**

This Level 2 DFD provides a detailed view of a key process, such as a student's journey from taking the aptitude test to completing an application.

- **Processes**:
  - **2.1 Take Aptitude Test**: The student interacts with this process, which reads **Aptitude Questions** from the **Admin Data Store** and records the **Score** in the **User Data Store**.
  - **2.2 Browse & Apply**: After passing the test, the student views **Internship Listings** from the **Internship Data Store** and submits an **Application**.
  - **2.3 Application Submission**: This process creates a new application record in the **Application Data Store** with a "Pending Admin Approval" status.
- **Data Stores**:
  - **User Data Store**: Stores the student's profile and aptitude score.
  - **Internship Data Store**: Stores the internship details.
  - **Application Data Store**: Stores the record of the submitted application.
  - **Admin Data Store**: Contains the aptitude test questions and other CMS data.



## 3.4 Use Case Diagram:

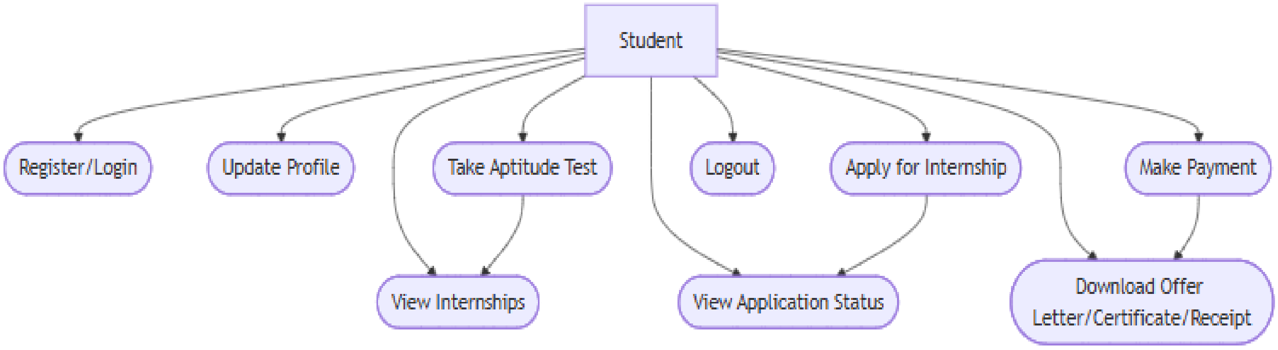
A Use Case Diagram illustrates the high-level functional requirements of a system by showing the interactions between users (actors) and the system itself. For the InternQuest platform, the main actors are the **Student**, **Company**, and **Admin**. Each actor has a distinct set of use cases, representing the functionalities they can perform.

---

**Student Use Case:**

The student is the primary user of the platform, with a journey that progresses through registration, profile creation, and internship application.

- **Register/Login**: The student can create a new account or sign in to an existing one.
- **Update Profile**: The student can fill in or update their personal, academic, and bank details.
- **Take Aptitude Test**: This is a mandatory step before applying for internships.
- **View Internships**: The student can browse the list of available internships once they pass the aptitude test.
- **Apply for Internship**: The student can submit an application for an internship of their choice.
- **View Application Status**: The student can track the progress of their applications through various stages.
- **Make Payment**: After a company selects them, the student can proceed with the simulated payment of □149 to get their offer letter.
- **Download Offer Letter/Certificate/Receipt**: The student can download their official documents at the appropriate stages of the internship.
- **Logout**: The student can securely exit their account.

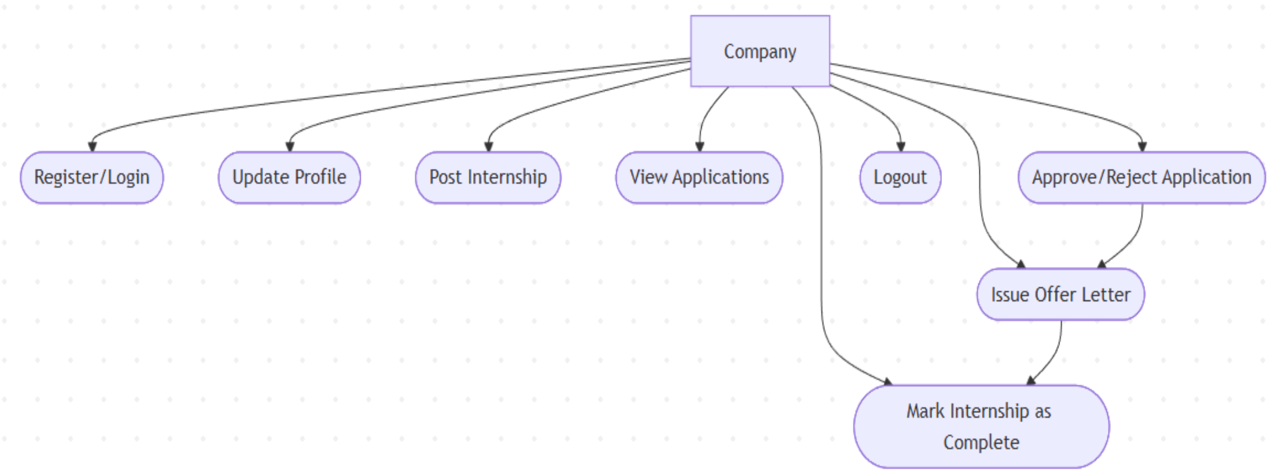


---

### Company Use Case:

The company acts as the internship provider and manages the application process from their end.

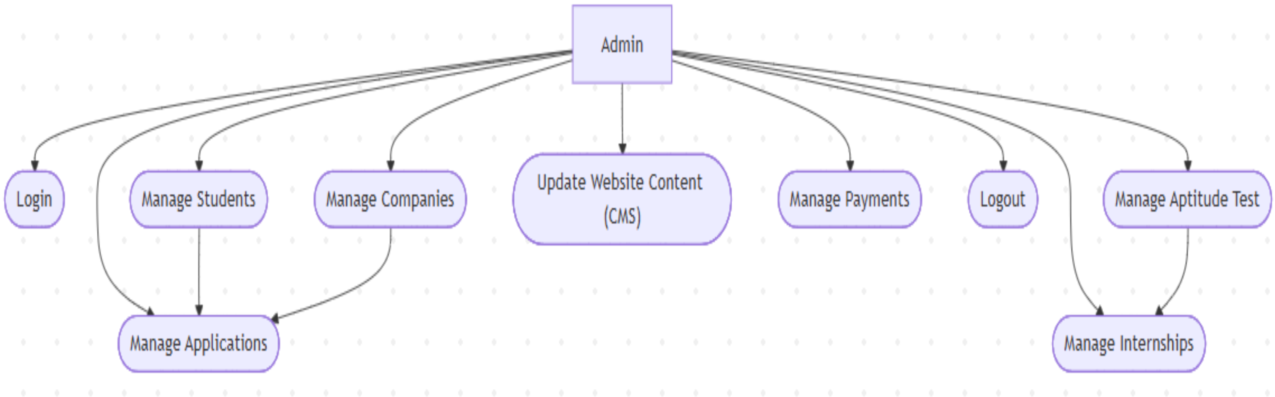
- **Register/Login**: The company can create an account or sign in.
- **Update Profile**: The company can update its profile details, including a description of the business.
- **Post Internship**: The company can post new internship opportunities with details like title and description.
- **View Applications**: The company can view all student applications submitted for their internships.
- **Approve/Reject Application**: The company can accept or decline a student's application.
- **Issue Offer Letter**: After a student makes a payment, the company can issue the offer letter.
- **Mark Internship as Complete**: Once the internship is finished, the company can mark it as complete.
- **Logout**: The company can securely exit their account.



**Admin Use Case:**

The administrator has a privileged role with complete control and oversight of the platform's operations.

- **Login**: The admin can sign in to their exclusive dashboard.
- **Manage Students**: The admin can view and manage all student accounts.
- **Manage Companies**: The admin can view and approve or delete company accounts.
- **Manage Aptitude Test**: The admin can add or delete questions for the student aptitude test.
- **Manage Internships**: The admin can manage all internship listings on the platform.
- **Manage Applications**: The admin has a comprehensive view of all applications and can approve or reject them on a case-by-case basis.
- **Update Website Content (CMS)**: The admin can modify the hero section of the website's homepage and manage the Frequently Asked Questions (FAQs).
- **Manage Payments**: The admin can monitor all payment records.
- **Logout**: The admin can securely exit their account.



## 3.5 Deployment Diagram:

A deployment diagram illustrates the physical architecture of a system, showing how software components are mapped onto hardware nodes. For the InternQuest platform, the architecture is a **client-side web application** that leverages a **serverless backend**. This design simplifies the deployment process by eliminating the need for a traditional application server. The system is deployed across three primary nodes: the user's device, Firebase Hosting, and the Firebase backend services.

---

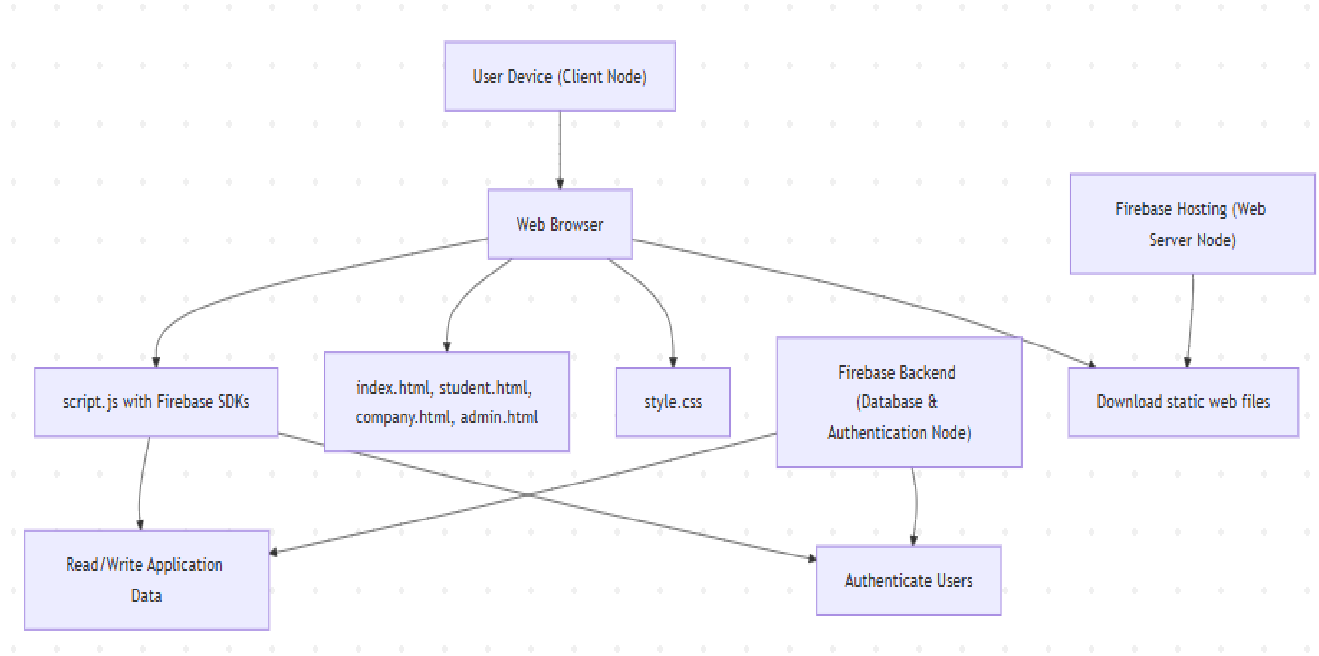
**Node Components:**

- **User Device (Client Node):**
  - This represents the end-user's hardware, such as a desktop computer, laptop, or mobile device.
  - It hosts the **Web Browser**, which runs the client-side application. The application consists of the index.html, student.html, company.html, and admin.html files for structure, along with style.css for styling and script.js for all the application logic.
  - The script.js file contains the Firebase SDKs, which enable direct communication with Firebase services without an intermediary server.
- **Firebase Hosting (Web Server Node):**
  - This is the cloud-based server where the static web files of the InternQuest application are deployed.
  - It acts as a content delivery network (CDN), providing fast and reliable delivery of the web pages to the user's browser. It's essentially a file server for the front end.
- **Firebase Backend (Database & Authentication Server Node):**
  - This node represents the suite of serverless services provided by Google's Firebase platform.

- **Firebase Firestore** is the database component that stores all application data, including user profiles, internships, and application statuses.
- **Firebase Authentication** handles the secure management of user accounts and credentials.

**Connections and Data Flow:**

The diagram shows a direct connection from the User Device's Web Browser to the Firebase Backend. This is a key characteristic of a serverless architecture: the client-side JavaScript communicates directly with the database and authentication services. The browser first downloads the necessary static files from Firebase Hosting. Once loaded, the script.js file handles all subsequent data interactions by making API calls directly to Firestore and Authentication. This model reduces latency and the complexity of managing a traditional server.

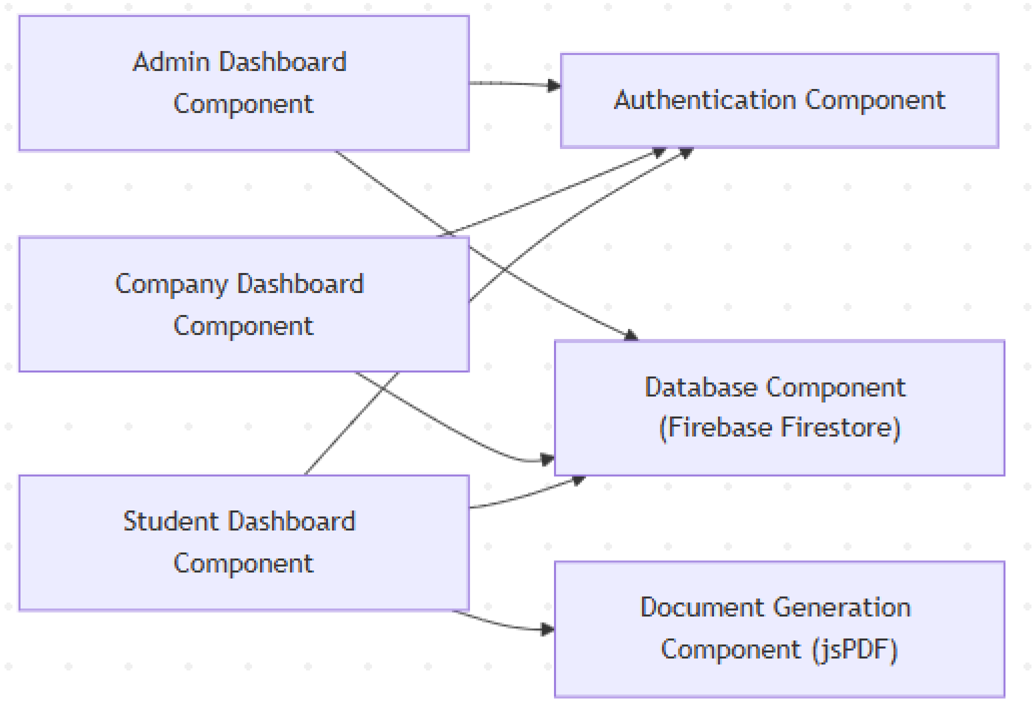

User Device (Client Node)
Web Browser
Firebase Hosting (Web Server Node)
script.js with Firebase SDKs
index.html, student.html, company.html, admin.html
style.css
Firebase Backend (Database & Authentication Node)
Download static web files
Read/Write Application Data
Authenticate Users

## 3.6 Component Diagram:

A component diagram illustrates the structural relationships between the various software components of a system. It provides a high-level view of the architecture, showing how different parts of the software are organized and how they depend on each other. The InternQuest platform is built around several key components that work together to provide a seamless user experience.

- **Authentication Component**: This component is responsible for user identity and access management. It handles all registration and login requests from students, companies, and the admin. It securely interacts with the **Firebase Authentication** service to verify credentials and manage user sessions. The Authentication Component is a foundational element, as all other user-facing components depend on it for access control.
- **Dashboard Components**: This is a collective term for the three main user interfaces:
  - **Student Dashboard Component**: Manages the student-specific functionalities, including profile management, the aptitude test, and the display of internship listings and application statuses. It depends on the Authentication, Database, and Document Generation Components to function.
  - **Company Dashboard Component**: Manages company-specific tasks, such as posting internships and reviewing applications. It depends on the Authentication and Database Components.
  - **Admin Dashboard Component**: Manages the administrative oversight, including user management, CMS updates, and application oversight. It depends on the Authentication and Database Components.
- **Database Component**: This is the core data management layer of the system. It handles all interactions with the **Firebase Firestore** database, processing requests to read, write, update, and delete data. All dashboard components rely on this component to retrieve and store information.
- **Document Generation Component**: This component is responsible for creating PDF files such as offer letters, certificates, and payment receipts. It takes data provided by

the Database Component (via other components) and uses the **jsPDF** library to format and generate the documents. This component is crucial for providing tangible proof of a student's internship and payment.


Admin Dashboard Component
Authentication Component
Company Dashboard Component
Database Component (Firebase Firestore)
Student Dashboard Component
Document Generation Component (jsPDF)

## 3.7 Activity Diagram:

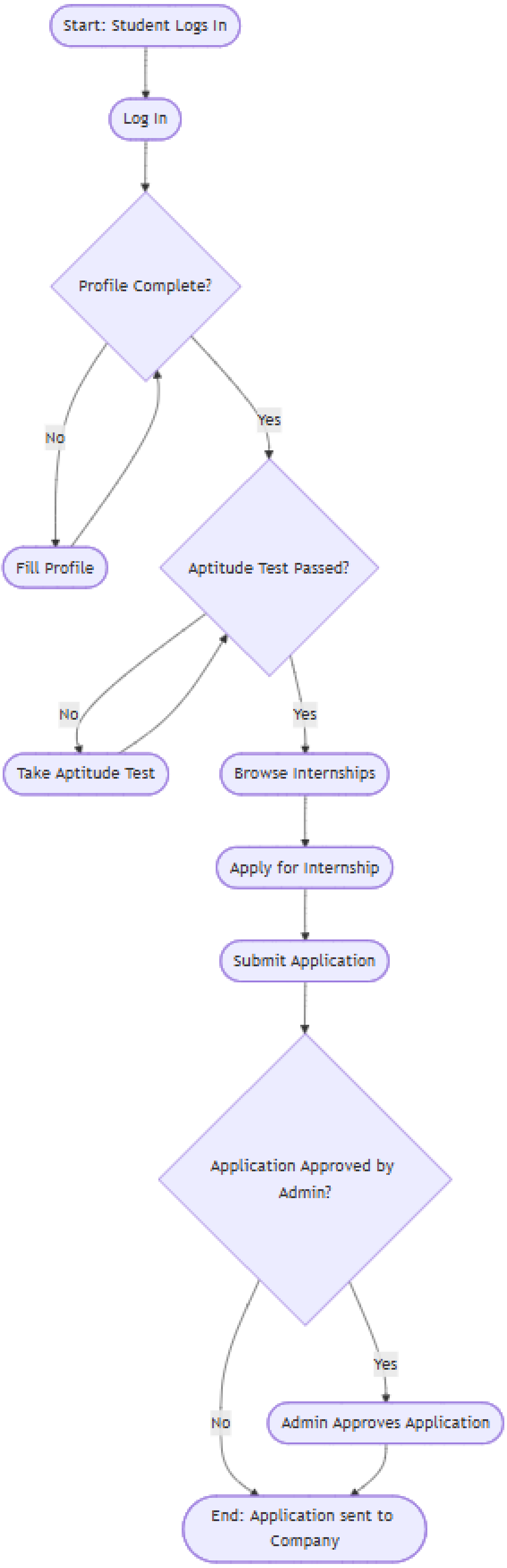
An activity diagram models the flow of control within a system's processes, much like a flowchart. It shows the sequence of actions and the decisions that guide the workflow. For the InternQuest platform, activity diagrams are particularly useful for visualizing the user journey and the automated steps that take place behind the scenes.

---

**Activity Diagram: Student Internship Application Flow**

This diagram illustrates the complete process a student goes through to find and apply for an internship, from logging in to a company reviewing their application.

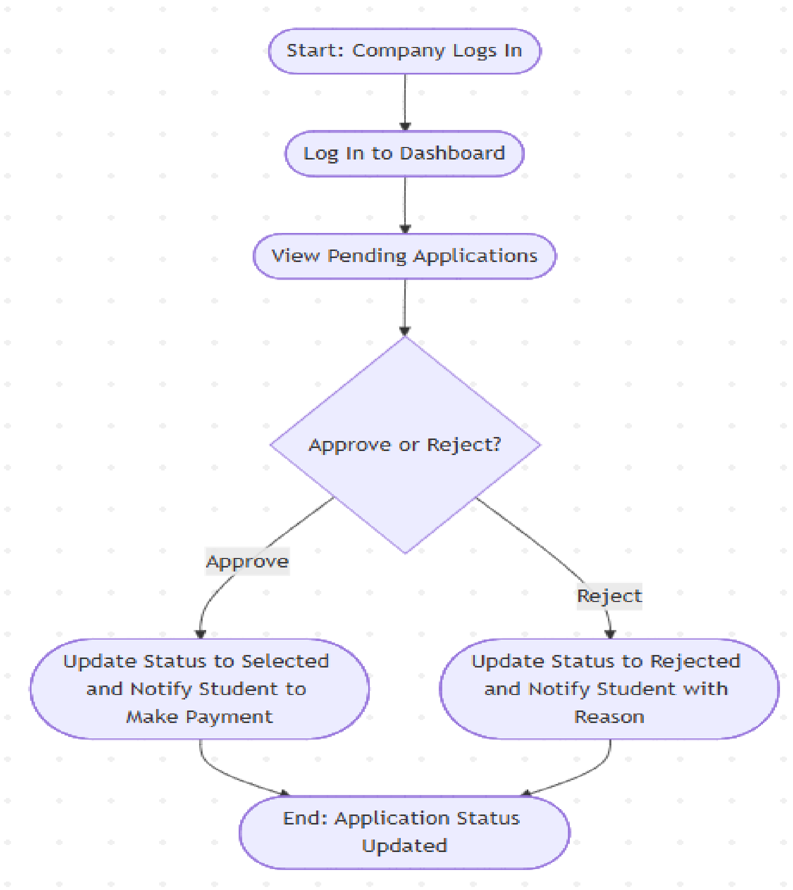
- **Start Node**: The process begins when the student logs into the system.
- **Activity: Log In**: The student enters their credentials.
- **Decision Node: Profile Complete?**: The system checks if the student's profile, including all mandatory fields like bank details, has been filled out. If the profile is incomplete, the flow returns to the **Activity: Fill Profile**.
- **Activity: Fill Profile**: The student completes all required profile information.
- **Decision Node: Aptitude Test Passed?**: The system checks if the student has passed the aptitude test with a score of 12 or more. If the test has not been passed, the flow proceeds to the **Activity: Take Aptitude Test**.
- **Activity: Take Aptitude Test**: The student takes the test.
- **Activity: Browse Internships**: Once both the profile is complete and the test is passed, the student can view all available internship listings.
- **Activity: Apply for Internship**: The student selects an internship and submits their application.
- **Activity: Submit Application**: The system automatically submits the application.
- **Decision Node: Application Approved by Admin?**: A key step in the workflow is the admin's review. If the application is not approved by the admin, the flow ends. If approved, the application's status is updated to "Pending Company Approval".
- **Activity: Admin Approves Application**: The admin reviews and approves the application, which is then sent to the company.
- **End Node**: The flow concludes, with the application now in the hands of the company for review.


Start: Student Logs In
Log In
Profile Complete?
No
Yes
Fill Profile
Aptitude Test Passed?
No
Yes
Take Aptitude Test
Browse Internships
Apply for Internship
Submit Application
Application Approved by Admin?
Yes
No
Admin Approves Application
End: Application sent to Company

### Activity Diagram: Company Application Review Flow

This diagram outlines the process a company follows when a student's application is sent for their review.

- **Start Node**: The process starts when a company logs in and navigates to the "Pending Approvals" section of their dashboard.
- **Activity: Log In to Dashboard**: The company signs in to its dashboard.
- **Activity: View Pending Applications**: The company can see all applications that require their review.
- **Decision Node: Approve or Reject?**: The company reviews a specific application and decides whether to approve or reject it.
    - **If Approved**: The application's status is updated to "Selected." This action triggers a notification to the student, prompting them to make a payment.
    - **If Rejected**: The company can provide a rejection reason. The application's status is updated to "Rejected," and the student is notified.
- **End Node**: The flow concludes with the application's final status being updated in the system.


Start: Company Logs In
Log In to Dashboard
View Pending Applications
Approve or Reject?
Approve
Reject
Update Status to Selected and Notify Student to Make Payment
Update Status to Rejected and Notify Student with Reason
End: Application Status Updated

## 3.8 Objective Diagram:

An Objective Diagram, also known as a Goal-Oriented Requirements Engineering (GORE) diagram, is a powerful tool used to model the goals of a system and the relationships between those goals and the tasks (functionalities) that help achieve them. Unlike a DFD or a Use Case Diagram, which focus on processes and user interactions, an Objective Diagram provides a clear, hierarchical view of the "why" behind the system's features. It links high-level business goals to the specific, implementable functionalities of the InternQuest platform.

The diagram begins with the ultimate goal and branches out to show how sub-goals and individual tasks contribute to achieving it. It helps to ensure that every component of the system has a direct purpose related to the project's overall objectives.

**High-Level Goal: Bridge Education-Employment Gap**

The top-level goal of InternQuest is to bridge the significant gap that exists between a student's academic life and their professional career.

- **Sub-Goal 1: Empower Students**: This goal focuses on providing students with the tools and knowledge they need to succeed in their careers.
    - **Task: Skill Acquisition**: The system enables students to gain new skills and industry knowledge. This is achieved through the **Functionality: Aptitude Test & Internship Listings**. The test assesses their readiness, and the listings provide a direct path to practical experience.
    - **Task: Gain Experience and Validation**: Students need proof of their work. This is supported by the **Functionality: Document Generation (Offer Letter, Certificate)**, which provides tangible, verifiable proof of their internship.
- **Sub-Goal 2: Facilitate Company Hiring**: This goal addresses the needs of companies by making the process of finding interns easier and more efficient.
    - **Task: Access to a Qualified Talent Pool**: Companies need a reliable source of motivated students. This is supported by the **Functionality: Company**

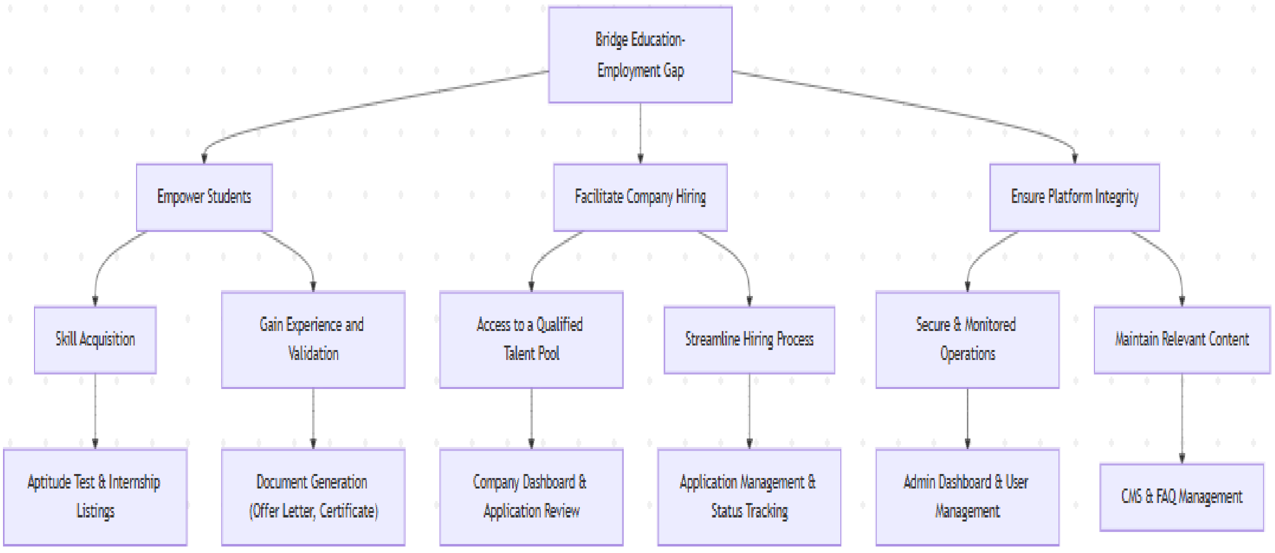
**Dashboard & Application Review**. The dashboard allows companies to post internships and easily review applications.

  - **Task: Streamline Hiring Process**: The process should be simple and fast. This is enabled by the **Functionality: Application Management & Status Tracking**. The system automates the workflow from application submission to approval and offers a centralized view for companies to manage their applicants.

- **Sub-Goal 3: Ensure Platform Integrity**: This goal focuses on the administrative and operational health of the system.

  - **Task: Secure & Monitored Operations**: The platform needs to be secure and well-managed. This is handled by the **Functionality: Admin Dashboard & User Management**, which gives the admin oversight and control.

  - **Task: Maintain Relevant Content**: The website needs to stay up-to-date and useful for users. This is achieved through the **Functionality: CMS & FAQ Management**, which allows the admin to update content easily.

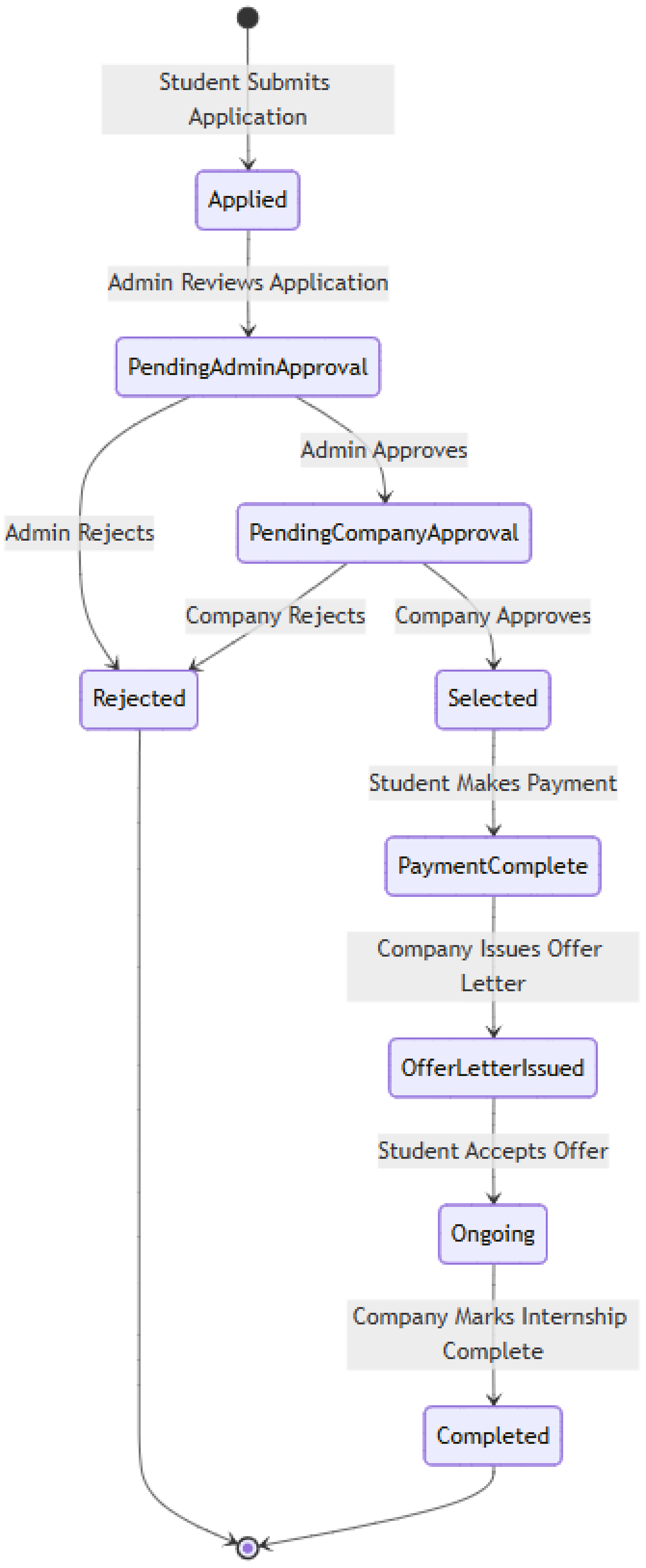

Bridge Education-Employment Gap
Empower Students
Facilitate Company Hiring
Ensure Platform Integrity
Skill Acquisition
Gain Experience and Validation
Access to a Qualified Talent Pool
Streamline Hiring Process
Secure & Monitored Operations
Maintain Relevant Content
Aptitude Test & Internship Listings
Document Generation (Offer Letter, Certificate)
Company Dashboard & Application Review
Application Management & Status Tracking
Admin Dashboard & User Management
CMS & FAQ Management

## 3.9 State Chart Diagram:

A state chart diagram is used to model the behavior of a system or a specific object within it by showing its various states and the transitions that cause it to move from one state to another. For the InternQuest platform, a state chart diagram is particularly useful for illustrating the lifecycle of a student's internship application, from its initial submission to its final status. This diagram helps in understanding the complex workflow and the events that trigger changes.

The primary object modeled here is an **Internship Application**. Its state changes based on actions taken by the student, the company, and the admin.

- **Applied**: This is the initial state of an application. It begins when a student submits an application for an internship.
- **Pending Admin Approval**: After a student applies, the application moves to this state. It remains here until an admin reviews and approves it. The event that triggers the transition is the **admin's approval** of the application.
- **Pending Company Approval**: Once the admin approves, the application's status changes to "Pending Company Approval". The event that causes the next transition is the **company's decision** to either approve or reject the application.
- **Selected**: If a company chooses to accept a student, the application moves to the "Selected" state. At this point, the student is notified to pay a nominal fee to receive their offer letter. The event that triggers the next state change is the **student's payment**.
- **Payment Complete**: When the student completes the simulated payment of □149, the application's status is automatically updated to "Payment Complete". This new state signals to the company that they can now issue the official offer letter. The event that transitions to the next state is the **company issuing the offer letter**.
- **Offer Letter Issued**: Once the company issues the offer letter, the application's status changes to "Offer Letter Issued". The student can now view and download the document. The student's acceptance of the offer leads to the next state.
- **Ongoing**: The application enters the "Ongoing" state when the student accepts the offer letter. This indicates that the internship has officially begun. The final state transition is triggered by the **company marking the internship as complete**.
- **Completed**: This is the final state for a successful application. It is reached when the company marks the internship as complete on their dashboard. The student can now download their certificate of completion.
- **Rejected**: At any point in the workflow, an application can be rejected by either the admin or the company. The application transitions to the "Rejected" state, and the student is notified of the outcome.

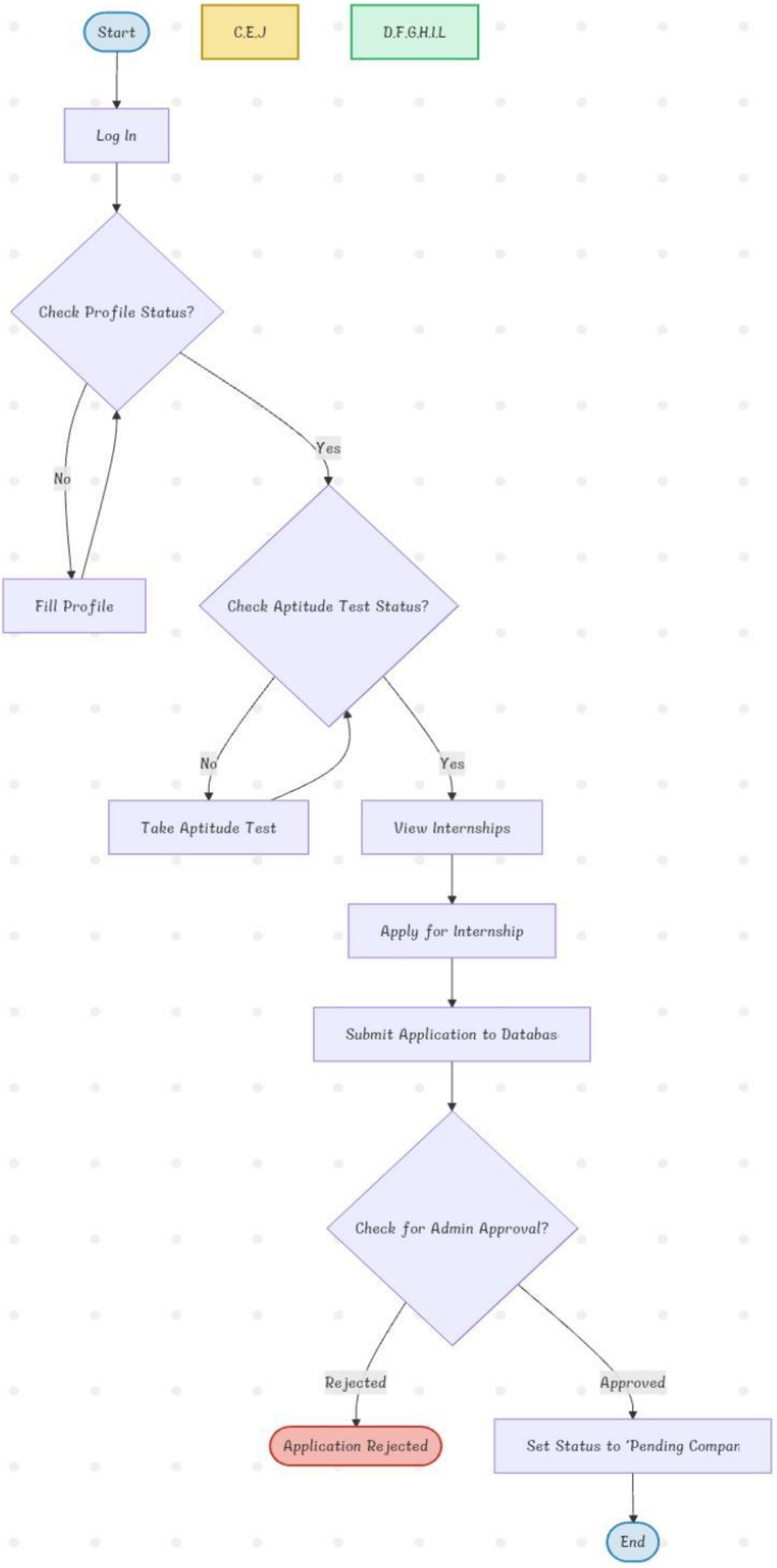

Student Submits Application
Applied
Admin Reviews Application
PendingAdminApproval
Admin Approves
Admin Rejects
PendingCompanyApproval
Company Rejects
Company Approves
Rejected
Selected
Student Makes Payment
PaymentComplete
Company Issues Offer Letter
OfferLetterIssued
Student Accepts Offer
Ongoing
Company Marks Internship Complete
Completed

## 3.10 Flow Chart:

### Student Internship Application Flow:

This flowchart illustrates the complete journey of a student from logging into the platform to their application being reviewed by a company.

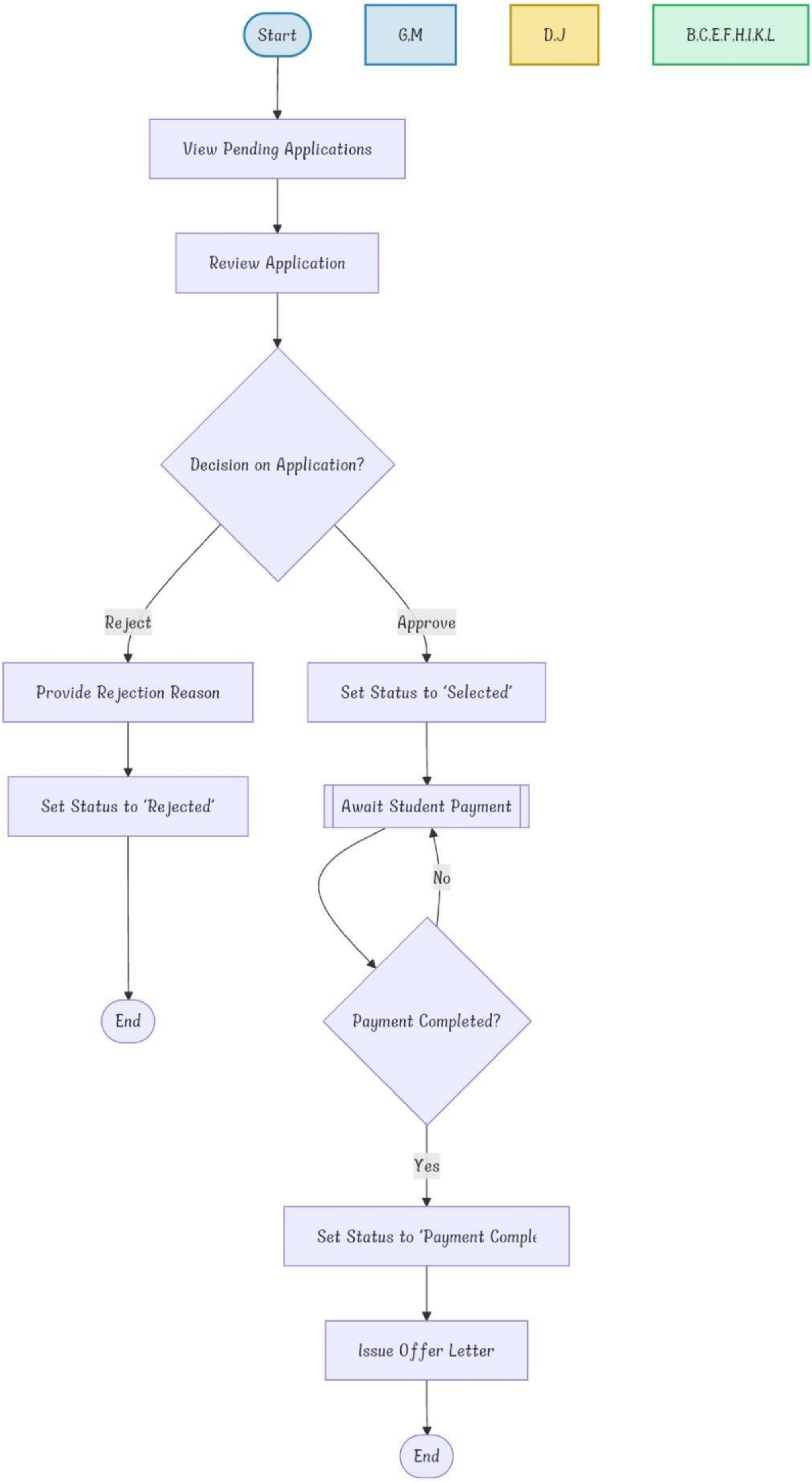
- Start (Oval): The process begins when the student opens the application.
- Log In (Rectangle): The student enters their credentials to log in.
- Check Profile Status (Diamond): The system checks if the student's profile is complete, including all mandatory fields like bank details.
- If "No" (Arrow to the right):
- Fill Profile (Rectangle): The student is directed to the "My Profile" section to complete their details.
- (Flow returns to the "Check Profile Status" diamond).
- If "Yes" (Arrow down):
- Check Aptitude Test Status (Diamond): The system checks if the student has passed the aptitude test with a score of at least 12.
- If "No" (Arrow to the right):
- Take Aptitude Test (Rectangle): The student takes the aptitude test.
- (Flow returns to the "Check Aptitude Test Status" diamond).
- If "Yes" (Arrow down):
- View Internships (Rectangle): The student is now able to view all available internship listings.
- Apply for Internship (Rectangle): The student selects an internship and clicks the "Apply" button.
- Submit Application to Database (Rectangle): The system submits the application, setting its status to "Applied" and storing it in the database.
- Check for Admin Approval (Diamond): The application waits for review by the admin.
- If "Rejected" (Arrow to the right):
- Application Rejected (Terminator): The process ends for this application, and its status is updated to "Rejected".
- If "Approved" (Arrow down):
- Set Status to "Pending Company Approval" (Rectangle): The application status is updated and becomes visible to the relevant company.
- End (Oval): The process concludes for the student's side, and the flow for the company begins.


Start
C,E,J
D,F,G,H,I,L
Log In
Check Profile Status?
No
Yes
Fill Profile
Check Aptitude Test Status?
No
Yes
Take Aptitude Test
View Internships
Apply for Internship
Submit Application to Databas
Check for Admin Approval?
Rejected
Approved
Application Rejected
Set Status to 'Pending Compan
End

## Flowchart: Company Application Review Flow:

This flowchart details the process for a company to manage a student application from initial review to final decision.

- Start (Oval): The company logs in and accesses their dashboard.
- View Pending Applications (Rectangle): The company navigates to the "Pending Approvals" section to see new applications.
- Review Application (Rectangle): The company reviews the student's profile and application details.
- Decision on Application (Diamond): The company decides whether to approve or reject the application.
- If "Reject" (Arrow to the right):
- Provide Rejection Reason (Rectangle): The company enters a reason for rejecting the application.
- Set Status to "Rejected" (Rectangle): The application status is updated in the database.
- End (Oval): The process concludes.
- If "Approve" (Arrow down):
- Set Status to "Selected" (Rectangle): The application's status is updated to "Selected," and the student is notified to make a payment.
- Await Student Payment (Delay Symbol): The process pauses until the student makes the required payment.
- Check for Payment Completion (Diamond): The system checks if the payment has been successfully processed.
- If "No" (Arrow up): The flow loops back to the "Await Student Payment" step.
- If "Yes" (Arrow down):
- Set Status to "Payment Complete" (Rectangle): The application status is automatically updated to "Payment Complete".
- Issue Offer Letter (Rectangle): The company can now issue the official offer letter to the student.
- End (Oval): The process concludes for this stage, with the internship officially beginning once the student accepts the offer.


Start
G,M
D,J
B,C,E,F,H,I,K,L
View Pending Applications
Review Application
Decision on Application?
Reject
Approve
Provide Rejection Reason
Set Status to 'Rejected'
End
Set Status to 'Selected'
Await Student Payment
No
Payment Completed?
Yes
Set Status to 'Payment Comple
Issue Offer Letter
End

## 3.11 Gantt Chart:

A Gantt chart is a project management tool used to visually represent a project's timeline, including its tasks, dependencies, and duration. For the InternQuest project, the Gantt chart provides a clear schedule for the development process, from initial requirements gathering to final documentation. It helps in tracking progress and managing the workload over time.

**The chart is divided into several key phases:**

- **Requirements and Analysis**: This initial phase involves understanding the problem, defining the system's objectives, and outlining the hardware and software requirements. This phase lays the foundation for all subsequent work.

- **System Design**: This phase focuses on creating the blueprints for the system. It includes designing the database structure, creating user interfaces (UI) and user experience (UX) flows, and developing the logical diagrams such as DFDs and use case diagrams.

- **Frontend Development**: This is the core coding phase for the user-facing part of the platform. It involves building the HTML structure, applying CSS for styling, and writing the JavaScript logic for the student, company, and admin dashboards.

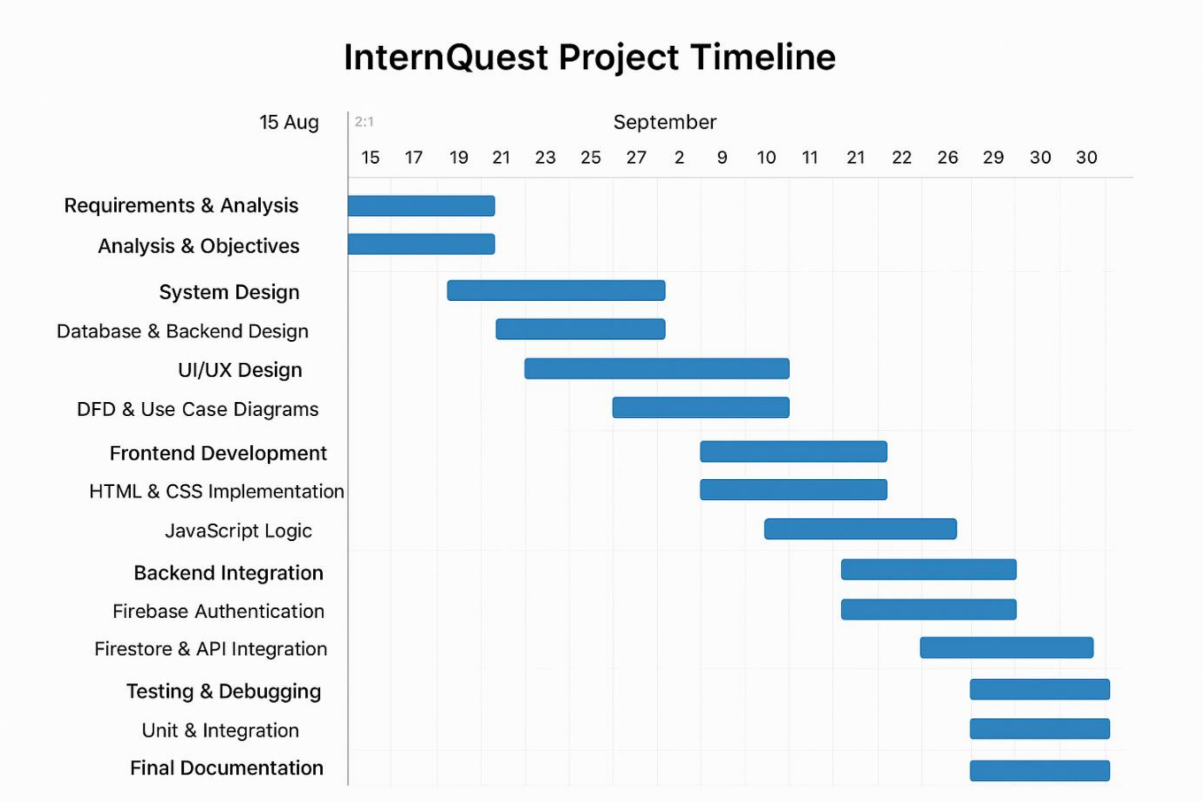
- **Backend Integration**: This phase connects the frontend to the Firebase backend. It includes setting up Firebase Authentication for user management, configuring Firestore for data storage, and implementing the logic for all API calls to the database.

- **Core Feature Implementation**: This phase is dedicated to building the most critical functionalities of the platform, such as the aptitude test logic, the application workflow, and the PDF document generation using the jsPDF library.

- **Testing and Debugging**: This is a crucial phase where the entire system is tested for bugs and errors. It includes unit testing individual components and integration testing to ensure that different modules work together seamlessly.

- **Final Documentation**: This final phase involves compiling all project details, including the analysis, design diagrams, code snippets, and results, into the final project report or "black book."

The horizontal bars in the chart represent the duration of each task. Dependencies are shown where one task cannot begin until a preceding task is completed. This visualization ensures a systematic and organized approach to project completion.



# Chapter 4: Implementation and Testing

## 4.1 Implementation:

The implementation phase of InternQuest involved translating the system's design into a functional web application. This process focused on writing and integrating the code that makes the platform work as intended. The project was built using a client-side approach, where most of the logic resides in the script.js file, communicating with a serverless backend powered by **Firebase**.

The **front-end** was developed using HTML and CSS to create the user interface for each of the three dashboards: student, company, and admin. The **backend** integration was handled through Firebase, which provided key services:

- **Firebase Authentication** manages user registration and logins for all three user types. This service securely handles user credentials and provides a unique user ID (UID) for each user.
- **Cloud Firestore** serves as the primary database, storing all application data in collections. The project utilizes collections for users (which includes student, company, and admin profiles), internships, and payments. Data is retrieved and updated using Firestore's SDK, which is integrated directly into the script.js file.

A key part of the implementation was the use of **real-time listeners** (onSnapshot) to ensure that data on all dashboards is constantly updated without requiring a page refresh. This feature is particularly important for tracking application statuses and viewing new applications.

Another significant implementation detail was the use of **atomic transactions** (runTransaction) for critical processes, such as a student submitting an application. This ensures that all related database updates happen simultaneously or not at all, preventing data inconsistencies.

Finally, the **PDF generation** feature was implemented using the jsPDF library. This client-side approach allows for the creation and download of official documents like offer letters, certificates, and payment receipts directly in the user's browser, eliminating the need for server-side processing.

## Codes:

### index.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>InternQuest - Find Your Internship</title>
    <link rel="icon" href="https://placehold.co/32x32/2a6fdb/ffffff?text=IQ" type="image/x-icon">
    <link
href="https://fonts.googleapis.com/css2?family=Montserrat:wght@400;600;700&family=Inter:wght@400;600;700&display
=swap" rel="stylesheet">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.5.1/css/all.min.css">
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <header class="navbar">
        <a href="index.html" class="logo">
            <img src="logo.jpg" alt="InternQuest Logo" style="height: 40px; border-radius: 8px;">
        </a>
        <nav>
            <a href="#features">Features</a>
            <a href="#companies">Companies</a>
            <a href="#faq">FAQ</a>
            <a href="#contact">Contact</a>
            <a href="student.html" class="btn btn-primary">Student Login</a>
            <a href="company.html" class="btn btn-secondary">Company Login</a>
        </nav>
    </header>

    <main>
        <div id="messageContainer" class="message-container"></div>
        <div id="loadingOverlay" class="loading-overlay" style="display: none;">
            <div class="spinner"></div>
        </div>

        <section class="hero-section">
            <div class="hero-content">
                <h1 class="font-montserrat">Unlock Your Future with a 15-Day Internship</h1>
                <p>InternQuest connects 12th-pass students with valuable internships to gain real-world experience and boost
their career.</p>
                <div class="hero-actions">
                    <a href="student.html" class="btn btn-primary btn-lg">Start Your Journey</a>
                    <a href="company.html" class="btn btn-secondary btn-lg">Hire Interns</a>
                </div>
            </div>
        </section>

        <section id="features" class="section">
            <h2 class="section-title">Why Choose InternQuest?</h2>
            <div class="card-container">
                <div class="card floating-card">
                    <i class="fas fa-graduation-cap fa-2x"></i>
                    <h3>For Students</h3>
                    <ul>
                        <li><i class="fas fa-check-circle"></i> Simple application process.</li>
                        <li><i class="fas fa-check-circle"></i> Aptitude test to showcase skills.</li>
                        <li><i class="fas fa-check-circle"></i> Track your application status.</li>
                        <li><i class="fas fa-check-circle"></i> Instant offer letters and certificates.</li>
                    </ul>
                </div>
                <div class="card floating-card">
                    <i class="fas fa-building fa-2x"></i>
                    <h3>For Companies</h3>
                    <ul>
                        <li><i class="fas fa-check-circle"></i> Access a pool of talented students.</li>
                        <li><i class="fas fa-check-circle"></i> Easy application management.</li>
                        <li><i class="fas fa-check-circle"></i> Streamlined hiring process.</li>
                        <li><i class="fas fa-check-circle"></i> Manage current and past interns.</li>
                    </ul>
                </div>
                <div class="card floating-card">
                    <i class="fas fa-chart-line fa-2x"></i>
                    <h3>For Admins</h3>
                    <ul>
                        <li><i class="fas fa-check-circle"></i> Full control over the platform.</li>
                        <li><i class="fas fa-check-circle"></i> Manage users, internships, and payments.</li>
                        <li><i class="fas fa-check-circle"></i> CMS for updating website content.</li>
                        <li><i class="fas fa-check-circle"></i> Re-generate documents as needed.</li>
                    </ul>
                </div>
            </div>
        </section>

        <section id="companies" class="section bg-light">
            <h2 class="section-title">Our Partner Companies</h2>
            <p>Join these companies and kickstart your career.</p>
            <div class="companies-grid" id="companyList"></div>
        </section>

        <section id="faq" class="section">
            <h2 class="section-title">Frequently Asked Questions</h2>
            <div id="faqContainer" class="faq-container"></div>
        </section>

        <section id="contact" class="section bg-light">
            <h2 class="section-title">Get In Touch</h2>
            <p>Have questions? We're here to help!</p>
            <form class="contact-form" id="contactForm">
                <input type="text" id="contactName" placeholder="Your Name" required>
                <input type="email" id="contactEmail" placeholder="Your Email" required>
                <textarea id="contactMessage" placeholder="Your Message" rows="5" required></textarea>
                <button type="submit" class="btn btn-primary">Send Message</button>
            </form>
        </section>
    </main>

    <footer class="footer">
        <p>&copy; 2024 InternQuest. All rights reserved.</p>
        <button id="adminLoginBtn" class="btn btn-link">Admin Login</button>
    </footer>

    <div id="adminLoginModal" class="modal-overlay" style="display: none;">
        <div class="modal card">
            <h2>Admin Login</h2>
            <form id="adminAuthForm">
                <input type="email" id="adminEmail" placeholder="Email" required>
                <input type="password" id="adminPassword" placeholder="Password" required>
                <button type="submit" class="btn btn-primary btn-full-width">Login</button>
            </form>
            <button id="closeModalBtn" class="modal-close-btn">&times;</button>
        </div>
    </div>

    <script src="script.js" type="module"></script>
```

```
</body>
</html>
```

**style.css:**

```
/*
 * General Layout and Typography
 * These styles define the global look and feel, including colors, fonts, and box sizing.
*/
:root {
   --primary-color: #2a6fdb;
   --primary-hover: #1e4d9c;
   --secondary-color: #f1f5f9;
   --secondary-hover: #e2e8f0;
   --accent-color: #fca5a5;
   --text-color: #334155;
   --bg-light: #f8fafc;
   --bg-dark: #1e293b;
   --card-bg: #fff;
   --border-radius: 0.75rem;
   --shadow: 0 4px 6px rgba(0, 0, 0, 0.05), 0 1px 3px rgba(0, 0, 0, 0.025);
   --transition-speed: 0.2s;
}

* {
   box-sizing: border-box;
   margin: 0;
   padding: 0;
   font-family: 'Inter', sans-serif;
}

body {
   background-color: var(--bg-light);
   color: var(--text-color);
   line-height: 1.6;
   display: flex;
   flex-direction: column;
   min-height: 100vh;
}

main {
   flex-grow: 1;
}

a {
   color: var(--primary-color);
   text-decoration: none;
   transition: color var(--transition-speed) ease-in-out;
}

a:hover {
   color: var(--primary-hover);
}

/*
 * Navigation Bar
 * Defines the sticky header with logo and navigation links.
*/
```

```
.navbar {
    display: flex;
    justify-content: space-between;
    align-items: center;
    padding: 1rem 5%;
    background-color: var(--card-bg);
    box-shadow: var(--shadow);
    position: sticky;
    top: 0;
    z-index: 1000;
}

/* Update the .logo class to accommodate the new image-based logo */
.logo {
    font-weight: 700;
    font-size: 1.5rem;
    color: var(--primary-color);
    display: flex;
    align-items: center;
}

.logo img {
    height: 40px;
    border-radius: 8px;
}

.navbar nav {
    display: flex;
    align-items: center;
    gap: 1rem;
}

.user-name {
    font-weight: 600;
    color: var(--primary-color);
    margin-right: 1rem;
}

/*
 * Buttons
 * A set of reusable button styles.
*/
.btn {
    border: none;
    cursor: pointer;
    font-weight: 600;
    padding: 0.75rem 1.5rem;
    border-radius: var(--border-radius);
    transition: all var(--transition-speed) ease-in-out;
    text-align: center;
}
```

## script.js:

```
// Firebase SDKs
import { initializeApp } from "https://www.gstatic.com/firebasejs/11.6.1/firebase-app.js";
import { getAuth, signInWithCustomToken, signInWithEmailAndPassword, createUserWithEmailAndPassword,
onAuthStateChanged, signOut, signInAnonymously } from "https://www.gstatic.com/firebasejs/11.6.1/firebase-auth.js";
```

```
import { getFirestore, doc, getDoc, setDoc, updateDoc, collection, query, where, getDocs, onSnapshot, addDoc, deleteDoc,
runTransaction, deleteField } from "https://www.gstatic.com/firebasejs/11.6.1/firebase-firestore.js";

// Global variables provided by the platform
const appId = typeof __app_id !== 'undefined' ? __app_id : 'default-app-id';
const firebaseConfig = {
    apiKey: "AIzaSyA1w7fPrXlw_I9tZYvzBTOfLKoJvjNrjPY",
    authDomain: "krish-11166.firebaseapp.com",
    projectId: "krish-11166",
    storageBucket: "krish-11166.firebasestorage.app",
    messagingSenderId: "587101854752",
    appId: "1:587101854752:web:aa5e5e503fde6af3aaf045",
    measurementId: "G-BVV6JDQQYT"
};
const initialAuthToken = typeof __initial_auth_token !== 'undefined' ? __initial_auth_token : null;

// Initialize Firebase
const app = initializeApp(firebaseConfig);
const auth = getAuth(app);
const db = getFirestore(app);

let userId = null;
let userRole = null;
let userDocData = null;

// Admin UID - Hardcoded for a single admin user
const ADMIN_UID = 'sSGcrEx3vmcZHPbGCDpo9i8eQQG3';

// --- UTILITY FUNCTIONS ---
function showMessage(msg, type = 'success') {
    const container = document.getElementById('messageContainer');
    if (!container) return;
    const messageEl = document.createElement('div');
    messageEl.className = `message ${type}`;
    messageEl.textContent = msg;
    container.appendChild(messageEl);
    setTimeout(() => {
        messageEl.style.opacity = '0';
        messageEl.style.transform = 'translateY(-20px)';
        setTimeout(() => messageEl.remove(), 300);
    }, 3000);
}

function showSection(sectionId) {
    const sections = document.querySelectorAll('.dashboard-section');
    sections.forEach(sec => {
        sec.style.display = 'none';
        sec.classList.remove('active');
    });
    const sectionToShow = document.getElementById(sectionId);
    if (sectionToShow) {
        sectionToShow.style.display = 'block';
        sectionToShow.classList.add('active');
    }

    const sidebarBtns = document.querySelectorAll('.sidebar-btn');
    sidebarBtns.forEach(btn => btn.classList.remove('active'));
    const activeBtn = document.querySelector(`.sidebar-btn[data-section="${sectionId}"]`);
    if (activeBtn) {
        activeBtn.classList.add('active');
    }
}

function showLoadingOverlay() {
    const overlay = document.getElementById('loadingOverlay');
    if (overlay) {
        overlay.style.display = 'flex';
    }
}

function hideLoadingOverlay() {
    const overlay = document.getElementById('loadingOverlay');
    if (overlay) {
        overlay.style.display = 'none';
    }
}

// Function to validate student age
function isAgeEligible(dob) {
    const today = new Date();
    const birthDate = new Date(dob);
    let age = today.getFullYear() - birthDate.getFullYear();
    const m = today.getMonth() - birthDate.getMonth();
    if (m < 0 || (m === 0 && today.getDate() < birthDate.getDate())) {
        age--;
    }
    return age >= 15;
}

// --- AUTHENTICATION & INITIALIZATION ---
async function handleAuth() {
    showLoadingOverlay();
    if (initialAuthToken) {
        try {
            await signInWithCustomToken(auth, initialAuthToken);
        } catch (error) {
            console.error("Custom token sign-in failed.", error);
        }
    }
    hideLoadingOverlay();
}
```

## 4.2 Testing:

Testing is an essential part of the development process to ensure the system is reliable, bug-free, and meets all specified requirements. The testing for InternQuest was conducted in several stages:

- **Unit Testing**: Individual functions and components within the script.js file were tested to ensure they performed as expected. For example, the isAgeEligible() function was tested with various dates of birth to verify its correctness, and the setupAptitudeTest() function was tested to confirm that the scoring logic was accurate.

- **Integration Testing**: This phase focused on testing the communication between different components of the system. For example, the profile save operation was tested to ensure that the data from the HTML form was correctly written to the Firestore database. Similarly, the application submission process was tested to confirm that a single action simultaneously updated the records of the student, company, and admin.

- **System Testing**: The entire system was tested end-to-end for each user role to ensure the complete workflow functioned smoothly. This included testing the full cycle from student registration and aptitude test completion to company approval and certificate download. The different user roles were tested to ensure that they could only access the data they were authorized to see.

- **User Acceptance Testing (UAT)**: The final stage involved testing the system with potential end-users to gather feedback on its usability and performance. Minor bugs related to the user interface and user experience were identified and fixed based on this feedback.

Through these rigorous testing stages, the InternQuest system was validated as a reliable, secure, and user-friendly platform that successfully meets its project objectives.

# Chapter 5: Results

The InternQuest project was successfully implemented to create a functional web platform that meets its objectives of connecting students and companies. The final product is a single-page application that dynamically renders content for three different user types: student, company, and admin. The user interface is clean and organized, allowing for intuitive navigation as defined in the style.css file.

A key result is the successful implementation of the **student dashboard**, which guides students through their internship journey. Students can log in and manage their profile, including mandatory fields like contact information, academic details, and bank account information. The system effectively manages the aptitude test, restricting access to internships until a qualifying score of 12 out of 15 is achieved within the time limit. The platform also correctly tracks application statuses in real-time and allows students to complete a simulated payment of ▪149.

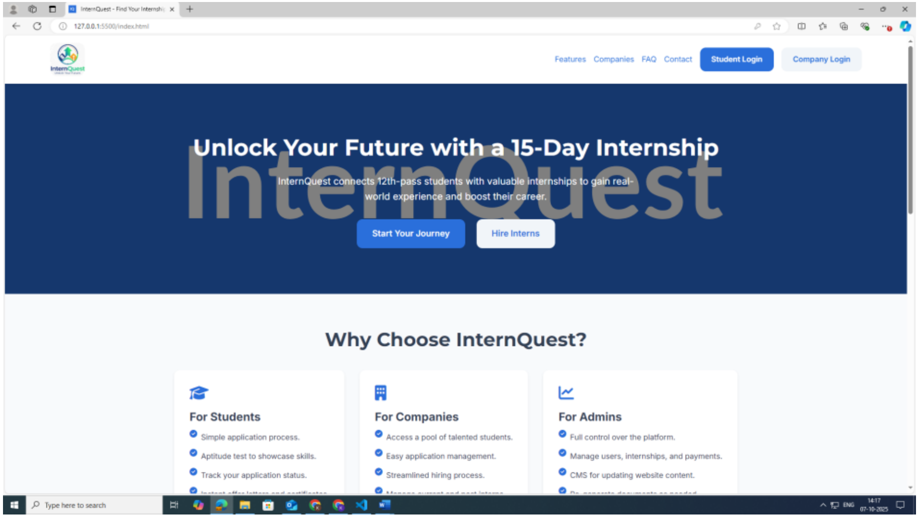
The **company and admin dashboards** were also successfully developed, providing a streamlined experience for managing internships and users. Companies can easily post new internship opportunities and review applications. The application status for both the student and the company updates in real-time, showcasing the effectiveness of the Firebase Firestore integration. The admin dashboard provides a complete overview of all users, internships, and applications, and includes a functional CMS for updating the website's content.

A significant achievement of the project is the client-side document generation feature. The platform can successfully create and download PDF documents, including **offer letters, payment receipts, and certificates of completion**, using the jsPDF library. This provides a tangible and valuable outcome for students.

The final system demonstrates that a robust and scalable internship platform can be built using a serverless architecture with Firebase, a modern web development approach that prioritizes efficiency and real-time functionality.

# Screenshots of Website:

## Homepage:



Unlock Your Future with a 15-Day Internship

InternQuest connects 12th-pass students with valuable internships to gain real-world experience and boost their career.

Why Choose InternQuest?

For Students

- Simple application process.
- Aptitude test to showcase skills.
- Track your application status.

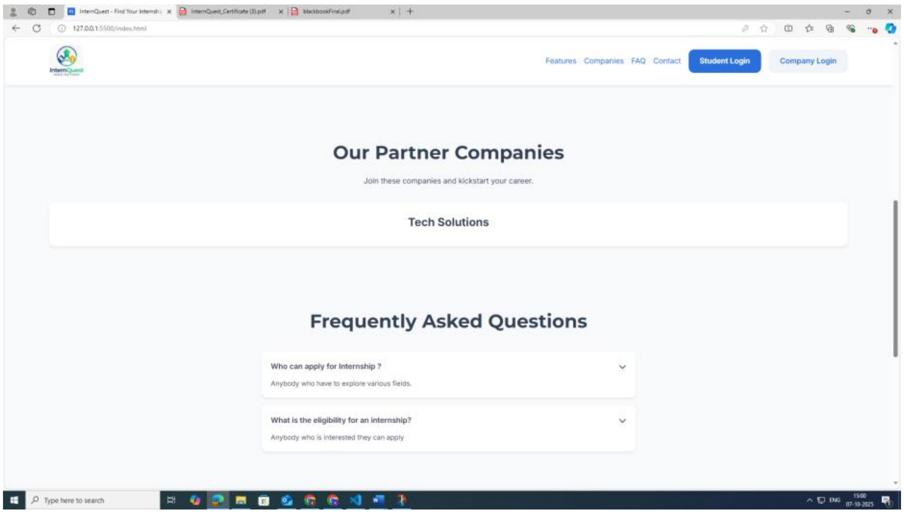
For Companies

- Access a pool of talented students.
- Easy application management.
- Streamlined hiring process.

For Admins

- Full control over the platform.
- Manage users, internships, and payments.
- CMS for updating website content.

## Partner Companies and FAQ Section:



Our Partner Companies

Join these companies and kickstart your career.

Tech Solutions

Frequently Asked Questions

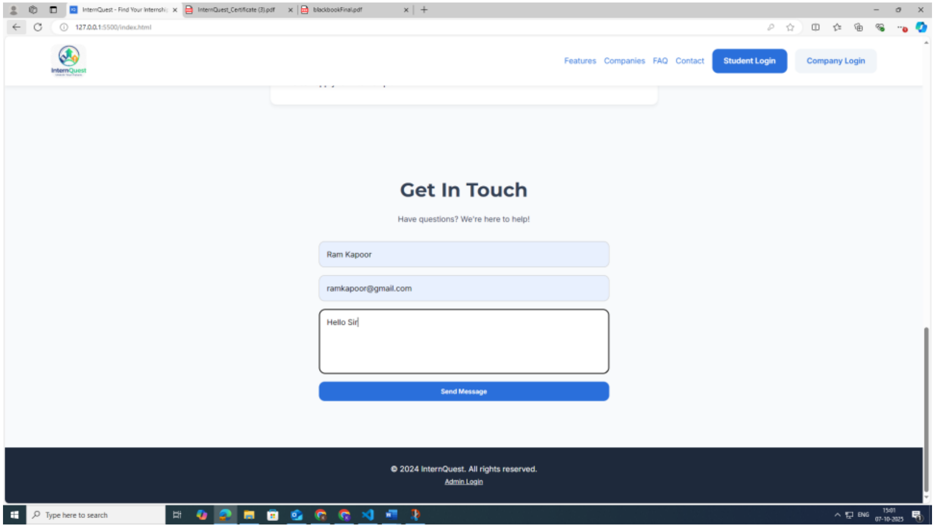
Who can apply for Internship ?

Anybody who have to explore various fields.

What is the eligibility for an internship?

Anybody who is interested they can apply

## Contact Us:



Get In Touch

Have questions? We're here to help!

Ram Kapoor

ramkapoor@gmail.com

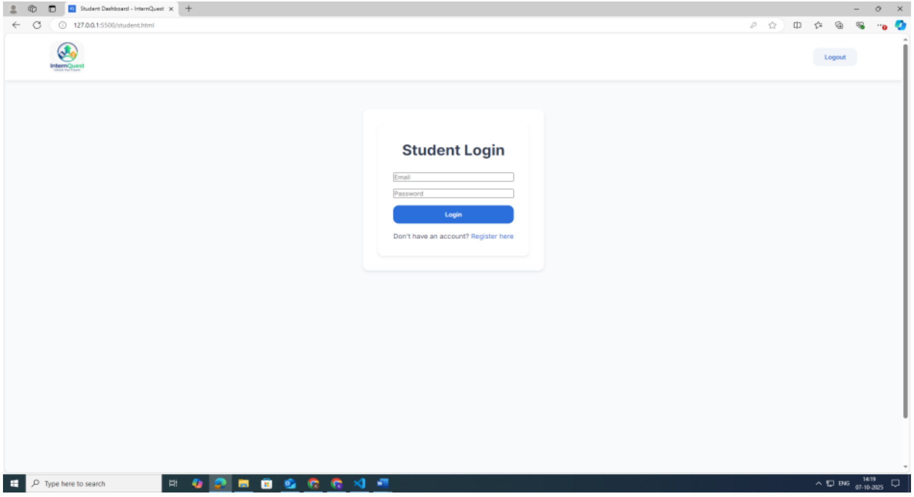
Hello Sir

© 2024 InternQuest. All rights reserved.

Admin Login

## Student Dashboard:

### Student Login:



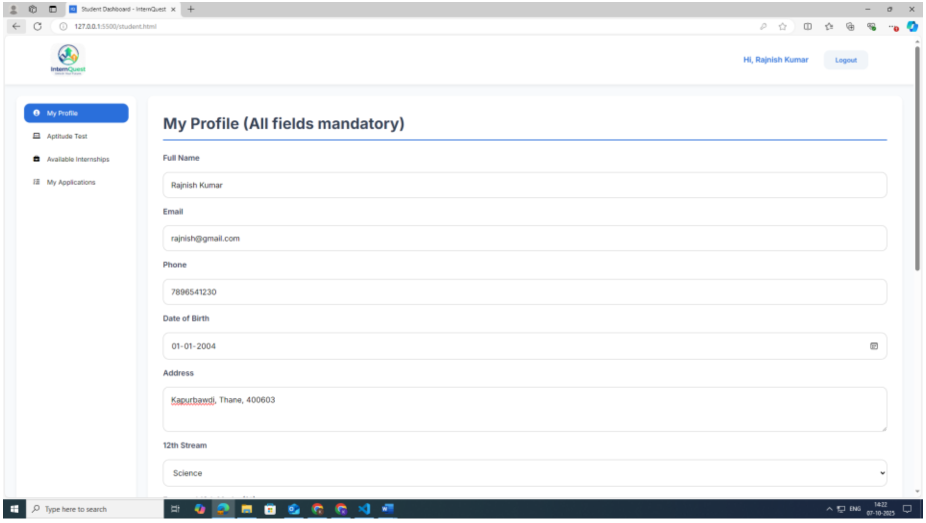
Student Login

Email

Password

Don't have an account? Register here

### Student Profile:



My Profile (All fields mandatory)

Full Name

Rajnish Kumar

Email

rajnish@gmail.com

Phone

7896541230

Date of Birth

01-01-2004

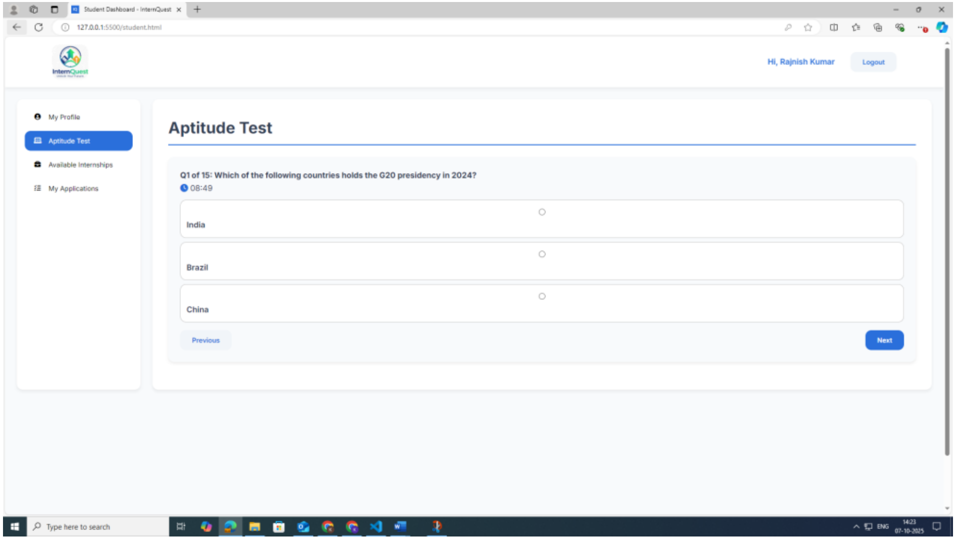
Address

Kapurbawdi, Thane, 400603

12th Stream

Science

### Aptitude Test:



Aptitude Test

Q1 of 15: Which of the following countries holds the G20 presidency in 2024?

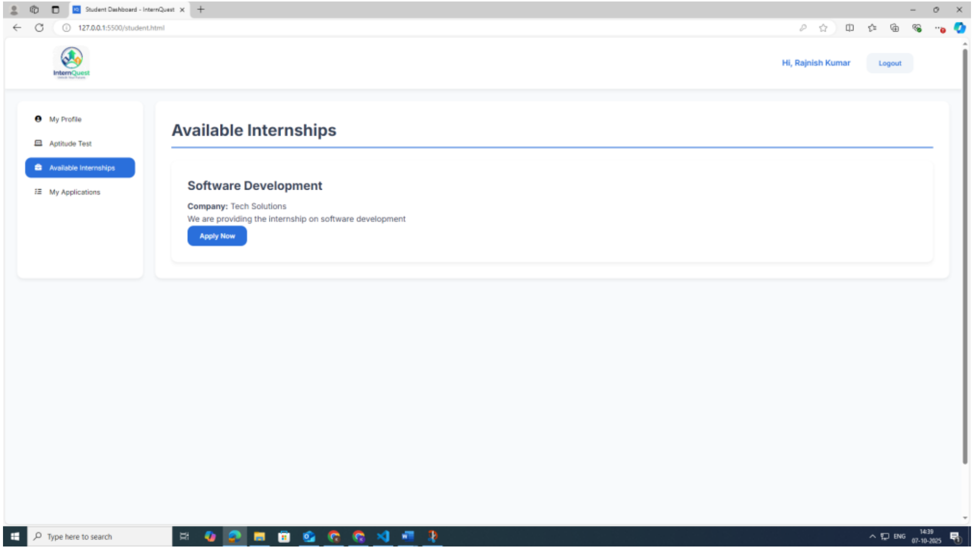
08:49

○ India

○ Brazil

○ China

## Available Internships:



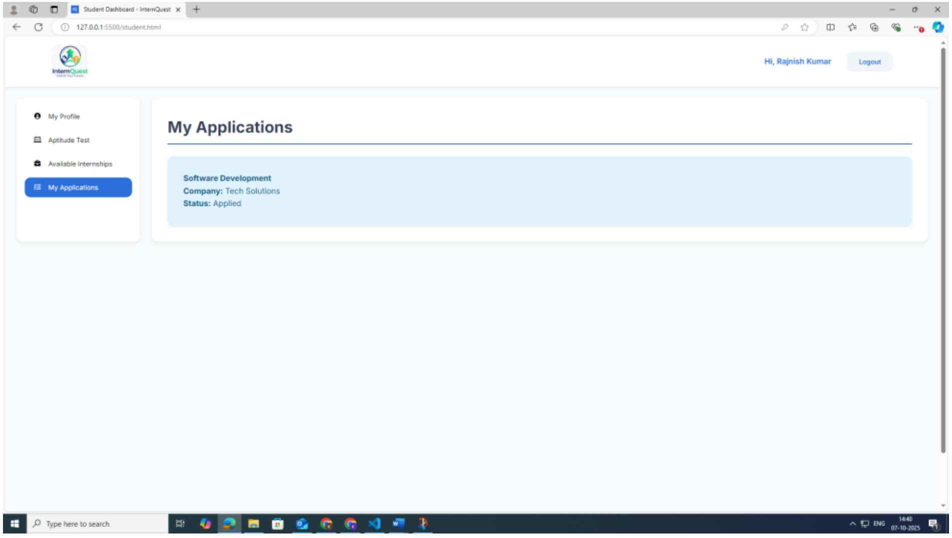
Available Internships

Software Development

**Company:** Tech Solutions

We are providing the internship on software development

## My Applications:



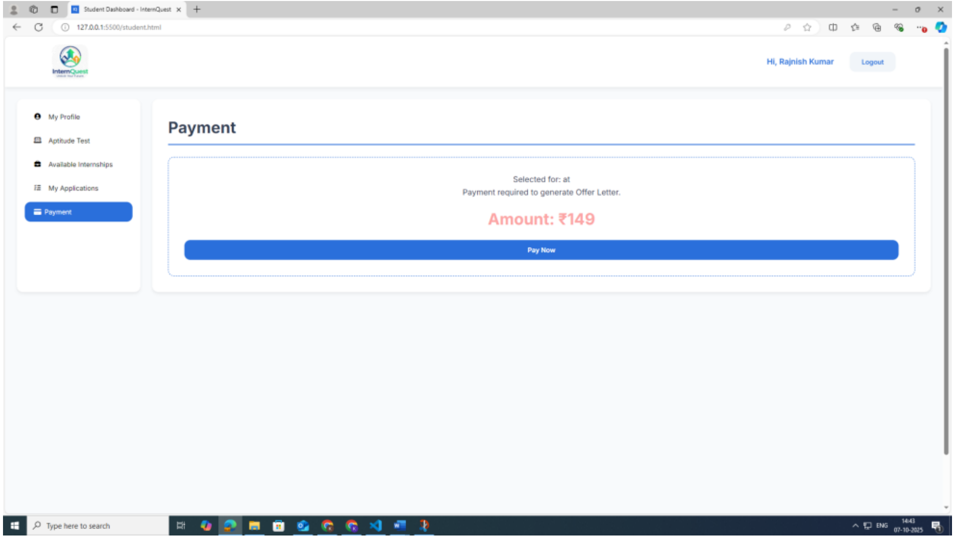
My Applications

**Software Development**

**Company:** Tech Solutions

**Status:** Applied

## Payment:



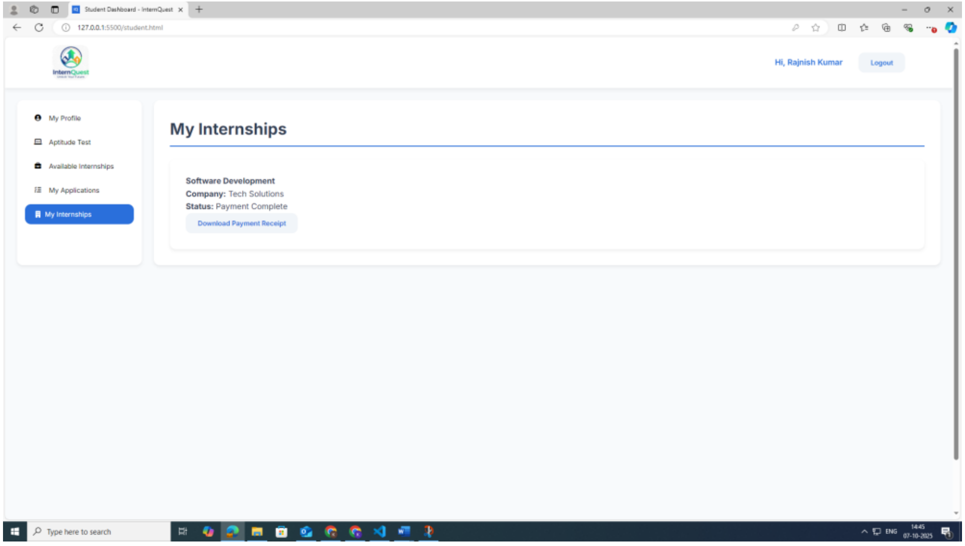
Payment

Selected for: at

Payment required to generate Offer Letter.

Amount: ₹149

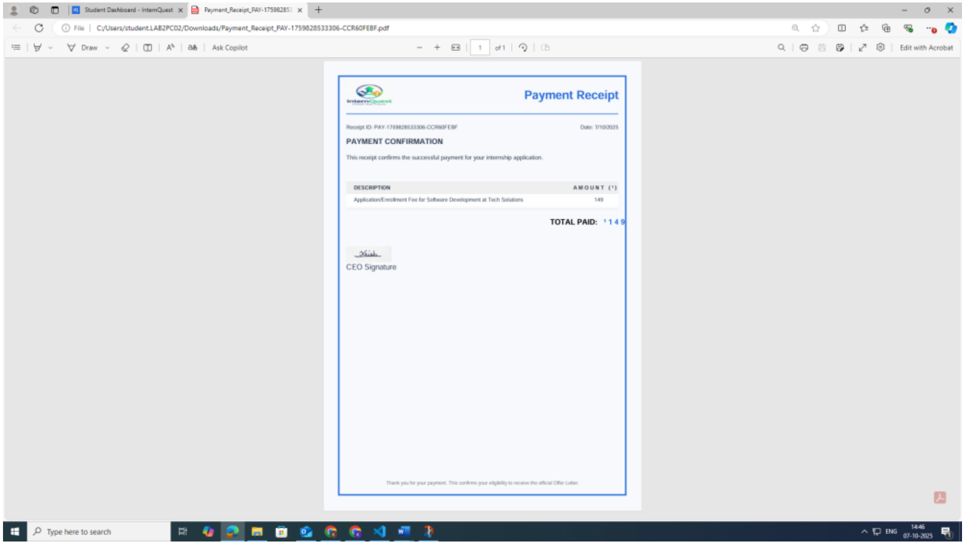
## My Internships:



My Internships

**Software Development**
**Company:** Tech Solutions
**Status:** Payment Complete

## Payment Reciept:



Student Dashboard - InternQuest
Payment_Receipt_PAY-17598285...
File | C:/Users/student.LAB2PC02/Downloads/Payment_Receipt_PAY-1759828533306-CCR60FEBF.pdf
Draw
Ask Copilot
1
of 1
Edit with Acrobat
Type here to search

Payment Receipt

Receipt ID: PAY-1759828533306-CCR60FEBF    Date: 7/10/2025

**PAYMENT CONFIRMATION**

This receipt confirms the successful payment for your internship application.

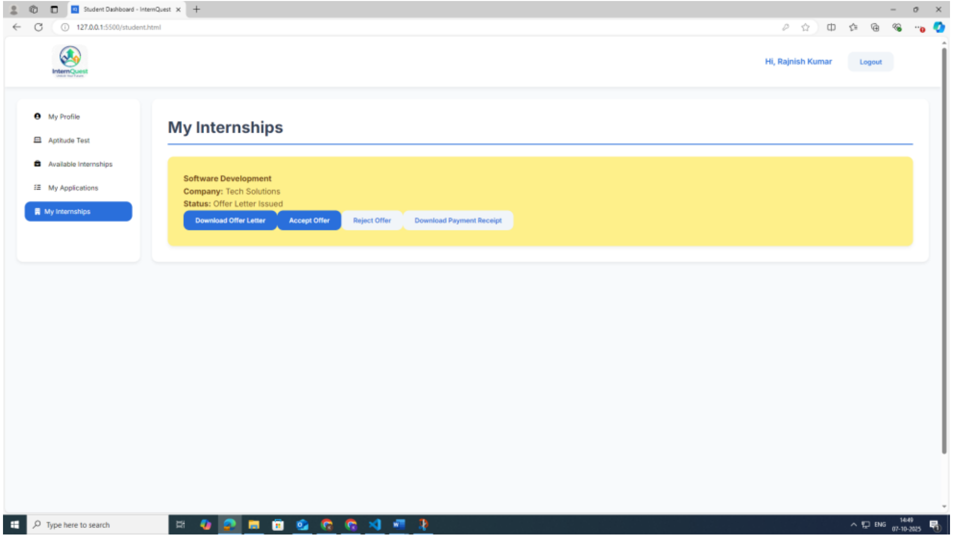
| DESCRIPTION | AMOUNT (¹) |
|---|---|
| Application/Enrollment Fee for Software Development at Tech Solutions | 149 |

**TOTAL PAID:** ¹149

CEO Signature

Thank you for your payment. This confirms your eligibility to receive the official Offer Letter.

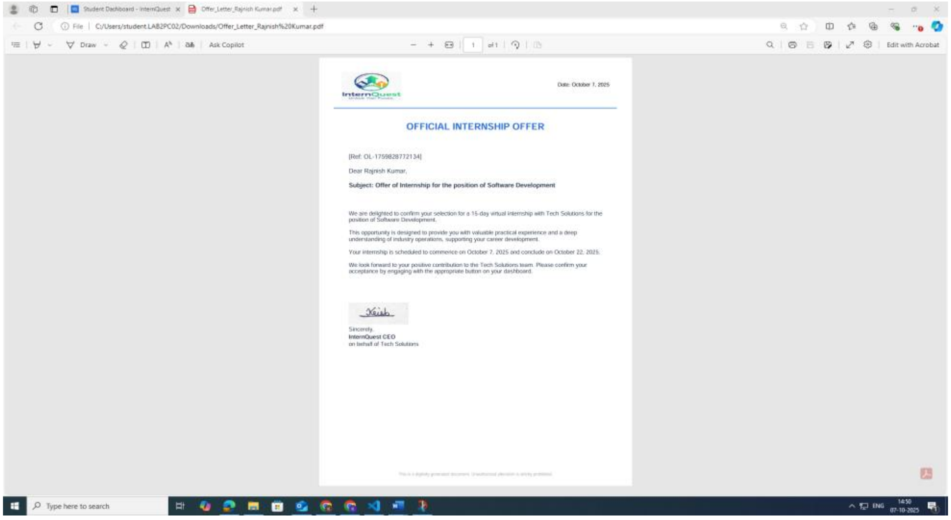
## My Internships:



My Internships

**Software Development**
**Company:** Tech Solutions
**Status:** Offer Letter Issued

## Offer Letter:



Date: October 7, 2025

OFFICIAL INTERNSHIP OFFER

[Ref: OL-1759828772134]

Dear Rajnish Kumar,

**Subject: Offer of Internship for the position of Software Development**

We are delighted to confirm your selection for a 15-day virtual internship with Tech Solutions for the position of Software Development.

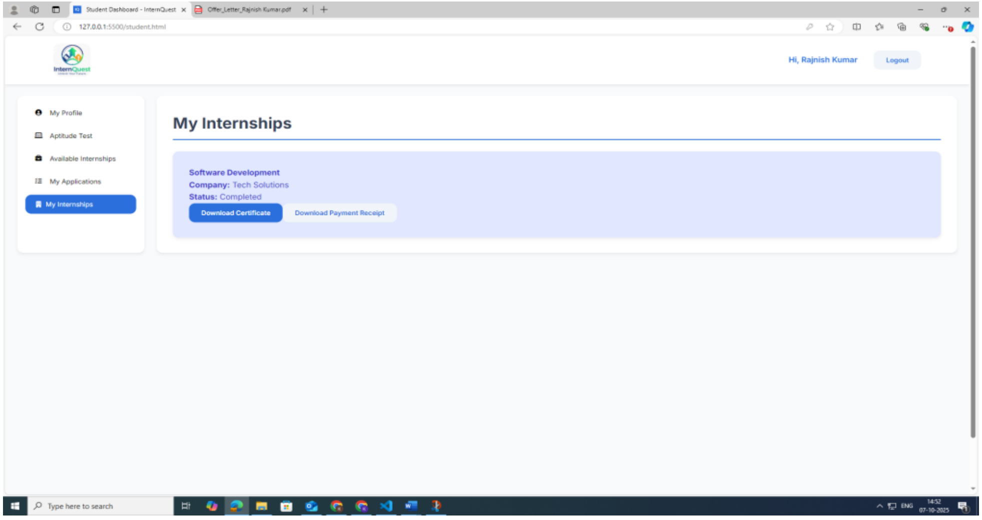
This opportunity is designed to provide you with valuable practical experience and a deep understanding of industry operations, supporting your career development.

Your internship is scheduled to commence on October 7, 2025 and conclude on October 22, 2025.

We look forward to your positive contribution to the Tech Solutions team. Please confirm your acceptance by engaging with the appropriate button on your dashboard.

Sincerely,
**InternQuest CEO**
on behalf of Tech Solutions

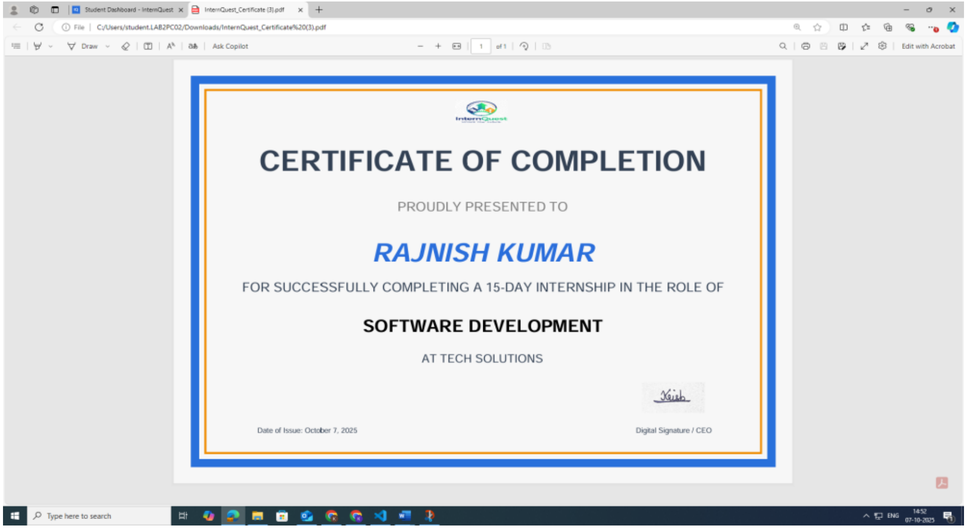
## My Internships:



My Internships

**Software Development**
**Company:** Tech Solutions
**Status:** Completed

## Certificate:



CERTIFICATE OF COMPLETION

PROUDLY PRESENTED TO

*RAJNISH KUMAR*

FOR SUCCESSFULLY COMPLETING A 15-DAY INTERNSHIP IN THE ROLE OF

**SOFTWARE DEVELOPMENT**

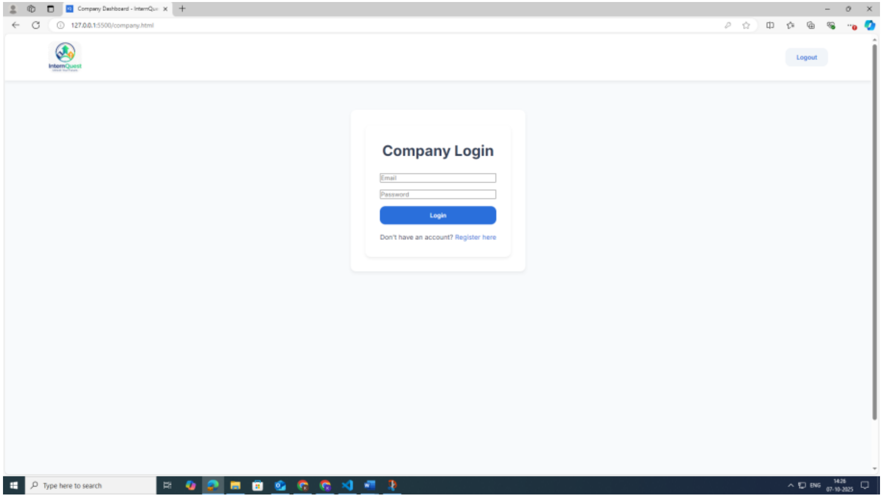
AT TECH SOLUTIONS

Date of Issue: October 7, 2025

Digital Signature / CEO

# Company:

## Company Login:



127.0.0.1:5500/company.html

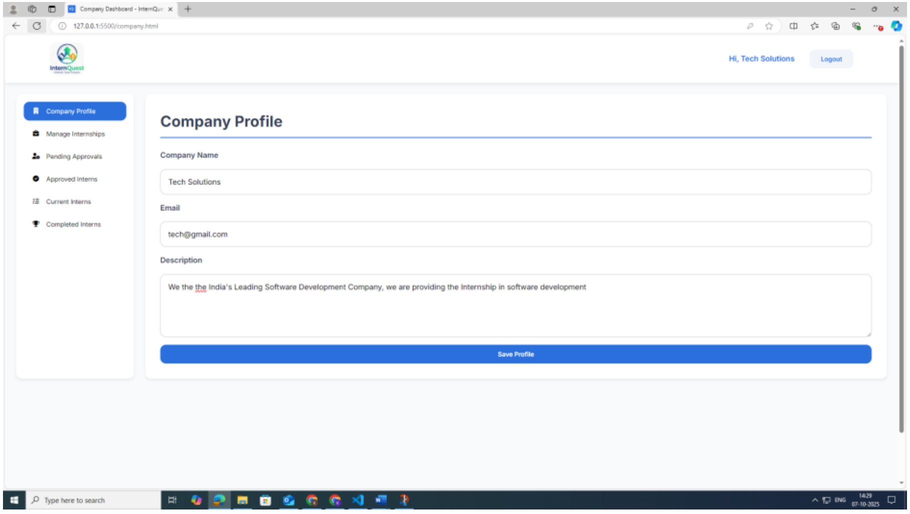
Company Login

Email

Password

Don't have an account? Register here

## Profile:



127.0.0.1:5500/company.html

Hi, Tech Solutions

Company Profile

Company Name

Tech Solutions

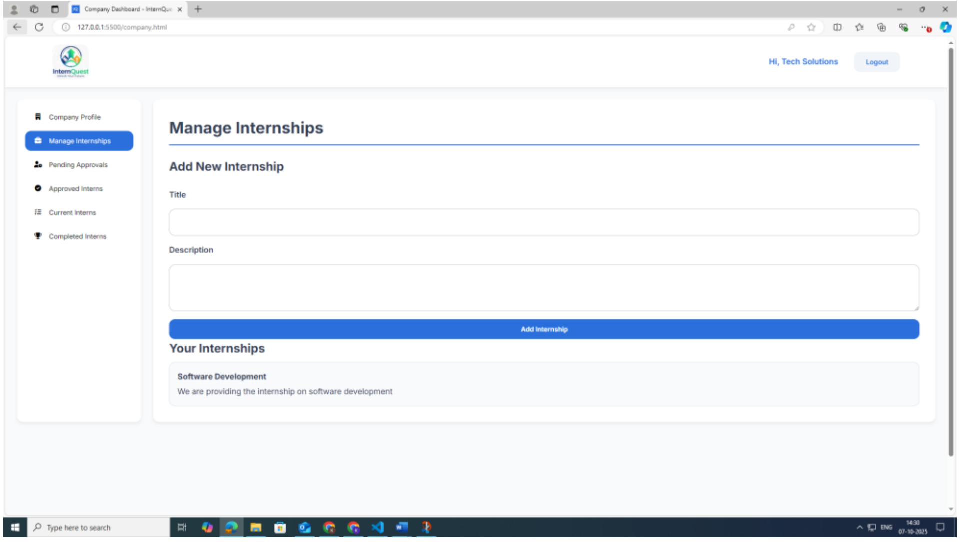
Email

tech@gmail.com

Description

We the the India's Leading Software Development Company, we are providing the Internship in software development

## Manage Internships:



127.0.0.1:5500/company.html

Hi, Tech Solutions

Manage Internships

Add New Internship

Title

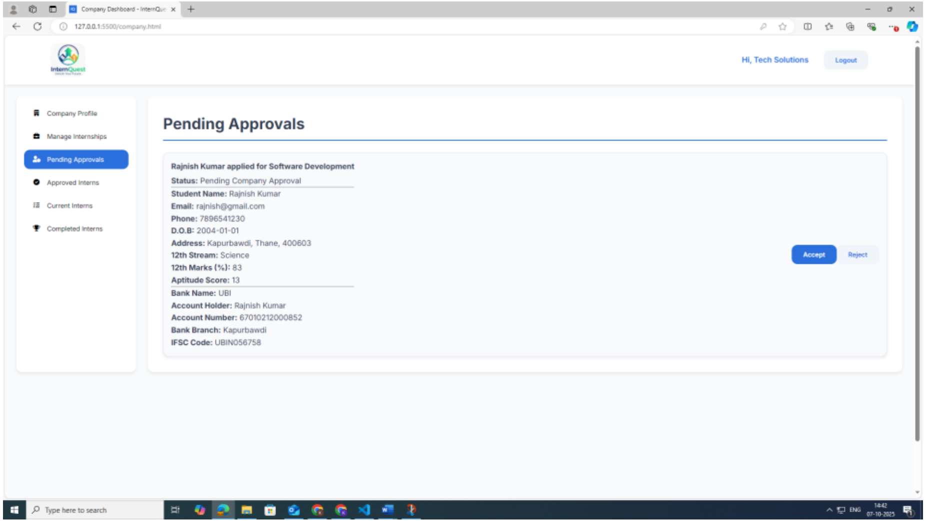
Description

Your Internships

Software Development

We are providing the internship on software development

## Pending Approvals:



Pending Approvals

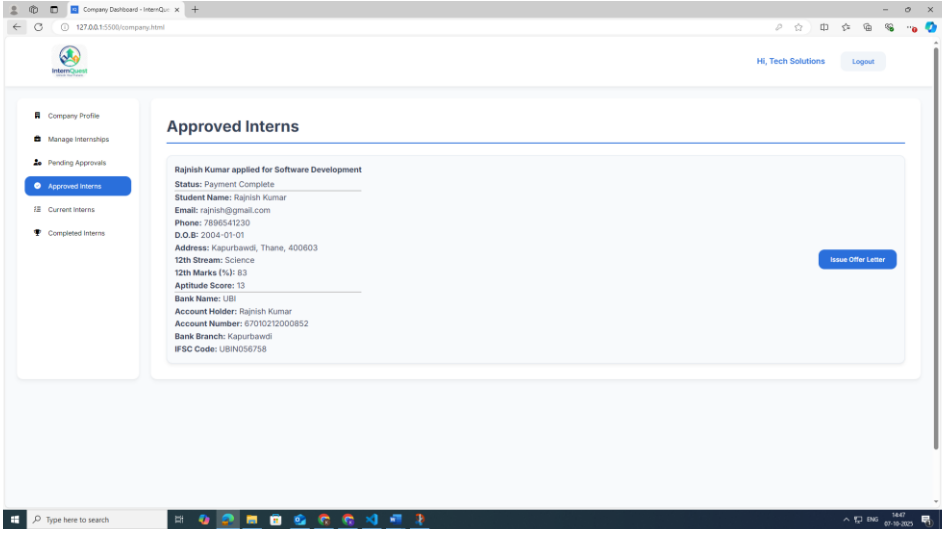
Rajnish Kumar applied for Software Development

**Status:** Pending Company Approval

**Student Name:** Rajnish Kumar
**Email:** rajnish@gmail.com
**Phone:** 7896541230
**D.O.B:** 2004-01-01
**Address:** Kapurbawdi, Thane, 400603
**12th Stream:** Science
**12th Marks (%):** 83
**Aptitude Score:** 13

**Bank Name:** UBI
**Account Holder:** Rajnish Kumar
**Account Number:** 67010212000852
**Bank Branch:** Kapurbawdi
**IFSC Code:** UBIN056758

## Approved Interns:



Approved Interns

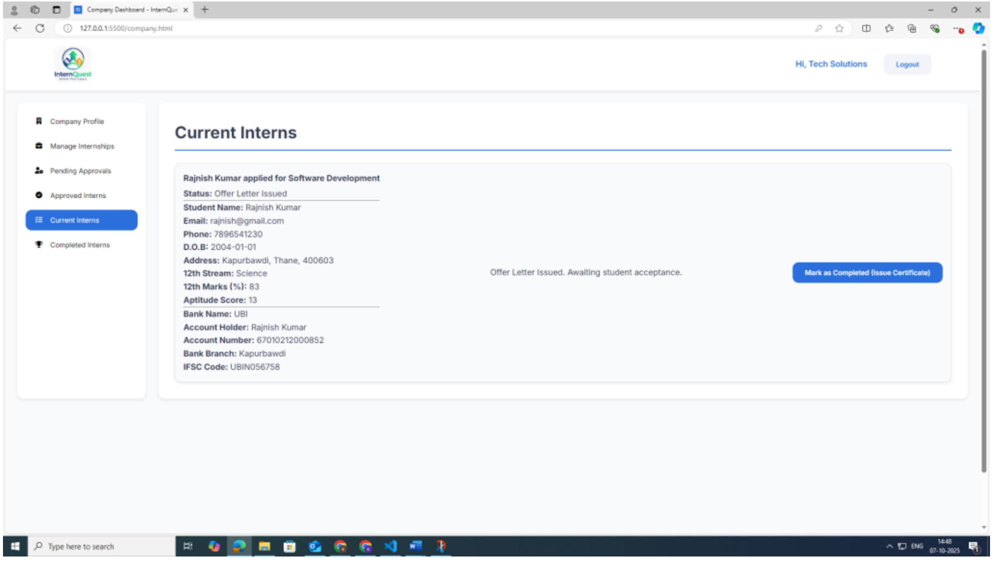
Rajnish Kumar applied for Software Development

**Status:** Payment Complete

**Student Name:** Rajnish Kumar
**Email:** rajnish@gmail.com
**Phone:** 7896541230
**D.O.B:** 2004-01-01
**Address:** Kapurbawdi, Thane, 400603
**12th Stream:** Science
**12th Marks (%):** 83
**Aptitude Score:** 13

**Bank Name:** UBI
**Account Holder:** Rajnish Kumar
**Account Number:** 67010212000852
**Bank Branch:** Kapurbawdi
**IFSC Code:** UBIN056758

## Current Interns:



Current Interns

Rajnish Kumar applied for Software Development

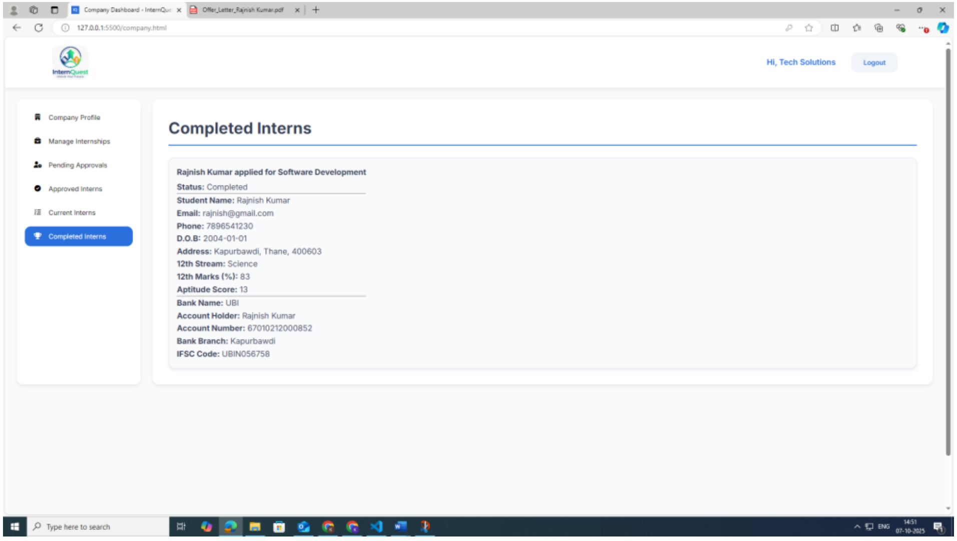
**Status:** Offer Letter Issued

**Student Name:** Rajnish Kumar
**Email:** rajnish@gmail.com
**Phone:** 7896541230
**D.O.B:** 2004-01-01
**Address:** Kapurbawdi, Thane, 400603
**12th Stream:** Science
**12th Marks (%):** 83
**Aptitude Score:** 13

**Bank Name:** UBI
**Account Holder:** Rajnish Kumar
**Account Number:** 67010212000852
**Bank Branch:** Kapurbawdi
**IFSC Code:** UBIN056758

Offer Letter Issued. Awaiting student acceptance.

## Completed Interns:



### Completed Interns

**Rajnish Kumar applied for Software Development**

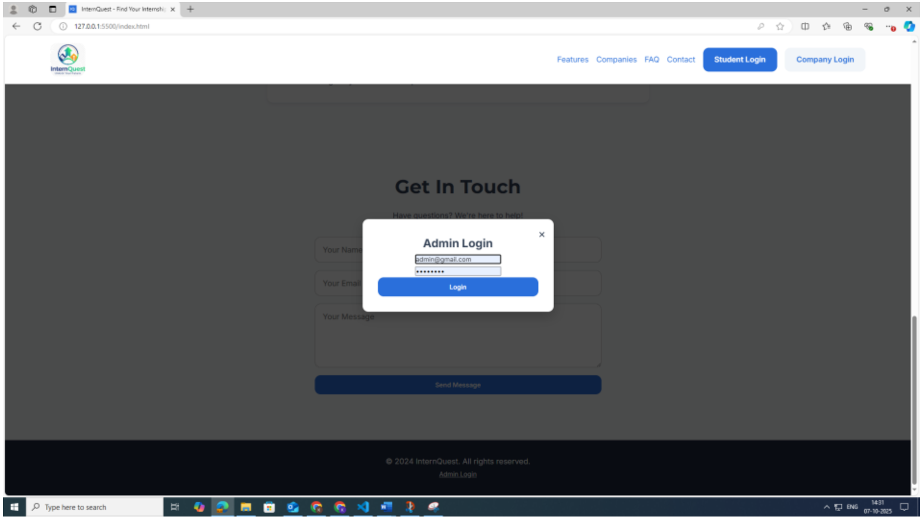
**Status:** Completed

**Student Name:** Rajnish Kumar
**Email:** rajnish@gmail.com
**Phone:** 7896541230
**D.O.B:** 2004-01-01
**Address:** Kapurbawdi, Thane, 400603
**12th Stream:** Science
**12th Marks (%):** 83
**Aptitude Score:** 13

**Bank Name:** UBI
**Account Holder:** Rajnish Kumar
**Account Number:** 67010212000852
**Bank Branch:** Kapurbawdi
**IFSC Code:** UBIN056758

# Admin:

## Admin Login:

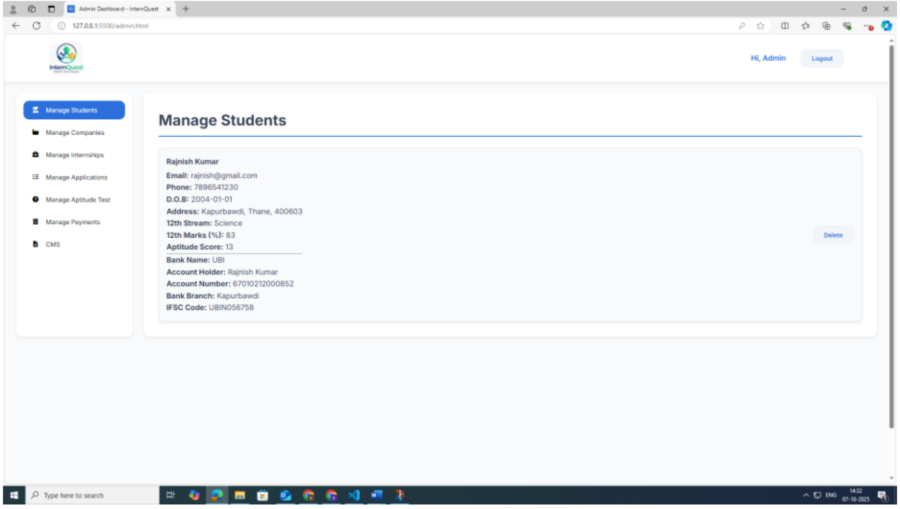


Admin Login

admin@gmail.com

••••••••

## Manage Students:

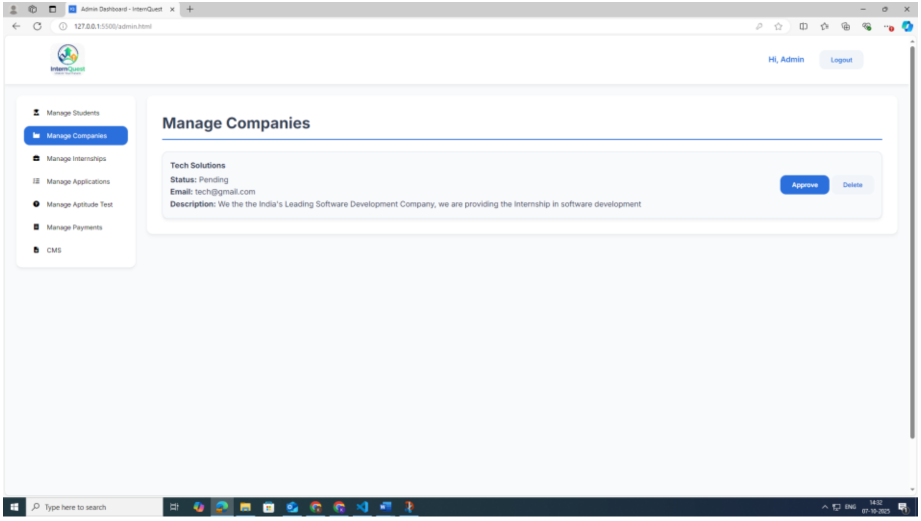


Manage Students

**Rajnish Kumar**

**Email:** rajnish@gmail.com
**Phone:** 7896541230
**D.O.B:** 2004-01-01
**Address:** Kapurbawdi, Thane, 400603
**12th Stream:** Science
**12th Marks (%):** 83
**Aptitude Score:** 13
**Bank Name:** UBI
**Account Holder:** Rajnish Kumar
**Account Number:** 67010212000852
**Bank Branch:** Kapurbawdi
**IFSC Code:** UBIN056758

## Manage Companies:

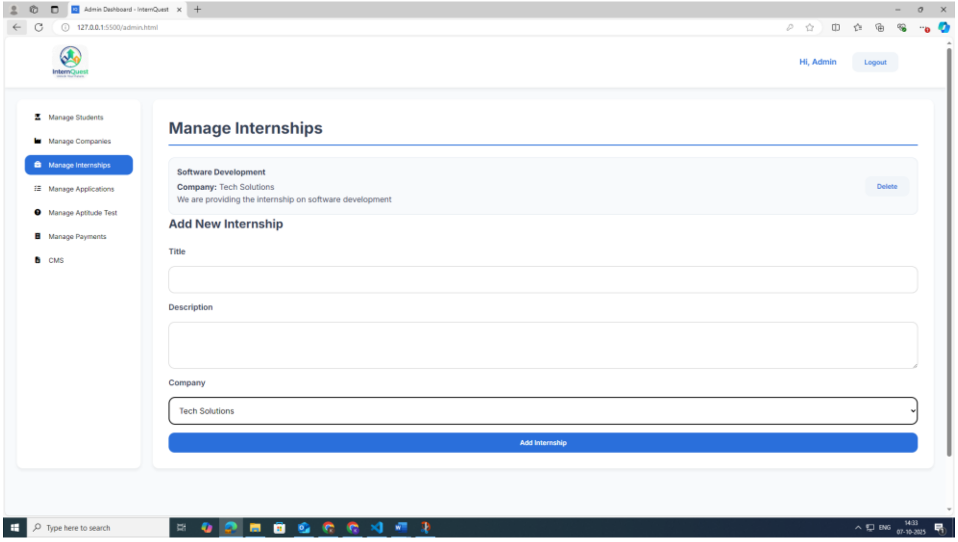


Manage Companies

**Tech Solutions**

**Status:** Pending
**Email:** tech@gmail.com
**Description:** We the the India's Leading Software Development Company, we are providing the Internship in software development

## Manage Internships:



Manage Internships

**Software Development**
**Company:** Tech Solutions
We are providing the internship on software development

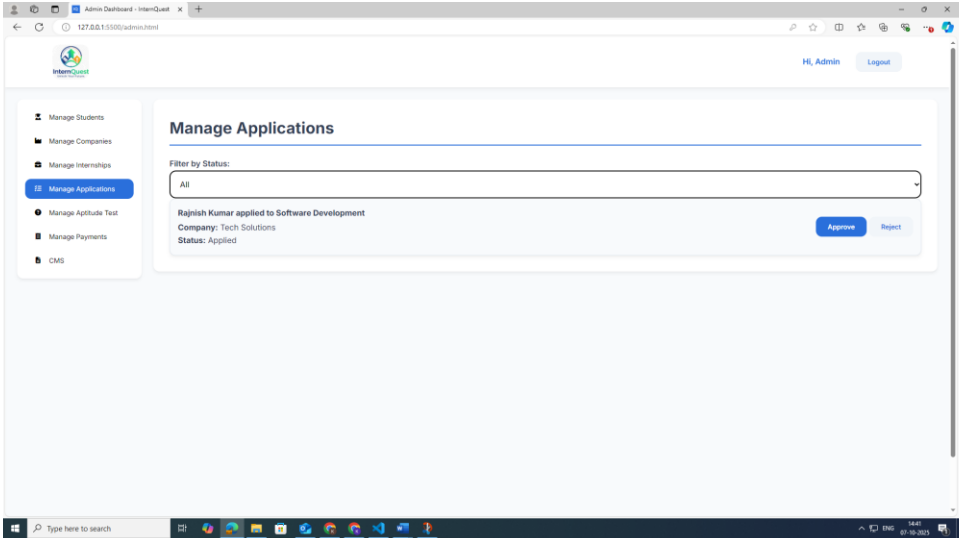
Add New Internship

Title

Description

Company: Tech Solutions

## Manage Applications:

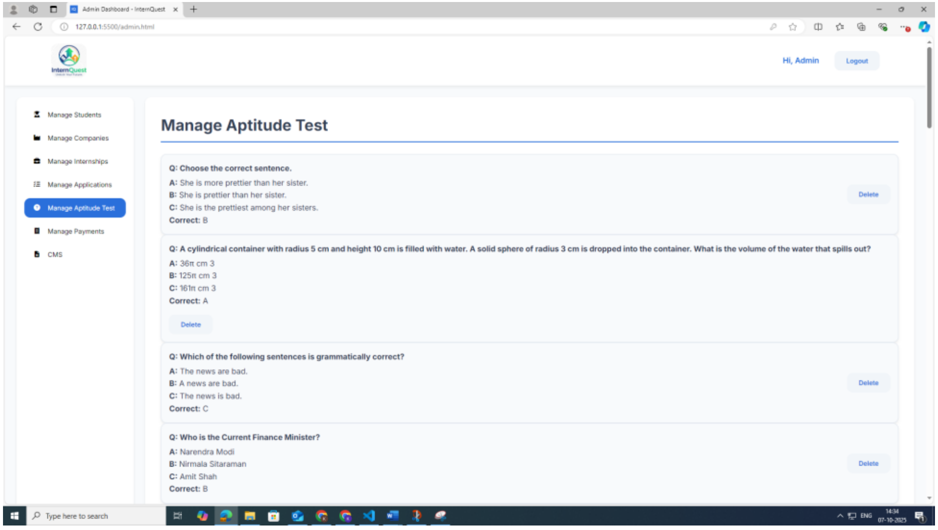


Manage Applications

Filter by Status: All

**Rajnish Kumar applied to Software Development**
**Company:** Tech Solutions
**Status:** Applied

## Manage Aptitude Test:



Manage Aptitude Test

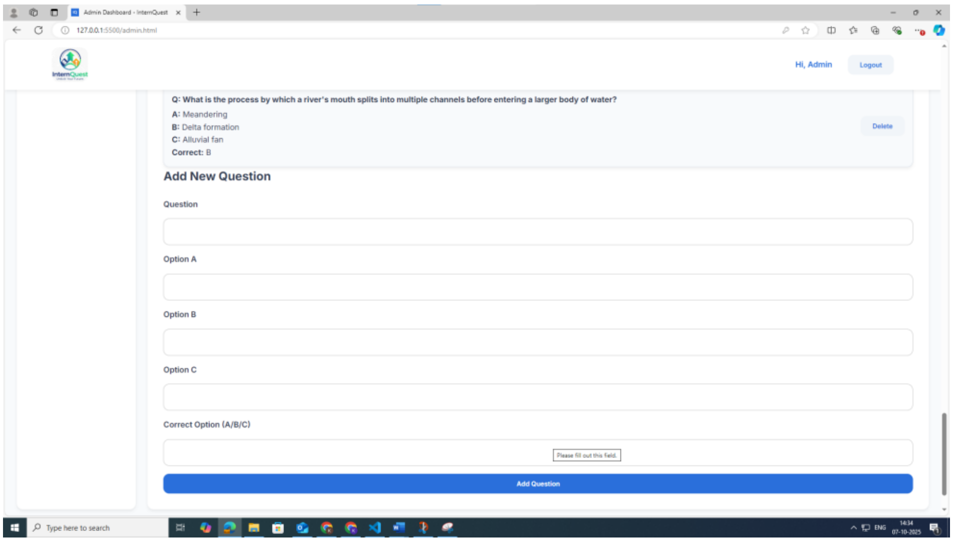
**Q: Choose the correct sentence.**
**A:** She is more prettier than her sister.
**B:** She is prettier than her sister.
**C:** She is the prettiest among her sisters.
**Correct:** B

**Q: A cylindrical container with radius 5 cm and height 10 cm is filled with water. A solid sphere of radius 3 cm is dropped into the container. What is the volume of the water that spills out?**
**A:** 36π cm 3
**B:** 125π cm 3
**C:** 161π cm 3
**Correct:** A

**Q: Which of the following sentences is grammatically correct?**
**A:** The news are bad.
**B:** A news are bad.
**C:** The news is bad.
**Correct:** C

**Q: Who is the Current Finance Minister?**
**A:** Narendra Modi
**B:** Nirmala Sitaraman
**C:** Amit Shah
**Correct:** B

## Add New Aptitude Question:



Admin Dashboard - InternQuest
127.0.0.1:5500/admin.html
Hi, Admin
Logout
Delete

**Q: What is the process by which a river's mouth splits into multiple channels before entering a larger body of water?**

**A:** Meandering

**B:** Delta formation

**C:** Alluvial fan

**Correct:** B

Add New Question

Question

Option A

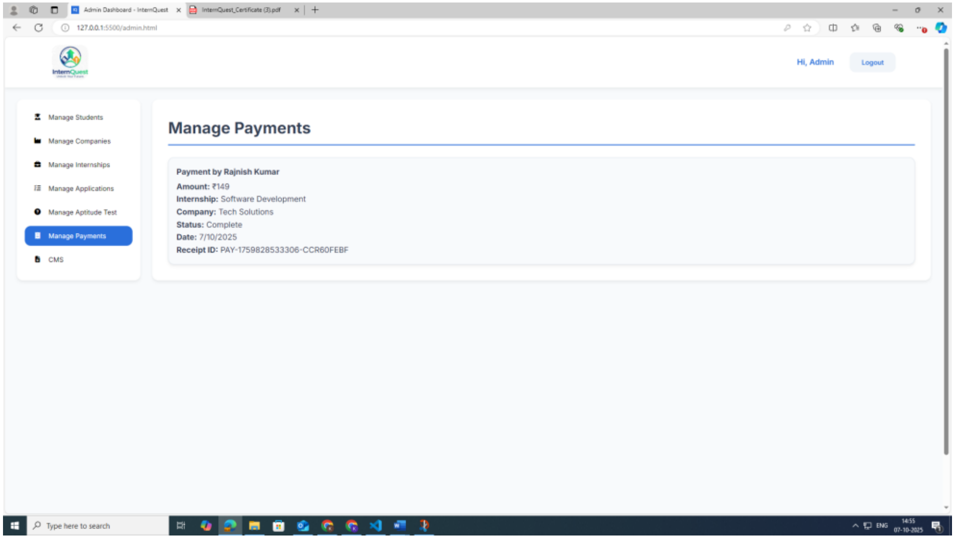
Option B

Option C

Correct Option (A/B/C)

Please fill out this field.
Add Question
Type here to search

## Manage Payments:



Manage Payments

**Payment by Rajnish Kumar**

**Amount:** ₹149

**Internship:** Software Development

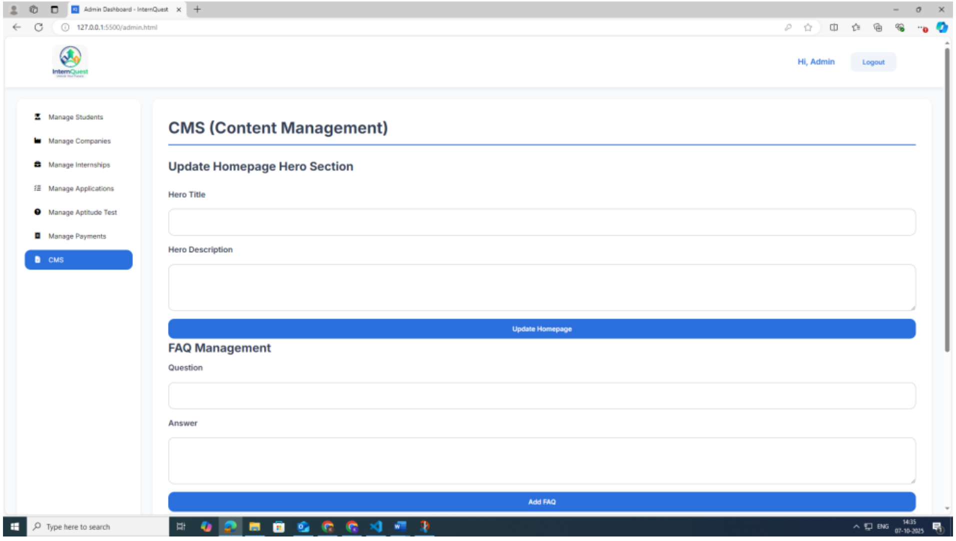
**Company:** Tech Solutions

**Status:** Complete

**Date:** 7/10/2025

**Receipt ID:** PAY-1759828533306-CCR60FEBF

## CMS:



CMS (Content Management)

Update Homepage Hero Section

Hero Title

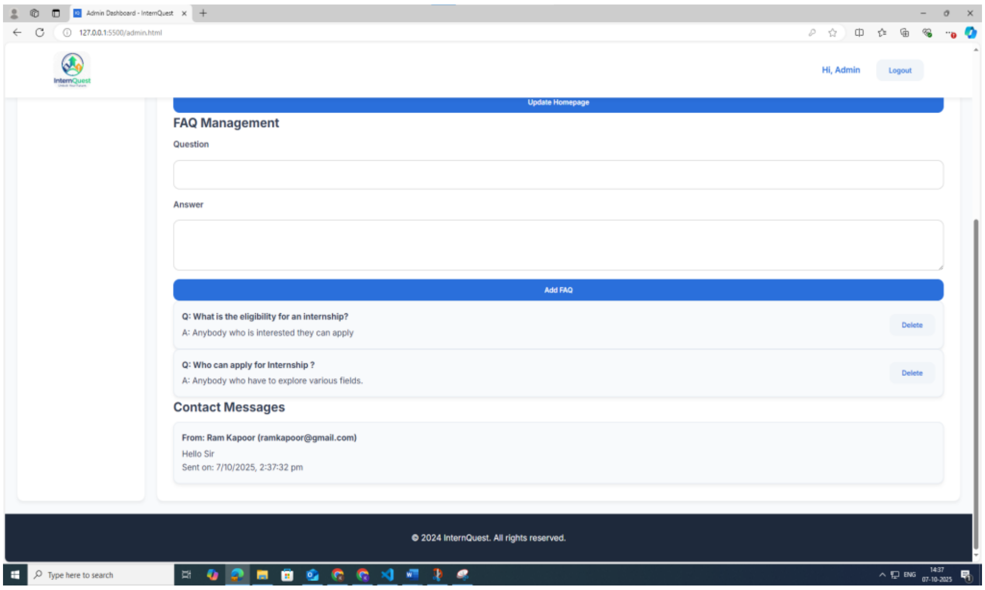
Hero Description

FAQ Management

Question

Answer

## FAQ Management and Contact Messages:



FAQ Management

Question

Answer

Q: What is the eligibility for an internship?
A: Anybody who is interested they can apply

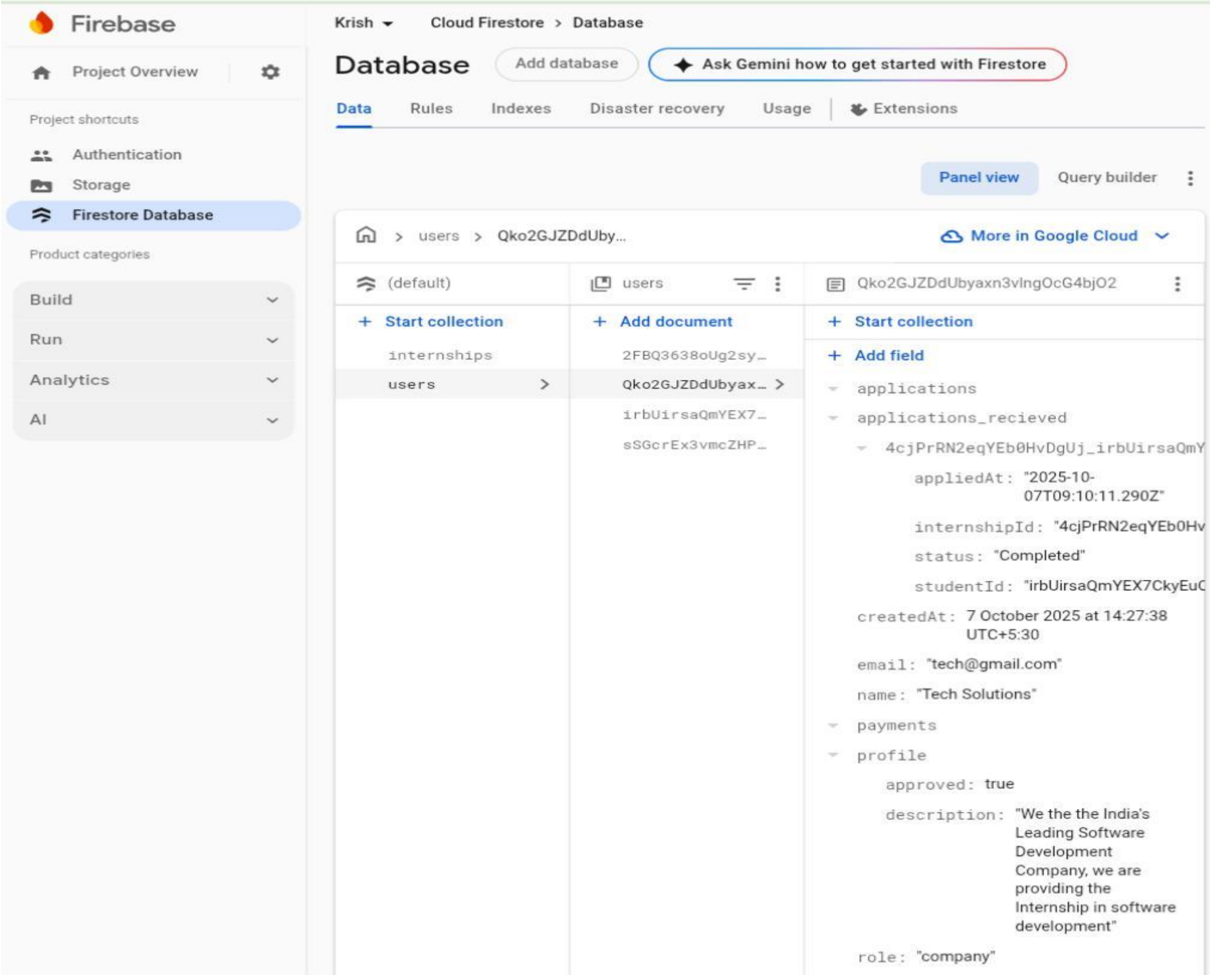
Q: Who can apply for Internship ?
A: Anybody who have to explore various fields.

Contact Messages

From: Ram Kapoor (ramkapoor@gmail.com)
Hello Sir
Sent on: 7/10/2025, 2:37:32 pm

© 2024 InternQuest. All rights reserved.

## Backend Database of Website:



(default)
- internships
- users

users
- 2FBQ3638oUg2sy…
- Qko2GJZDdUbyax…
- irbUirsaQmYEX7…
- sSGcrEx3vmcZHP…

Qko2GJZDdUbyaxn3vlngOcG4bjO2

```
applications
applications_recieved
  4cjPrRN2eqYEb0HvDgUj_irbUirsaQmY
    appliedAt: "2025-10-
               07T09:10:11.290Z"
    internshipId: "4cjPrRN2eqYEb0Hv
    status: "Completed"
    studentId: "irbUirsaQmYEX7CkyEuC
createdAt: 7 October 2025 at 14:27:38
           UTC+5:30
email: "tech@gmail.com"
name: "Tech Solutions"
payments
profile
  approved: true
  description: "We the the India's
               Leading Software
               Development
               Company, we are
               providing the
               Internship in software
               development"
role: "company"
```

# Chapter 6: Conclusion and Future Work

**Conclusion:**

The InternQuest project successfully created a functional and streamlined web platform to bridge the gap between education and the professional world for 12th-pass students. The system achieves its core objective by transforming the post-exam holiday period into a productive time for skill acquisition and career exploration. The platform's success lies in its user-friendly dashboards for students, companies, and administrators, which simplify the entire internship lifecycle from application to certification. The use of Firebase as a serverless backend proved to be an effective choice, enabling real-time data synchronization and ensuring scalability and reliability.

The project successfully implemented key functionalities such as user authentication, profile management, and a pre-screening aptitude test to filter for committed candidates. The seamless workflow for internship applications, managed by the company dashboard, along with the automated generation of valuable documents like offer letters and certificates, provides a tangible and valuable outcome for students.

---

**Future Work:**

While the current version of InternQuest is fully functional, there are several areas for future development to enhance the platform's capabilities and user experience.

- **Integrated Payment Gateway**: The current system uses a simulated payment process for the nominal fee of ▪149. Future development will focus on integrating a real payment gateway (e.g., Stripe, Razorpay) to handle secure financial transactions.
- **Communication Features**: Implementing a secure, in-platform messaging system would allow direct communication between students and companies, streamlining the interview and onboarding process.
- **Enhanced Analytics**: The admin dashboard could be expanded with more advanced analytics and reporting tools to provide deeper insights into platform usage, such as internship completion rates, student demographics, and company engagement.
- **Skill Assessment Modules**: To move beyond a static aptitude test, the platform could offer specialized skill-based assessments (e.g., coding challenges for IT internships, portfolio submissions for design roles) to provide more accurate matching.
- **Mobile Application**: Developing a native mobile application for both iOS and Android platforms would improve accessibility and provide a more convenient user experience.

---

# Chapter 7: References

The development of the InternQuest project was made possible by referencing a combination of foundational web development resources and specialized libraries. These sources were essential for defining the project's structure, implementing its core functionalities, and ensuring its technical integrity.

- **Firebase Documentation**: This served as the primary resource for all backend-as-a-service functionalities. The documentation provided in-depth guides for implementing **Firebase Authentication** for user management and **Firebase Firestore** for database operations, including real-time data synchronization.

- **jsPDF Library**: The official documentation for the **jsPDF library** was critical for implementing the client-side PDF generation. This resource guided the process of creating dynamic documents such as offer letters, payment receipts, and certificates of completion directly within the user's browser.

- **Web Development Resources**: Standard web development tutorials and documentation from sources like **W3Schools** and **MDN Web Docs** were consulted to ensure best practices in **HTML5, CSS3, and JavaScript (ES6+)**.

- **Project Files**: The project's own files, including script.js, index.html, and style.css, served as a direct reference for the implementation details and functional requirements of the system. These files provided the foundational code for all user dashboards, the aptitude test, and the document generation logic.