

# AURIONPRO PAYMENTS

PRESENTED BY –  
SAMIKSHA MANE,  
HARSH MALI

# Topics To Discuss :

1. The Process

2. Technology Used

3. Scope of Project

4. Database ER-Diagram

5. Key Features

6. Output and flow

7. Challenges & Learning

8. Work Distribution

9. Conclusion

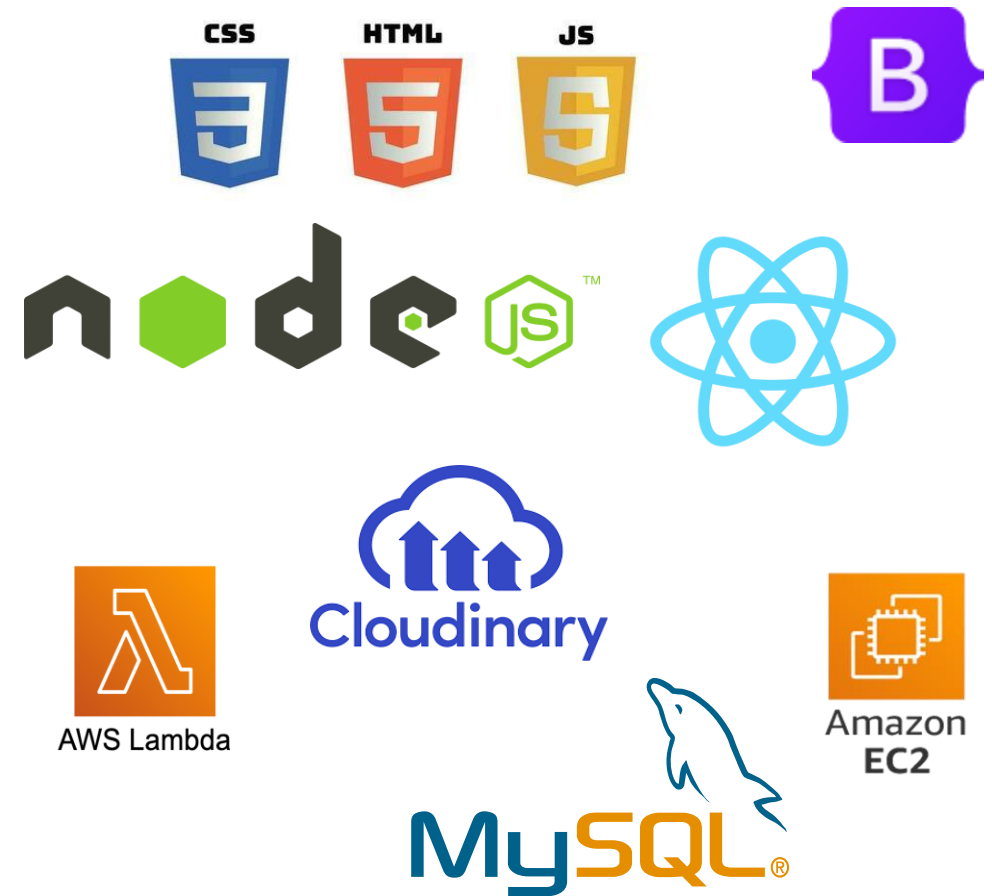
# The Process :

- Identifying Requirements
- Identifying Entities
- Creating ER Diagrams
- Creating Migrations
- Creating Basic CRUD in backend
- Designing Front-End Modules
- Creating required backend features on the go
- Testing project by adding dummy data
- Revising developed features based on testing results
- Deploying project



# Operating Environment

- Frontend  
React.js, HTML & CSS, Bootstrap
- Storage  
Cloudinary
- Backend  
Nodejs
- Database  
MYSQL
- Deployment  
Aws Ec2, Aws lambda



# Scope Of Project

## Admin User Features

The corporate banking application is designed to facilitate banking operations for corporate clients, and banks. The includes functionalities such as customer onboarding, client management, payment processing, salary disbursement, and report generation.

Customer Onboarding: Admin users can onboard new companies/customers by verifying the registered details of the company. Admins can reject or accept registration requests.

Manage Clients: Approve/Reject Payments. Bank users can approve or reject client payment requests, salary payment requests, add beneficiary requests from the clients.

Visualizations: Admin can see the current state of system through visualizations

# Scope Of Project

## Company User Features

Company Registration: Company users can register themselves by entering their details and required documents to verify.

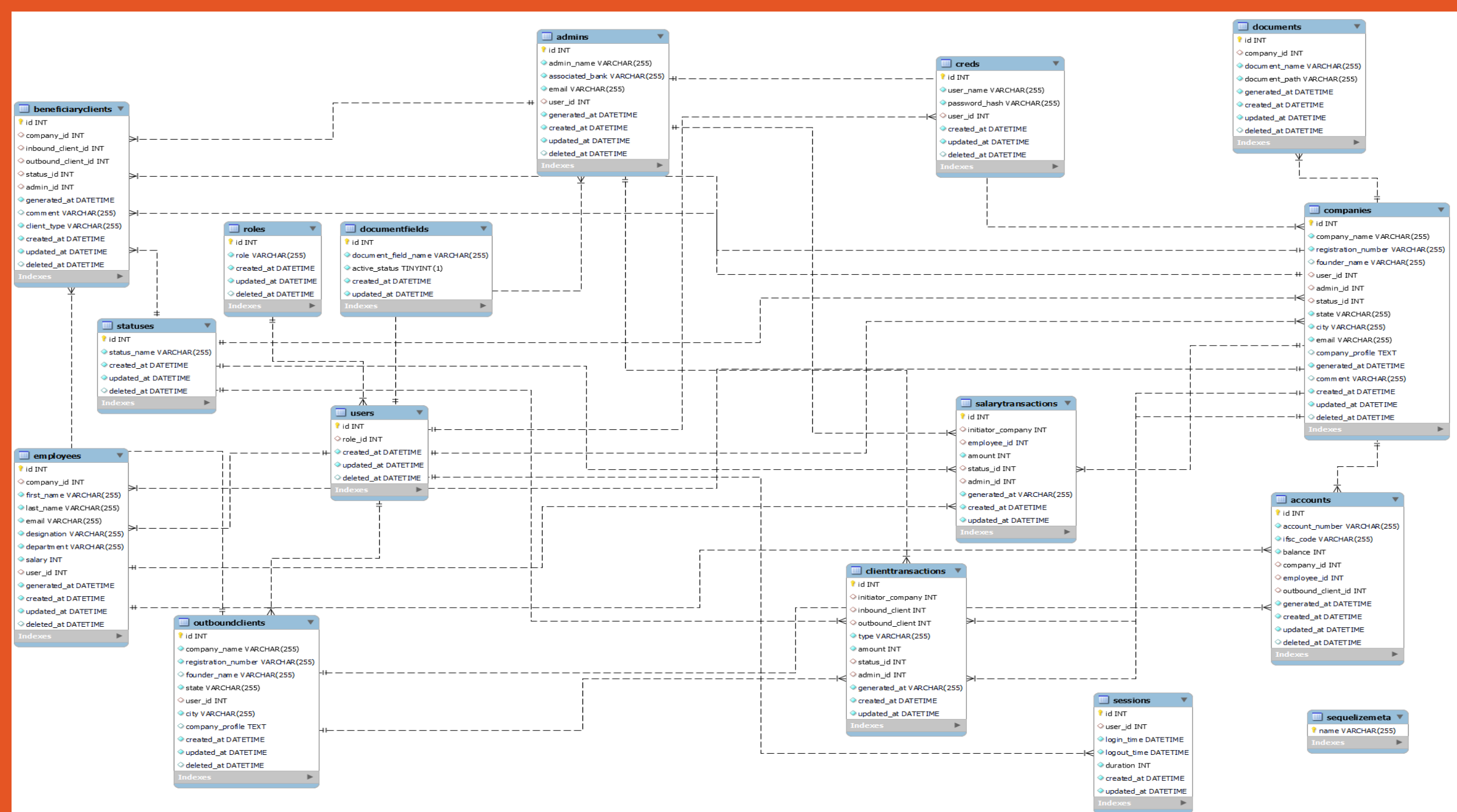
Manage Beneficiaries : Companies can add beneficiary clients which are within the system or outside the system. Company can also make payment to the added clients once verified by the admin

Manage Employees: Companies can add Employees along with their salary details to disburse employee salaries

Report: Companies can see history of transactions approved or rejected.

# Database ER-Diagram







# Key Features :

- **Authentication and Authorization:** Authentication of Admin and Company on each of the front end routes as well as backend apis. Role based authorization for all the features in the application. Package: jsonwebtoken
- **Encrypted Transportation:** Data sent to backend system is encrypted using RSA algorithm which is an asymmetric encryption algorithm. Package: encrypt-rsa
- **Visualizations:** Different visualization graphs to get an overview of the system statistics. Package: react-chartjs-2, chart.js
- **Batch processing:** Batch processing of all the transaction requests for fast approvals. Admin can approve transactions in batch or each at a time according to the needs.

# Key Features :

- **Session reports:** Admins can see the session report of every user. Minimum time, maximum time, average time the user was logged in to the system.
- **Filtrations:** Extensive filtering options for both Admin and Company users on the front end, allowing them to efficiently search and refine the data presented. Users can select from a wide range of filters that can be applied based on various data properties, such as company name, date range, state, etc.
- **Captcha Verification:** Captcha verification to restrict access to bots and computer programs which reduces the risk of crucial data being scraped. Package:react-google-recaptcha

# Key Features :

- **Approval based system:** A robust approval-based system ensures that all critical activities performed by companies, such as transactions and registrations are thoroughly verified and approved by the admin before being processed. By incorporating administrative oversight, the system maintains accuracy, enhances operational control, reduces the likelihood of human errors or mismanagement and minimizes the risk of glitches or failures, leading to a more reliable and well-governed platform.
- **Automated Email Notifications:** The company is promptly notified via email regarding the approval or rejection of its registration and transaction requests. This automated communication keeps companies informed about the status of their requests in real-time. This allows for improved communication, faster response times, and the ability for companies to take timely action. Package: nodemailer



# OUTPUT

# Challenges And Learning

- Implementing Batch Processing
- Designing relations in database
- Applying cross table filters
- File uploads to storage service
- Implementing responsiveness
- Debugging
- Jwt Authentication
- Extensive error handling
- Database transactions



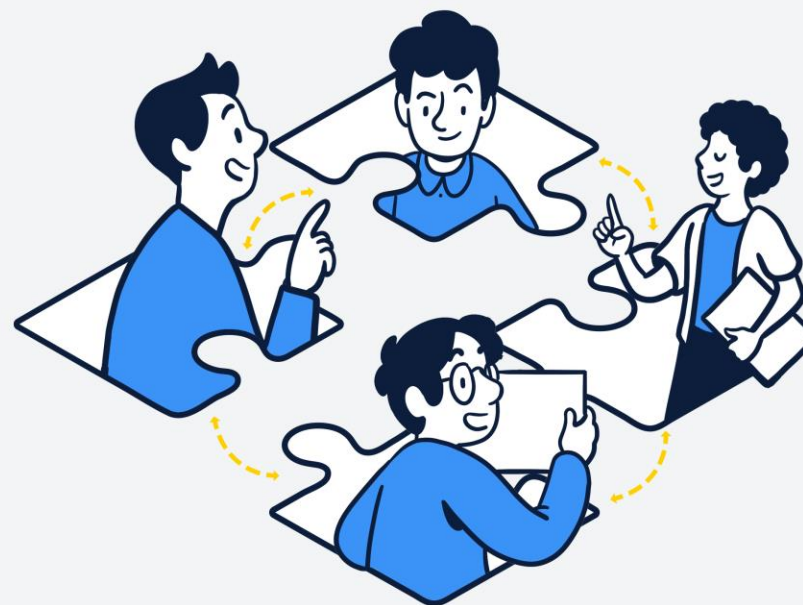
# Challenges And Learning

- Route locking
- Middleware development and implementation
- State management in front end
- Cloud services
- Building and deploying full application
- Clean programing
- Building rest api's with standards
- Serverless deployment
- File parsing
- Asynchronous programming
- Password security using hashing

# Work Distribution

Training Sessions:  
8 Jul to 9 Sept

Project Duration:  
27 Aug to 11 Sept



## Conclusion

- With this Project, we learnt how to develop a full stack project using React.js , Node js and MYSQL database.
- This project gave us a broad idea of how a payments System works in order to serve their customers.

