



CAPTA

Software Architecture and Design Implementation

Contact : Mahesh
info@leantechlabs.in

Table of Contents :

Overview	3
User Case	4 - 11
Work Flow Business Logic	12 - 13
User Interface design	14 - 16
Database Schema	17 - 24
User Roles & Permissions	26
Agile Sprints	27 - 28
ER diagram	29
DFD diagram & Mind Map	29
Project Development Guidelines	29 - 31

Overview

In a strategic move to enhance its technological infrastructure and streamline its operations, Cognite, a prominent platform, has entered into a collaboration with Leantech Labs, a leading software development company. The primary objective of this partnership is to jointly develop a cutting-edge software solution named CATPA

User Case

Use Case: Manage CRT Training Process

Actors:

1. **Administrator (Admin)**
2. **Moderator**
3. **Trainer**
4. **College Representative (TPO/Placement Officer)**

Use Case 1: Initiate CRT Training Process

Primary Actor: Admin or Moderator

Preconditions:

- The Capta system is operational and accessible.
- Relevant user accounts (Admin, Moderator) have been set up.

Main Flow:

1. **Login to Capta System:**
 - The Admin or Moderator accesses the Capta system through a secure login.
2. **Add New Moderator (Optional):**
 - If necessary, the Admin adds a new Moderator account to the system.
 - Admin provides essential details such as username, password, and permissions for the new Moderator.
3. **Add College Details:**
 - The Admin or Moderator navigates to the "College Management" section.
 - Adds details of educational institutions (colleges, schools, or universities) to the system.
 - Inputs information such as institution name, type, location, contact details, and any other relevant data.
4. **Create Training Modules:**
 - Admin or Moderator moves to the "Training Module Management" section.
 - Initiates the creation of new training modules.
 - Specifies details such as module name, duration, content, and any other necessary information.
5. **Attach Relevant Curriculum:**

- While creating training modules, Admin or Moderator attaches relevant curriculum or training materials.
- This could include documents, presentations, videos, or any resources essential for the training modules.

Postconditions:

- College details, including institution name, type, and contact information, are successfully stored in the system.
- Training modules, with associated details such as name, duration, and content, are created and saved.
- Curriculum or training materials are attached and linked to the respective training modules.
- If a new Moderator is added, their account is created with the necessary permissions.

Use Case 2: Agreement and MOU Generation:

Primary Actor: College Representative

Preconditions:

- College details are successfully available in the Capta system.

Main Flow:

1. Login to Capta System:

- The College Representative accesses the Capta system by logging in with appropriate credentials.

2. Agree to or Initiate Business Deal for CRT Training:

- The representative either agrees to an existing business deal (if available) or initiates a new business deal manually.
- In the case of manual initiation, the representative provides essential details of the business deal, including the scope of training, number of modules, and any specific requirements.

3. Admin or Moderator Generates MOU:

- Upon the college's agreement or initiation of a business deal, the Admin or Moderator is notified or manually prompted to generate an MOU.
- The Admin or Moderator accesses the "MOU Generation" section.
- System prompts the Admin or Moderator to input specific details relevant to the MOU.
- The Admin or Moderator generates the MOU document based on the provided information.

Postconditions:

- Details of the business deal or agreement are stored in the system.
- An MOU document is successfully generated, reflecting the terms and conditions of the training partnership.

- The generated MOU is associated with the respective college in the system.

Use Case 3: Verify and MOU Confirmation

Preconditions:

- MOU details have been generated.

Main Flow:

1. **Verify MOU Details:**
 - The Admin or Moderator logs into the Capta system.
 - Navigates to the "MOU Management" section.
 - Accesses the list of generated MOUs.
 - Selects the specific MOU for verification.
2. **Review MOU Details:**
 - Admin or Moderator reviews the details of the MOU.
 - Ensures that all terms and conditions are accurate and align with the agreed-upon business deal.
3. **System Validation:**
 - The system validates the MOU details to ensure completeness and accuracy.
 - If there are any discrepancies or missing information, the system prompts the Admin or Moderator to address those issues.
4. **Confirm MOU:**
 - Admin or Moderator, satisfied with the MOU details, proceeds to confirm the MOU.
 - A confirmation prompt is displayed, asking for final verification before confirmation.
5. **Update MOU Status:**
 - Upon confirmation, the system updates the status of the MOU to reflect that it has been verified and confirmed.
 - The MOU is now considered official and ready for the training preparation phase.

Postconditions:

- The status of the MOU is updated to indicate that it has been verified and confirmed.
- The system stores the updated status in the database.

Use Case 4: Module Creation and Verification

Preconditions:

- College details and MOU details have been confirmed in the system.

Main Flow:

1. Create Training Modules:

- The Moderator logs into the Capta system.
- Navigates to the "Training Module Management" section.
- Initiates the creation of new training modules for the specified college and associated MOU.
- Provides details such as module name, duration, content, and any other relevant information.

2. Attach Relevant Curriculum:

- While creating training modules, Moderator attaches relevant curriculum or training materials.
- This could include documents, presentations, videos, or any resources essential for the training modules.

3. Fill Module Confirmation Sheet:

- After creating training modules, the Moderator is prompted to fill out a module confirmation sheet.
- Specifies details such as the number of sessions, session content, trainers involved, and any other relevant information.

4. System Validation:

- The system validates the information entered by the Moderator.
- Ensures that the details in the module confirmation sheet align with the created training modules.

5. Submit for Admin Verification:

- The Moderator submits the module confirmation sheet for verification by the Admin.
- The system generates a notification or alert for the Admin to review the submitted sheet.

6. Admin Verifies the Module Confirmation Sheet:

- Admin logs into the Capta system.
- Accesses the notifications or the "Verification" section.
- Reviews the details in the module confirmation sheet.
- Validates that the information matches the created training modules and is consistent with the agreed-upon terms in the MOU.

7. Update Confirmation Status:

- Based on the verification, the Admin updates the status of the module confirmation sheet.
- If verified successfully, the status is marked as verified; otherwise, it may be marked as pending or rejected with appropriate feedback.

Postconditions:

- Training modules are successfully created and stored in the system.

- The module confirmation sheet is verified and its status is updated accordingly.

Use Case 5: Batch Creation and Trainer Allocation

Primary Actor: Moderator

Preconditions:

- Training modules have been confirmed by the administrator.
- Necessary training curriculum is attached to the modules.
- The college details, including the list of students, are available in the system.

Main Flow:

1. Moderator Creates Batches:

- The Moderator logs into the Capta system.
- Navigates to the "Batch Management" section.
- Initiates the creation of new batches for the specified college.
- Provides batch details such as batch name, start date, end date, and any additional relevant information.
- The system prompts Moderator to add students to the newly created batches.

2. Moderator Adds Students to Batches:

- Moderator selects a batch to add students.
- Adds students to the batch by selecting them from the available student list for the college.
- Specifies the number of students per batch and any specific requirements.

3. Moderator Allocates Trainers:

- After creating batches and adding students, Moderator navigates to the "Trainer Allocation" section.
- Initiates the allocation of trainers to the created batches.
- Selects the relevant training module associated with the batches.
- Specifies the number of sessions per week or any specific scheduling requirements.

4. System Validates and Confirms:

- The system validates the information entered by the Moderator, ensuring that batch creation and trainer allocation meet system constraints.
- If any issues are detected, the system provides appropriate feedback or prompts for correction.

5. Batches, Students, and Trainers are Allocated:

- Upon successful validation, the system creates the specified batches, assigns students to those batches, and allocates trainers based on the provided information.
- The information is stored in the system database for future reference.

Postconditions:

- Batches are successfully created for the college, including details such as batch name, start date, and end date.
- Students are assigned to their respective batches.
- Trainers are allocated to the specified batches or sessions.
- The system database is updated with the newly created batches, student assignments, and trainer allocations.

Use Case 6: Training Session and Reporting

Primary Actor: Trainer

Preconditions:

- Batches, trainers, and modules have been successfully allocated by the Moderator.
- Relevant training materials and curriculum are available.
- The system is operational.

Main Flow:

1. Trainer Delivers Sessions:

- The Trainer logs into the Capta system.
- Accesses the schedule to view the allocated batches and training modules.
- Conducts training sessions as per the schedule, covering the specified content and duration.

2. Updates Session Reports:

- After each training session, the Trainer updates the session reports.
- Specifies details such as session date, time, content covered, and any observations or notes.
- The system may prompt the Trainer to upload relevant materials, such as presentation slides or additional resources used during the session.

3. Takes Attendance:

- During the training session, the Trainer takes attendance of the participating students.
- Marks the presence of each student in the system, ensuring accurate tracking of attendance.

4. Generate Daily Session Reports:

- After completing the day's training sessions, the Trainer navigates to the "Reporting" section.
- Initiates the generation of daily session reports for the conducted sessions.
- The system prompts the Trainer to specify the date or date range for which the reports are needed.

5. System Generates Reports:

- The system retrieves session data for the specified date or date range.
- Generates detailed reports that include information on session timings, content covered, attendance records, and any additional notes provided by the Trainer.

Postconditions:

- Session details, including content covered and attendance records, are successfully updated in the system.
- Daily session reports are generated and stored in the system, providing insights into training activities.

Use Case 7: Generate Daily Session / Trainers Allocated / Next Day Training

7.1 Generate Daily Session Reports

Primary Actor: Admin

Preconditions:

- Training sessions have been conducted, and data is available in the system.
- Trainers have marked attendance and updated session details.
-

Main Flow:

1. The Admin logs into the Capta system.
2. Admin navigates to the reporting section.
3. Admin selects the option to generate daily session reports.
4. System prompts Admin to input the date for which the report is needed.
5. Admin enters the date for which the report is to be generated.
6. The system retrieves session data for the specified date.
7. Admin reviews and verifies the generated daily session reports.
8. Admin saves or exports the reports for record-keeping or distribution.

Postconditions:

- Daily session reports for the specified date are generated and accessible to the Admin.

7.2 Trainers Allocated Session Reports

Preconditions:

- Trainers have been allocated to specific modules and batches.
- Relevant data is available in the system.

Main Flow:

1. The Admin logs into the Capta system.

2. Admin navigates to the reporting section.
3. Admin selects the option to generate trainers allocated reports.
4. System prompts Admin to input the date or date range for which the report is needed.
5. Admin enters the date or date range for which the report is to be generated.
6. The system retrieves data on trainers' allocations for the specified date or range.
7. Admin reviews and verifies the generated trainers allocated reports.
8. Admin saves or exports the reports for record-keeping or distribution.

Postconditions:

- Trainers allocated reports for the specified date or date range are generated and accessible to the Admin.

7.3 Generate Reports on Next Day Training

Preconditions:

- Training schedules for the next day are planned and available in the system.

Main Flow:

1. The Admin logs into the Capta system.
2. Admin navigates to the reporting section.
3. Admin selects the option to generate reports on next day training.
4. System retrieves data on the scheduled training sessions for the next day.
5. Admin reviews and verifies the generated reports on next day training.
6. Admin saves or exports the reports for distribution or planning purposes.

Postconditions:

- Reports on the scheduled training sessions for the next day are generated and accessible to the Admin.

Alternate Flow:

- If at any point the admin, moderator, or trainer encounters an issue or needs assistance, the system should provide appropriate error messages or support features.

Work Flow and Business Logic

1. User Management:

- **Admin:** Has the authority to manage user accounts, including creating, updating, and deleting them.
- **Moderator:** Cannot manage user accounts.

2. Moderator Management:

- **Admin:** Can manage (create, update, delete) moderator accounts.
- **Moderator:** Cannot manage other moderator accounts.

3. Trainer Management:

- **Admin:** Can manage (create, update, delete) trainer accounts.
- **Moderator:** Can manage trainer accounts.

4. College Management:

- **Admin:** Has the authority to manage colleges, including adding new colleges and updating existing information.
- **Moderator:** Cannot manage colleges.

5. MOU Management (Memorandum of Understanding):

- **Admin:** Can initiate, generate, and confirm MOUs with colleges.
- **Moderator:** Can manage MOUs.

6. Training Module Management:

- **Admin:** Can create, update, and delete training modules.
- **Moderator:** Can manage training modules.

7. Curriculum Management:

- **Admin:** Can manage the curriculum associated with training modules.
- **Moderator:** Can manage curriculum.

8. Batch Management:

- **Admin:** Can manage batches, including creating, updating, and deleting them.
- **Moderator:** Can manage batches.

9. Session Management:

- **Admin:** Can manage training sessions, including scheduling and updating.
- **Moderator:** Cannot manage training sessions.
- **Trainer:** Can manage training sessions they are assigned to.

10. Report Management:

- **Admin:** Can manage reports, including viewing and analyzing them.
- **Moderator:** Can manage reports.

- **Trainer:** Can submit session reports.

11. Student Management:

- **Admin:** Can manage student details.
- **Moderator:** Can manage student details.
- **Trainer:** Cannot manage student details.

12. Trainer Allocation Management:

- **Admin:** Can allocate trainers to batches.
- **Moderator:** Can allocate trainers to batches.
- **Trainer:** Cannot allocate trainers.

13. Mark Attendance:

- **Admin:** Can mark attendance for training sessions.
- **Moderator:** Cannot mark attendance.
- **Trainer:** Can mark attendance for sessions they are assigned to.

14. Submit Session Reports:

- **Admin:** Can submit session reports.
- **Moderator:** Cannot submit session reports.
- **Trainer:** Can submit session reports.

User Interface Design

Navigation Bar

1. **Home:**
 - Overview of the system or a dashboard providing key metrics and notifications.
2. **User Management:**
 - **Add Trainer / Moderator:**
 - Allows admins to add new trainers or moderators to the system.
 - **Manage Trainer / Moderator:**
 - Provides functionality to view, edit, and delete trainer or moderator accounts.
3. **College Management:**
 - **Add College:**
 - Enables admins to add new colleges to the system.
 - **Manage College:**
 - Allows admins to view, edit, and delete college details.
 - **Add TPO/ College Point of Contact:**
 - Allows admins to add new Placement Officers or College Point of Contact.
 - **Manage TPO/ College Point of Contact:**
 - Provides functionality to view, edit, and delete TPO details.
4. **MOU Management:**
 - **Create MOU:**
 - Facilitates the initiation and creation of Memorandum of Understanding.
 - **MOU Confirmation:**
 - Confirmation process for the MOU, possibly involving admin or moderator approval.
 - **Manage MOU:**
 - Allows admins to view, edit, and manage MOU details.
5. **Training Curriculum / Module:**
 - **Create Curriculum:**
 - Allows admins to define the curriculum associated with training modules.
 - **Manage Curriculum:**
 - Provides functionality to view, edit, and delete curriculum details.
 - **Create Module:**
 - Enables admins to create new training modules.
 - **Manage Curriculum:**
 - Allows admins to view, edit, and delete module details.

6. **Module Confirmation:**

- **Module Confirmation Sheet:**
 - Facilitates the creation of confirmation sheets associated with training modules.
- **Manage Confirmation Sheet:**
 - Allows admins to view, edit, and manage module confirmation details.
- **Module Status:**
 - Provides information on the status of training modules.

7. **Batch Management:**

- **Create Batch:**
 - Enables admins to create new batches for training.
- **Manage Batch:**
 - Allows admins to view, edit, and delete batch details.
- **Allocate Batch:**
 - Facilitates the allocation of batches to trainers or sessions.

8. **Session Management:**

- **Take Attendance:**
 - Allows trainers to mark attendance for students during sessions.
- **Modify Attendance:**
 - Provides functionality to modify previously recorded attendance.
- **Upload Session Report:**
 - Enables trainers to upload reports after each training session.
- **Session Details / Schedule:**
 - Provides an overview of scheduled training sessions.

9. **Report Management:**

- **Curriculum Report:**
 - Generates reports related to the curriculum.
- **Colleges Report:**
 - Provides reports on colleges and their details.
- **Module Report:**
 - Generates reports on training modules.
- **Trainer Report:**
 - Provides reports on trainers and their performance.
- **Session Report:**
 - Generates reports on individual training sessions.
- **Attendance Report:**
 - Provides reports on student attendance.
- **Trainers Attendance Report:**
 - Generates reports on trainers' attendance.
- **Financial Report:**

- Provides financial reports related to training activities.

10. **Settings:**

- **System Settings:**
 - Allows admins to configure system-wide settings.
- **API Settings:**
 - Provides functionality to configure API-related settings.

Database Schema

User Table

1. Field	1. Type	1. Description
1. UserID	1. INT	1. Primary Key, Auto-incremented
1. Username	1. VARCHAR	1. Unique username for login
1. Password	1. VARCHAR	1. Hashed password for security
1. UserRole	1. ENUM	1. Admin, Moderator, Trainer
LastLogin	1. TIME	Last Admin Login Time
IP	1. VARCHAR	Login IP address

Moderator Table:

1. Field	1. Type	1. Description
1. ModeratorID	1. INT	1. Primary Key, Auto-incremented
1. UserID	1. INT	1. Foreign Key referencing User table
1. BankAccounts	1. ARRAY	1. Array of bank account details
1. Photograph	1. JSON	Link From Storage Bucket
1. AadharDetails	1. JSON	Link From Storage Bucket
1. PanCardDetails	1. JSON	Link From Storage Bucket
1. ...	1. ...	1. Other moderator-specific details

Trainer Table:

1. Field	1. Type	1. Description
1. TrainerID	1. INT	1. Primary Key, Auto-incremented
1. UserID	1. INT	1. Foreign Key referencing User table
1. BankAccounts	1. ARRAY	1. Array of bank account details

1. Photograph	1. JSON	Link From Storage Bucket
1. Resume	1. JSON	Link From Storage Bucket
1. Skills	1. ARRAY	1. Array of skills possessed by trainer
1. AadharDetails	1. JSON	Link From Storage Bucket
1. PanCardDetails	1. JSON	Link From Storage Bucket
1. Experience	1. ARRAY	1. Array of experience details

College Table:

Field	Type	Description
CollegeID	INT	Primary Key, Auto-incremented
Name	VARCHAR	Name of the college
Type	VARCHAR	Type of institution (e.g., College, School, University)
Location	VARCHAR	Location of the institution
Contact	VARCHAR	Contact information
TPOName	VARCHAR	Name of the TPO
TPOEmails	ARRAY	Array of TPO's email addresses
TPOPhoneNumbers	ARRAY	Array of TPO's phone numbers
DecisionMakerName	VARCHAR	Name of the decision-maker
DecisionMakerEmails	ARRAY	Array of decision-maker's email addresses
DecisionMakerPhoneNumbers	ARRAY	Array of decision-maker's phone numbers

In this schema:

- **ARRAY:** Represents an array data type for storing multiple values of the same type.

You can add more fields or modify data types based on your specific requirements. These additional fields provide a place to store contact details for both the TPO and the decision-maker associated with the college. Adjustments can be made as needed to capture all the relevant information for your use case.

MOU Table:

Field	Type	Description
MOUID	INT	Primary Key, Auto-incremented
Date	DATE	Date of MOU creation
Location	VARCHAR	Location where MOU is executed
FirstParty	JSON	JSON format for details of the first party
SecondParty	JSON	JSON format for details of the second party
TermsConditions	JSON	JSON format for terms and conditions of the MOU
PurposeScope	JSON	JSON format for purpose/scope of the collaboration
PaymentTerms	JSON	JSON format for payment terms
Termination	JSON	JSON format for termination details

In this schema:

- **MOUID:** Represents a unique identifier for each MOU.
- **FirstParty** and **SecondParty:** These fields store JSON-formatted details of the first and second parties.
- **TermsConditions, PurposeScope, PaymentTerms, and Termination:** These fields store JSON-formatted details for the respective sections of the MOU.

Here's a basic example of how the JSON format might look for a few fields:

<https://drive.google.com/file/d/1J2xRV7QUI7KvU3Yu2aXs6il775ONyK7t/view?usp=sharing>

Original Mou Document :

<https://drive.google.com/file/d/1J2xRV7QUI7KvU3Yu2aXs6il775ONyK7t/view?usp=sharing>

MOUConfirmation Table:

Field	Type	Description
ConfirmationID	INT	Primary Key, Auto-incremented
MOUID	INT	Foreign Key referencing MOU table
AdminID	INT	Foreign Key referencing User table for Admin
ModeratorID	INT	Foreign Key referencing User table for Moderator
ConfirmationDate	DATE	Date of confirmation
ConfirmationStatus	ENUM	Status of MOU confirmation (e.g., Pending, Confirmed, Rejected)
Comments	TEXT	Comments or notes related to the confirmation

In this schema:

- **ConfirmationID:** Represents a unique identifier for each confirmation entry.
- **MOUID:** Foreign Key referencing the MOU table, linking each confirmation to a specific MOU.
- **AdminID** and **ModeratorID:** Foreign Keys referencing the User table, indicating the admin and moderator involved in the confirmation process.
- **ConfirmationDate:** Represents the date when the confirmation is made.
- **ConfirmationStatus:** Represents the status of MOU confirmation, which can be enumerated as Pending, Confirmed, Rejected, etc.
- **Comments:** Provides space for any comments or notes related to the confirmation.

ModuleConfirmationSheet Table:

Field	Type	Description
ModuleID	INT	Primary Key, Auto-incremented
MOU ID	INT	Foreign Key referencing MOUConfirm table
CollegeName	VARCHAR(255)	Name of the college
POCName	VARCHAR(255)	Point of Contact (POC) Name
Designation	VARCHAR(255)	Designation of the POC
POCEmail	VARCHAR(255)	Email of the POC
POCContact	VARCHAR(20)	Contact number of the POC
Address	VARCHAR(255)	Address of the college
SuitableTransport	VARCHAR(255)	Details about suitable transport
Food	VARCHAR(255)	Details about food arrangements
Accommodation	VARCHAR(255)	Details about accommodation
LocalTransport	VARCHAR(255)	Details about local transport
MajorTransport	VARCHAR(255)	Details about major transport
PreviousVendor	VARCHAR(255)	Details about previous vendor

Feedback	TEXT	Feedback about the training
InterestGoals	TEXT	Goals and interests related to the training
Day1Company	VARCHAR(255)	Company involved on the first day
OtherCompanies	TEXT	Details about other involved companies
ModuleName	VARCHAR(255)	Name of the training module
HoursPerBatch	INT	Hours allocated per batch for training
ModulesCovered	VARCHAR(255)	Details about covered modules in training
ExecutionType	VARCHAR(10)	Type of execution (e.g., in-person, online)
StartDate	DATE	Start date of the training module
EndDate	DATE	End date of the training module
NumStudents	INT	Number of students participating
NumBatches	INT	Number of batches for the training
StartPreferredTimings	TIME	Preferred start timings for the training
EndPreferredTimings	TIME	Preferred end timings for the training
MarketingPerson	VARCHAR(255)	Name of the marketing person
MarketingContact	VARCHAR(20)	Contact number of the marketing person
MarketingEmail	VARCHAR(255)	Email of the marketing person
TrainingManager	VARCHAR(255)	Name of the training manager
TrainingContact	VARCHAR(20)	Contact number of the training manager
TrainingEmail	VARCHAR(255)	Email of the training manager
UnitBasis	VARCHAR(50)	Basis for calculating units
UnitCost	DECIMAL(10, 2)	Cost per unit
NumUnits	INT	Number of units
TotalCost	DECIMAL(10, 2)	Total cost
GST	DECIMAL(5, 2)	Goods and Services Tax
GrossIncome	DECIMAL(10, 2)	Gross income from the training
TDS	DECIMAL(5, 2)	Tax Deducted at Source
AmountToCoignAccount	DECIMAL(10, 2)	Amount transferred to Coign's account
INSTACKExams	VARCHAR(3)	Indicator for INSTACK exams
INSTACKMonths	INT	Number of months for INSTACK exams
Income	DECIMAL(10, 2)	Total income from the training
Expenses	DECIMAL(10, 2)	Total expenses for the training
TotalDaysTraining	INT	Total days of training
PerDayPerTrainer	DECIMAL(10, 2)	Cost per day per trainer
TotalContractValueBatch	DECIMAL(10, 2)	Total contract value for all batches
TotalTrainingHours	INT	Total hours of training

CompanySpecificHours	INT	Company-specific hours of training
TotalHours	INT	Total hours of the training module
PerHourPerBatch	DECIMAL(10, 2)	Cost per hour per batch
PerDayPerBatch	DECIMAL(10, 2)	Cost per day per batch
TotalStudents	INT	Total number of students
TotalBatches	INT	Total number of
TotalTrainingDays	INT	Total training days
TotalContractValueCOIGN	DECIMAL(10, 2)	Total contract value for COIGN Consultants
NumberOfTrainers	INT	Number of trainers involved
PerHeadPerDay	DECIMAL(10, 2)	Cost per head per day
NumberOfDaysPerTrainer	INT	Number of days per trainer
PortalCostPerStudent	DECIMAL(10, 2)	Cost of portal per student
NumberOfStudents	INT	Number of students involved
Travelling	DECIMAL(10, 2)	Cost of traveling
Commission	DECIMAL(10, 2)	Commission for the training
IndirectExpenses	DECIMAL(10, 2)	Indirect expenses for the training
TotalExpenses	DECIMAL(10, 2)	Total expenses for the training
TotalProfit	DECIMAL(10, 2)	Total profit calculated

Program Curriculum Table: (Curriculum Table)

Field	Type	Description
S.No	INT	Serial number (unique identifier)
CuriucllumID	INT	Curriculum identifier (Primary Key)
ModuleName	VARCHAR(255)	Name of the module
Topic	VARCHAR(255)	Main topic of the module
SubTopic	VARCHAR(255)	Sub-topic or specific area of the module
Day	INT	Day of the module (e.g., Day 1, Day 2)
Hours	INT	Number of hours for the module
PractisePrograms	VARCHAR(255)	Practical programs covered in the module

Module Curriculum Table:

Field	Type	Description
ModuleName	VARCHAR(255)	Name of the module
ModuleId	INT	Module ID identifier (Primary Key)
TotalHours	INT	Total number of hours for the module
TotalDays	INT	Total number of days for the module
TotalBatches	INT	Total number of batches for the module
StartDate	DATE	Start date of the module
EndDate	DATE	End date of the module
Curriculum	VARCHAR(255)	Selected curriculum for the module (Foreign Key ProgramCurriculum ModuleID)

Batch Creation Table:

Field	Type	Description
BatchID	INT	Batch identifier (unique)
ModuleID	INT	Identifier linking to the associated ModuleCurriculum
BatchName	VARCHAR(255)	Name or identifier for the batch
Students	INT	Number of students in the batch
TrainerID	INT	Identifier linking to the assigned trainer
Days	INT	Number of days for the batch
StartDate	DATE	Start date of the batch
EndDate	DATE	End date of the batch
Schedule	VARCHAR(255)	Details of the schedule for the batch
Status	VARCHAR(50)	Current status of the batch (e.g., Active, Completed)

Student Table :

Field	Type	Description
StudentID	INT	Student identifier (unique)
BatchID	INT	Identifier linking to the associated batch
StudentName	VARCHAR(255)	Full name of the student
Class	VARCHAR(20)	Class of the student (e.g., 10th, 11th, etc.)
Section	VARCHAR(20)	Section of the class (e.g., A, B, C)
RegistrationNumber	VARCHAR(50)	Registration number of the student
Email	VARCHAR(255)	Email address of the student

TrainerAllocation Table:

Field	Type	Description
AllocationID	INT	Allocation identifier (unique)
BatchID	INT	Identifier linking to the associated batch
TrainerID	INT	Identifier linking to the assigned trainer
StartDate	DATE	Start date of the trainer's work in the batch
EndDate	DATE	End date of the trainer's work in the batch
Commercial	DECIMAL(10, 2)	Commercial details for the trainer in the batch
Status	VARCHAR(50)	Current status of the allocation (e.g., Active, Completed)

Session Attendance

Field	Type	Description
AttendanceID	INT	Attendance identifier (unique)
BatchID	INT	Identifier linking to the associated batch
SessionDate	DATE	Date of the training session
TrainerID	INT	Identifier linking to the trainer taking attendance

StudentStatus	JSON	{Batch:{Students:{student_id:present}}}} fetch students details from batch id , create json.
---------------	------	--

Session Report

Field	Type	Description
ReportID	INT	Report identifier (unique)
BatchID	INT	Identifier linking to the associated batch
SessionDate	DATE	Date of the training session
TrainerID	INT	Identifier linking to the trainer conducting the session
TopicsCovered	TEXT	Topics covered during the session
MaterialsUsed	TEXT	Materials used during the session
OverallFeedback	TEXT	Overall feedback on the session
Suggestions	TEXT	Suggestions for improvement

User Roles & Permissions

Permission	Admin	Moderator	Trainer
User Management	Yes	Yes	No
Moderator Management	Yes	No	No
Trainer Management	Yes	Yes	No
College Management	Yes	No	No
MOU Management	Yes	Yes	No
Training Module Management	Yes	Yes	No
Curriculum Management	Yes	Yes	No
Batch Management	Yes	Yes	No
Session Management	Yes	Yes	Yes
Report Management	Yes	Yes	Yes
Student Management	Yes	Yes	No
Trainer Allocation Management	Yes	Yes	No
Mark Attendance	Yes	No	Yes
Submit Session Reports	Yes	No	Yes

Agile Sprints

Sprint 1: User Management

1. **Task:** UI for adding trainers/moderators.
2. **Task:** Backend logic for storing user details.
3. **Task:** UI for managing trainer/moderator accounts.
4. **Task:** Backend logic for retrieving/updating user details.

Sprint 2: College Management

5. **Task:** UI for adding new colleges.
6. **Task:** Backend logic for storing college details.
7. **Task:** UI for managing college details.
8. **Task:** Backend logic for retrieving/updating college information.

Sprint 3: MOU Management

9. **Task:** UI for creating MOU.
10. **Task:** Backend logic for storing MOU details.
11. **Task:** Workflow for MOU confirmation.
12. **Task:** Backend logic for updating MOU status.

Sprint 4: Training Curriculum / Module

13. **Task:** UI for creating curriculum.
14. **Task:** Backend logic for storing curriculum details.
15. **Task:** UI for managing curriculum.
16. **Task:** Backend logic for retrieving/updating curriculum.

Sprint 5: Module Confirmation

17. **Task:** UI for creating module confirmation sheets.
18. **Task:** Backend logic for storing confirmation sheet details.
19. **Task:** UI for managing module confirmation.
20. **Task:** Backend logic for retrieving/updating confirmation details.

Sprint 6: Batch Management

21. **Task:** UI for creating batches.
22. **Task:** Backend logic for storing batch details.
23. **Task:** UI for managing batches.
24. **Task:** Backend logic for retrieving/updating batch information.

Sprint 7: Session Management

- 25. **Task:** UI for taking attendance.
- 26. **Task:** Backend logic for recording attendance.
- 27. **Task:** UI for uploading session reports.
- 28. **Task:** Backend logic for storing session reports.

Sprint 8: Report Management

- 29. **Task:** Backend logic for generating curriculum reports.
- 30. **Task:** Backend logic for generating college reports.
- 31. **Task:** Backend logic for generating module reports.
- 32. **Task:** Backend logic for generating trainer reports.

Sprint 9: Settings

- 33. **Task:** UI for system settings.
- 34. **Task:** Backend logic for handling system settings.
- 35. **Task:** UI for API settings.
- 36. **Task:** Backend logic for handling API settings.

Sprint 10: Review and Optimization

- 37. **Task:** Review and optimize code.
- 38. **Task:** Address any issues or bugs.
- 39. **Task:** Refactor code for better performance.
- 40. **Task:** Final testing and documentation.

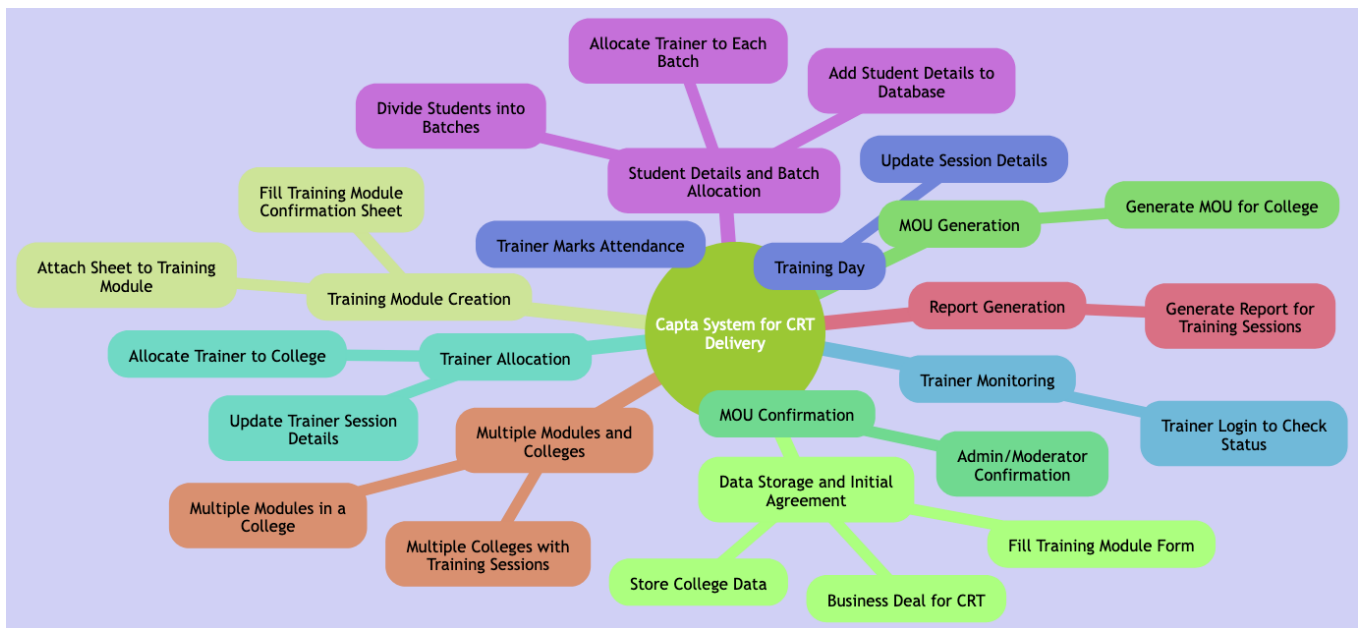
ER

<https://drive.google.com/file/d/1VveHIXqEJtnXOj9e36y67SP3VLflkkgG/view?usp=sharing>

DFD

<https://drive.google.com/file/d/19vM1lzQozlp0JCdPI2-B6TkxRFu3Hxrl/view?usp=sharing>

Mind Map



Project Development Guidelines

Introduction

Welcome to the Capta project! This documentation provides guidelines for writing code and documenting your work to ensure consistency and maintainability across the project.

Table of Contents

1. Coding Standards
2. Documentation
3. Development Workflow
4. Testing
5. Version Control
6. Deployment
7. Communication

1. Coding Standards

Follow these coding standards to maintain a consistent and readable codebase:

- JavaScript:
Adhere to the Airbnb JavaScript Style Guide.
Use meaningful variable and function names.
Keep functions small and focused.
- Node.js:
Follow best practices for asynchronous programming.
Utilize error-first callback pattern.
Make use of ES6 features when appropriate.

2. Documentation

Documentation is as important as code. Ensure that your code is well-documented:

- Code Comments:
Add comments for complex or non-trivial sections of code.
Clearly document the purpose of functions and methods.
- README:
Update the project README with instructions for installation, usage, and contribution.
Include a concise project overview.

3. Development Workflow

- Branching:

Create a new branch for each feature or bug fix.

Branch names should be descriptive (e.g., feature/user-authentication).

- Pull Requests:
Open pull requests early to encourage collaboration and discussion.
Reference the relevant issue in your pull request.

4. Testing

- Unit Tests:
Write unit tests for new features.
Ensure existing tests are maintained and updated.
- Integration Tests:
Test the integration of components and modules.

5. Version Control

- Commits:
Write clear and concise commit messages.
Follow the Conventional Commits standard.
- Git Flow:
Follow the Git Flow branching model.

6. Deployment

- Continuous Integration:
Set up CI/CD pipelines for automated testing and deployment.
Deploy to staging environments before production.
- Environment Configuration:
Use environment variables for configuration.
Store sensitive information securely.

7. Communication

- Team Meetings:
Participate in regular team meetings.
Discuss progress, challenges, and plans.
- Discord/Communication Channel:
Use Discord or another communication channel for quick updates and discussions.

Miscellaneous

Jira Dashboard :

<https://leantechlabs.atlassian.net/jira/projects>.

Git Hub:

<https://github.com/leantechlabs/capta>

Amazon Web Service :

https://console.aws.amazon.com/console/home?nc2=h_ct&src=header-signin

S3 Bucket :

Null

Google Firebase :

<https://firebase.google.com/>

Credentials :

Contact Mahesh / G Tarun