

CS-GY 6083 - Principles of Database Systems

Project Part II Report (Spring 2025)

Team Members: Siddharthan P S (sp8004), Zexi Peng (zp2277)

Submission Date: May 09, 2025

[Google Slide](#) for presentation

[Github](#)

S.N o	Table of Content	Page No
1.	Executive Summary	1
2.	Logical and Relational Model	2
3.	Assumptions	3
4.	Technical Stack & System Design	5
5.	DDL and Constraints	7
6.	Record Counts of Tables	21
7.	Screenshots of Web Application	21
8.	Security Features Implemented	38
9.	Business SQL Queries (Q1–Q6)	39
10.	Lessons Learned	45

1. Executive Summary:

This project is a role-based event management web application built using modern frontend technologies along with a structured backend and database layer. The main goal of the system is to allow admins and customers to interact with the platform based on their assigned roles, through a secure login and dashboard interface.

For the frontend, we used ReactJS to build a dynamic single-page application (SPA). React Context was used to manage authentication and user session across different components. Routes were protected to ensure only logged-in users can access the dashboard, profile, and other key features.

On the backend side, we used phpMyAdmin to manage our MySQL database. All user data, roles, and event-related information were stored and handled in tables created through phpMyAdmin. We performed SQL queries for login validation, role checking, profile updates, and password changes.

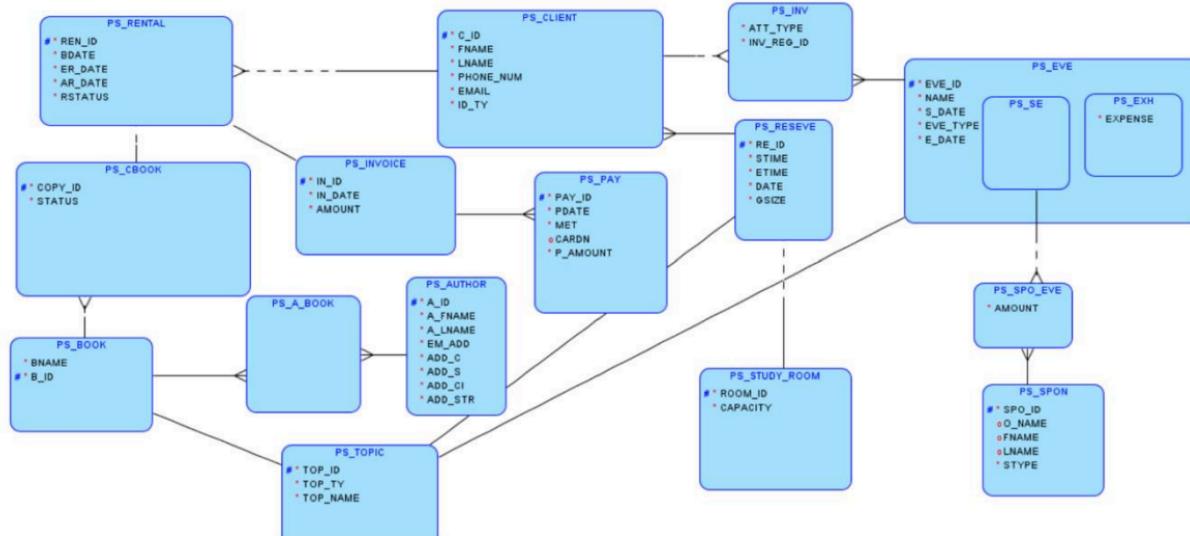
We implemented authentication with a login system, where users are verified against the data in the database. After successful login, user details are stored using **localStorage**, and the app redirects them to the appropriate dashboard.

For authorization, we set up role-based access control. Admins have access to additional features such as user management or stats, while customers have limited access according to their role. In addition, a graphical dashboard was implemented to display data summaries using charts. This visual component enhances the user interface and helps users quickly understand key insights.

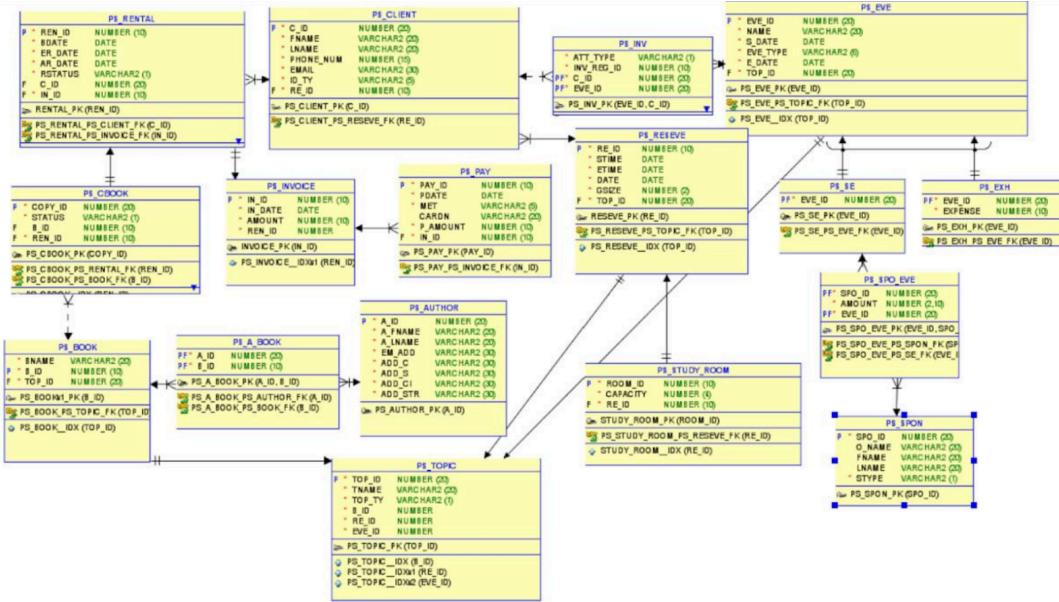
Overall, the project gave us hands-on experience with:

- ReactJS for frontend development
- Context API for session management
- Role-based routing
- MySQL for backend database
- phpMyAdmin for managing and testing SQL queries

2. Logical Representation



Relational Dataset:



3. Assumptions:

Client-Centric System

The system revolves around the **PL_CLIENT** table, which holds core user information like name, email, and phone number. Each client is assigned a unique **C_ID**, which is used as a foreign key in multiple other tables, suggesting clients are the main users of the system.

Book and Rental Management

- Books are stored in the **PL_BOOK** table, each linked to topics and authors.
- **PL_RENTAL** and **PL_EBOOK** suggest the system supports both physical and digital book rentals.
- Each book has a copy (**PL_COPY**) and status tracking, likely for availability and issue tracking.

Invoice and Payment Tracking

- **PL_INVOICE** stores payment-related information, connected to rentals.
- **PL_PAY** logs specific payment transactions with payment type and amount, connected via **PAY_ID**.

Event Management Included

- The presence of **PL_EVENT**, **PL_SPON**, **PL_SE**, and **PL_EXP** indicates the system supports managing **events**.
- Events have sponsors, sessions, and expenses, suggesting detailed event breakdown and planning.

Study Room Reservations

- Clients can reserve study rooms (**PL_STUDY_ROOM**) via the **PL_RESERVE** table.
- Each reservation logs date, time, and room size.

Author and Topic Data Normalized

- **PL_AUTHOR** and **PL_TOPIC** are separated from books, indicating normalization for efficient storage and lookup.

Inventory (INV) System Linked to Events

- **PL_INV** and its connection to **PL_EVENT** via **INV_REG_ID** imply events use inventory items (possibly materials or devices) that are tracked.

Email is Likely Unique

- From the design, it seems **EMAIL** in **PL_CLIENT** may be expected to be unique, though not explicitly marked, likely used for login or communication.

Support for Multiple Users per Role

- The database allows for many-to-many relationships between entities like clients and events, clients and rentals, authors and books.

Data Integrity is Maintained by Foreign Keys

- Most tables are strongly connected with foreign key constraints ensuring that deletion or updates are controlled (referential integrity).

4. Technical Stack & System Design:

Frontend

React.js: Used to build the entire single-page application (SPA), including login forms, dashboard UI, and protected routes.

React Router DOM: For client-side routing and navigation.

React Context API: For managing global authentication state.

Chart.js / Recharts: Used to display visual graphs on the dashboard, summarizing booking or event data.

HTML5 & CSS3: Used along with JSX for UI components and styling.

Backend / Database

phpMyAdmin + MySQL: Used for database design, data entry, and query testing. Tables were created, modified, and connected through phpMyAdmin.

SQL: For querying and manipulating data, managing relationships, and role-based user logic (SELECT, INSERT, UPDATE, JOIN, etc.).

System Design Overview:

1. Authentication & Authorization

Login Flow: Built using React components with form validation. On success, user info is stored in localStorage. Role-Based Access Control (RBAC): Implemented at both route level and component level. UI elements and pages are hidden or shown based on whether the user is an admin or customer.

2. Database Schema (Using phpMyAdmin)

Tables like PL_CLIENT, PL_RENTAL, PL_BOOK, PL_EVENT, and PL_INVOICE are created with foreign key relationships.

phpMyAdmin helped visualize, connect, and enforce integrity using relational diagrams and constraints. Normalized structure separates entities (e.g., books, authors, events, sponsors) to avoid duplication and ensure clean queries.

3. Dashboard Graph

Data is pulled from the backend (e.g., booking count, revenue, usage frequency) and shown using charts. Admins and customers see different datasets based on role.

4. Session Handling

Login state is persistent using localStorage.

Upon page reload, React reads from storage and maintains session across protected components.

5. UI/UX Features

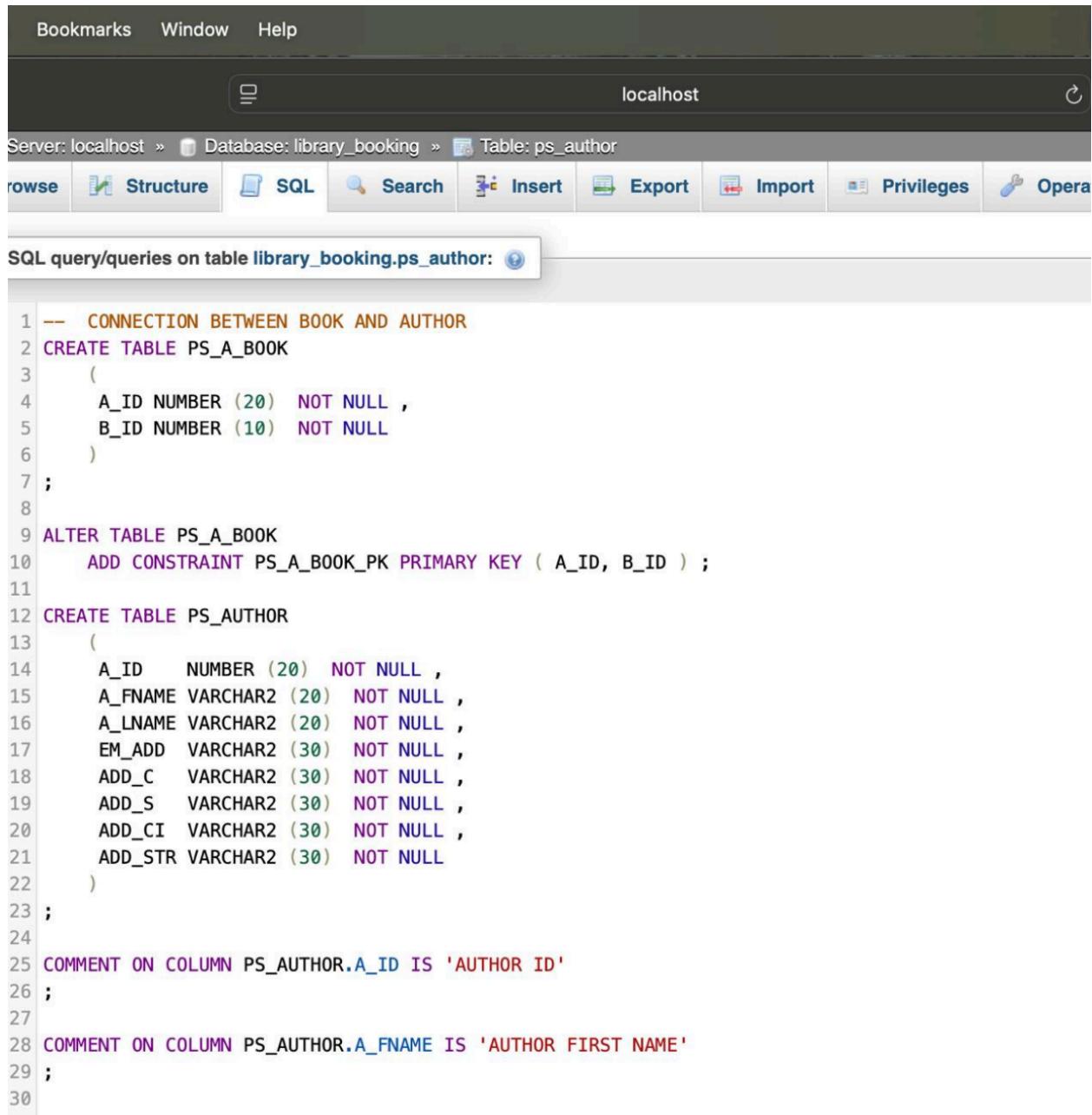
Clean, responsive layout with minimal design.

Basic form validation using HTML required and password matching.

Redirection based on auth state for user flow control.

5. DDL and Constraints:

All DDL scripts for table creation, including constraints and relationships, were defined during Part I and reused with slight changes here.



The screenshot shows the MySQL Workbench interface. The title bar says "localhost". The navigation bar includes "Bookmarks", "Window", "Help", "Structure", "SQL", "Search", "Insert", "Export", "Import", "Privileges", and "Operate". The main area is titled "SQL query/queries on table library_booking.ps_author:". The code listed is:

```
1 -- CONNECTION BETWEEN BOOK AND AUTHOR
2 CREATE TABLE PS_A_BOOK
3 (
4     A_ID NUMBER (20) NOT NULL ,
5     B_ID NUMBER (10) NOT NULL
6 )
7 ;
8
9 ALTER TABLE PS_A_BOOK
10    ADD CONSTRAINT PS_A_BOOK_PK PRIMARY KEY ( A_ID, B_ID ) ;
11
12 CREATE TABLE PS_AUTHOR
13 (
14     A_ID      NUMBER (20) NOT NULL ,
15     A_FNAME   VARCHAR2 (20) NOT NULL ,
16     A_LNAME   VARCHAR2 (20) NOT NULL ,
17     EM_ADD    VARCHAR2 (30) NOT NULL ,
18     ADD_C     VARCHAR2 (30) NOT NULL ,
19     ADD_S     VARCHAR2 (30) NOT NULL ,
20     ADD_CI    VARCHAR2 (30) NOT NULL ,
21     ADD_STR   VARCHAR2 (30) NOT NULL
22 )
23 ;
24
25 COMMENT ON COLUMN PS_AUTHOR.A_ID IS 'AUTHOR ID'
26 ;
27
28 COMMENT ON COLUMN PS_AUTHOR.A_FNAME IS 'AUTHOR FIRST NAME'
29 ;
30
```

Server: localhost » Database: library_booking » Table: ps_author

Browse Structure SQL Search Insert Export Import Privileges Operations

Run SQL query/queries on table library_booking.ps_author:

```
31 COMMENT ON COLUMN PS_AUTHOR.A_LNAME IS 'AUTHOR LAST NAME'
32 ;
33
34 COMMENT ON COLUMN PS_AUTHOR.EM_ADD IS 'EMAIL ADDRESS'
35 ;
36
37 COMMENT ON COLUMN PS_AUTHOR.ADD_C IS 'MAIL ADDRESS FOR COUNTRY'
38 ;
39
40 COMMENT ON COLUMN PS_AUTHOR.ADD_S IS 'MAIL ADDRESS FOR STATE'
41 ;
42
43 COMMENT ON COLUMN PS_AUTHOR.ADD_CI IS 'MAIL ADDRESS FOR CITY'
44 ;
45
46 COMMENT ON COLUMN PS_AUTHOR.ADD_STR IS 'MAIL ADDRESS FOR STREET'
47 ;
48
49 ALTER TABLE PS_AUTHOR
50   ADD CONSTRAINT PS_AUTHOR_PK PRIMARY KEY ( A_ID ) ;
51
52 CREATE TABLE PS_BOOK
53 (
54   BNAME  VARCHAR2 (20) NOT NULL ,
55   B_ID   NUMBER (10) NOT NULL ,
56   TOP_ID NUMBER (20) NOT NULL
57 )
58 ;
59
60 COMMENT ON COLUMN PS_BOOK.BNAME IS 'BOOK NAME'
```

```

63 COMMENT ON COLUMN PS_BOOK.B_ID IS 'BOOK ID'
64 ;
65 CREATE UNIQUE INDEX PS_BOOK__IDX ON PS_BOOK
66 (
67   TOP_ID ASC
68 )
69 ;
70 ;
71 ALTER TABLE PS_BOOK
72   ADD CONSTRAINT PS_BOOKv1_PK PRIMARY KEY ( B_ID ) ;
73 ;
74 -- BOOK COPY
75 CREATE TABLE PS_CBOOK
76 (
77   COPY_ID NUMBER (20) NOT NULL ,
78   STATUS  VARCHAR2 (1) NOT NULL ,
79   B_ID    NUMBER (10) ,
80   REN_ID  NUMBER (10) NOT NULL
81 )
82 ;
83 ;
84 ALTER TABLE PS_CBOOK
85   ADD
86   CHECK (STATUS IN ('A', 'N'))
87 ;
88 ;
89 COMMENT ON COLUMN PS_CBOOK.COPY_ID IS 'BOOK COPY ID'
90 ;
91 ;
92 COMMENT ON COLUMN PS_CBOOK.STATUS IS 'STATUS OF BOOKS :Available'',Not Available'
93 ;

```

ELECT * | SELECT | INSERT | UPDATE | DELETE | Clear | Format | Get auto-saved query

```

93 ;
94 CREATE UNIQUE INDEX PS_CBOOK__IDX ON PS_CBOOK
95 (
96   REN_ID ASC
97 )
98 ;
99 ;
100 ALTER TABLE PS_CBOOK
101   ADD CONSTRAINT PS_CBOOK_PK PRIMARY KEY ( COPY_ID ) ;
102 ;
103 -- STORE CLIENT INFO
104 CREATE TABLE PS_CLIENT
105 (
106   C_ID      NUMBER (20) NOT NULL ,
107   FNAME     VARCHAR2 (20) NOT NULL ,
108   LNAME     VARCHAR2 (20) NOT NULL ,
109   PHONE_NUM NUMBER (15) NOT NULL ,
110   EMAIL     VARCHAR2 (30) NOT NULL ,
111   ID_TY     VARCHAR2 (5) NOT NULL ,
112   RE_ID     NUMBER (10) NOT NULL
113 )
114 ;
115 ;
116 ALTER TABLE PS_CLIENT
117   ADD
118   CHECK (ID_TY IN ('DL', 'PS', 'SSN'))
119 ;
120 ;
121 COMMENT ON COLUMN PS_CLIENT.C_ID IS 'CLIENT ID NUMBER'
122 ;
123 ;

```

```

122 ;
123 COMMENT ON COLUMN PS_CLIENT.FNAME IS 'FIRST NAME'
124 ;
125 ;
126 COMMENT ON COLUMN PS_CLIENT.LNAME IS 'LAST NAME'
127 ;
128 ;
129 COMMENT ON COLUMN PS_CLIENT.PHONE_NUM IS 'PHONE NUMBER'
130 ;
131 ;
132 COMMENT ON COLUMN PS_CLIENT.EMAIL IS 'EMAIL'
133 ;
134 ;
135 COMMENT ON COLUMN PS_CLIENT.ID_TY IS 'ID TYPE:CAN ONLY BE Passport, SSN, or Driver License'
136 ;
137 ;
138 ALTER TABLE PS_CLIENT
139   ADD CONSTRAINT PS_CLIENT_PK PRIMARY KEY ( C_ID ) ;
140 ;
141 -- STORE EVENT
142 CREATE TABLE PS_EVE
143 (
144   EVE_ID    NUMBER (20) NOT NULL ,
145   NAME      VARCHAR2 (20) NOT NULL ,
146   S_DATE    DATE NOT NULL ,
147   EVE_TYPE  VARCHAR2 (6) NOT NULL ,
148   E_DATE    DATE NOT NULL ,
149   TOP_ID    NUMBER (20) NOT NULL
150 )
151 .
152 .

154 ALTER TABLE PS_EVE
155   ADD CONSTRAINT CH_INH_PS_EVE
156     CHECK (EVE_TYPE IN ('PS_EXH', 'SE'))
157 ;
158 ;
159 COMMENT ON COLUMN PS_EVE.EVE_ID IS 'EVENT ID'
160 ;
161 ;
162 COMMENT ON COLUMN PS_EVE.NAME IS 'ACTIVITY NAME'
163 ;
164 ;
165 COMMENT ON COLUMN PS_EVE.S_DATE IS 'START DATE'
166 ;
167 ;
168 COMMENT ON COLUMN PS_EVE.EVE_TYPE IS 'EVENT TYPE:Seminar or Exhibition'
169 ;
170 ;
171 COMMENT ON COLUMN PS_EVE.E_DATE IS 'END DATE'
172 ;
173 CREATE UNIQUE INDEX PS_EVE_IDX ON PS_EVE
174   (
175     TOP_ID ASC
176   )
177 ;
178 ;
179 ALTER TABLE PS_EVE
180   ADD CONSTRAINT PS_EVE_PK PRIMARY KEY ( EVE_ID ) ;
181 ;
182 CREATE TABLE PS_EXH
183   (

```

```

184     EVE_ID NUMBER (20) NOT NULL ,
185     EXPENSE NUMBER (10) NOT NULL
186   )
187 ;
188
189 COMMENT ON COLUMN PS_EXH.EVE_ID IS 'EVENT ID'
190 ;
191
192 ALTER TABLE PS_EXH
193   ADD CONSTRAINT PS_EXH_PK PRIMARY KEY ( EVE_ID ) ;
194
195 CREATE TABLE PS_INV
196   (
197     ATT_TYPE VARCHAR2 (1) NOT NULL ,
198     INV_REG_ID NUMBER (10) NOT NULL ,
199     C_ID      NUMBER (20) NOT NULL ,
200     EVE_ID    NUMBER (20) NOT NULL
201   )
202 ;
203
204 ALTER TABLE PS_INV
205   ADD
206     CHECK (ATT_TYPE IN ('I', 'R'))
207 ;
208
209 COMMENT ON COLUMN PS_INV.ATT_TYPE IS 'ATTENDANCE TYPE: AUTHOR/CUSTOMER'
210 ;
211
212 COMMENT ON COLUMN PS_INV.INV_REG_ID IS 'InvitationOrRegistrationID '
213 ;

214
215 ALTER TABLE PS_INV
216   ADD CONSTRAINT PS_INV_PK PRIMARY KEY ( EVE_ID, C_ID ) ;
217
218 -- INVOICE
219 CREATE TABLE PS_INVOICE
220   (
221     IN_ID   NUMBER (10) NOT NULL ,
222     IN_DATE DATE NOT NULL ,
223     AMOUNT  NUMBER (10) NOT NULL ,
224     REN_ID  NUMBER NOT NULL
225   )
226 ;
227
228 COMMENT ON COLUMN PS_INVOICE.IN_ID IS 'INVOICE ID'
229 ;
230
231 COMMENT ON COLUMN PS_INVOICE.IN_DATE IS 'INVOICE DATE'
232 ;
233
234 COMMENT ON COLUMN PS_INVOICE.AMOUNT IS 'AMOUNT'
235 ;
236 CREATE UNIQUE INDEX PS_INVOICE__IDXv1 ON PS_INVOICE
237   (
238     REN_ID ASC
239   )
240 ;
241
242 ALTER TABLE PS_INVOICE
243   ADD CONSTRAINT INVOICE_PK PRIMARY KEY ( IN_ID ) ;
244

```

```

245 -- PAYMENT
246 CREATE TABLE PS_PAY
247 (
248     PAY_ID    NUMBER (10) NOT NULL ,
249     PDATE     DATE NOT NULL ,
250     MET       VARCHAR2 (5) NOT NULL ,
251     CARDN    VARCHAR2 (20) ,
252     P_AMOUNT  NUMBER (10) NOT NULL ,
253     IN_ID     NUMBER (10) NOT NULL
254 )
255 ;
256
257 ALTER TABLE PS_PAY
258     ADD
259     CHECK (MET IN ('C', 'CR', 'D', 'P'))
260 ;
261
262 COMMENT ON COLUMN PS_PAY.PAY_ID IS 'PAYMENT ID'
263 ;
264
265 COMMENT ON COLUMN PS_PAY.PDATE IS 'PAYMENT DATE'
266 ;
267
268 COMMENT ON COLUMN PS_PAY.MET IS 'METHOD:'''Cash'', ''Credit'', ''Debit'', ''PayPal'''
269 ;
270
271 COMMENT ON COLUMN PS_PAY.CARDN IS 'CARD HOLDER NAME'
272 ;
273
274 COMMENT ON COLUMN PS_PAY.P_AMOUNT IS 'PAYMENT AMOUNT'

277 ALTER TABLE PS_PAY
278     ADD CONSTRAINT PS_PAY_PK PRIMARY KEY ( PAY_ID ) ;
279
280 -- RENTAL SERVICE
281 CREATE TABLE PS_RENTAL
282 (
283     REN_ID    NUMBER (10) NOT NULL ,
284     BDATE     DATE NOT NULL ,
285     ER_DATE   DATE NOT NULL ,
286     AR_DATE   DATE NOT NULL ,
287     RSTATUS   VARCHAR2 (1) NOT NULL ,
288     C_ID      NUMBER (20) ,
289     IN_ID     NUMBER (10) NOT NULL
290 )
291 ;
292
293 ALTER TABLE PS_RENTAL
294     ADD
295     CHECK (RSTATUS IN ('B', 'L', 'R'))
296 ;
297
298 COMMENT ON COLUMN PS_RENTAL.REN_ID IS 'RENTAL ID'
299 ;
300
301 COMMENT ON COLUMN PS_RENTAL.BDATE IS 'BORROW DATE'
302 ;
303
304 COMMENT ON COLUMN PS_RENTAL.ER_DATE IS 'EXPECTED RETURN DATE'
305 ;
306
307 COMMENT ON COLUMN PS_RENTAL.AR_DATE IS 'ActualReturnDate'

```

```

310 COMMENT ON COLUMN PS_RENTAL.RSTATUS IS 'RentalStatus:''Borrowed'', ''Returned'', ''Late'''
311 ;
312 CREATE UNIQUE INDEX PS_RENTAL__IDX ON PS_RENTAL
313 (
314     IN_ID ASC
315 )
316 ;
317
318 ALTER TABLE PS_RENTAL
319     ADD CONSTRAINT RENTAL_PK PRIMARY KEY ( REN_ID ) ;
320
321 CREATE TABLE PS_SEVE
322 (
323     RE_ID NUMBER (10) NOT NULL ,
324     STIME DATE NOT NULL ,
325     ETIME DATE NOT NULL ,
326     "DATE" DATE NOT NULL ,
327     GSIZE NUMBER (2) NOT NULL ,
328     TOP_ID NUMBER (20) NOT NULL
329 )
330 ;
331
332 COMMENT ON COLUMN PS_SEVE.RE_ID IS 'ReservationID'
333 ;
334
335 COMMENT ON COLUMN PS_SEVE.STIME IS 'StartTime'
336 ;
337
338 COMMENT ON COLUMN PS_SEVE.ETIME IS 'End Time'
339 ;
340

341 COMMENT ON COLUMN PS_SEVE.GSIZE IS 'Group Size'
342 ;
343 CREATE UNIQUE INDEX PS_SEVE__IDX ON PS_SEVE
344 (
345     TOP_ID ASC
346 )
347 ;
348
349 ALTER TABLE PS_SEVE
350     ADD CONSTRAINT SEVE_PK PRIMARY KEY ( RE_ID ) ;
351
352 CREATE TABLE PS_SE
353 (
354     EVE_ID NUMBER (20) NOT NULL
355 )
356 ;
357
358 COMMENT ON COLUMN PS_SE.EVE_ID IS 'EVENT ID'
359 ;
360
361 ALTER TABLE PS_SE
362     ADD CONSTRAINT PS_SE_PK PRIMARY KEY ( EVE_ID ) ;
363
364 -- SPONSOR EVENT REALTION SHIP
365 CREATE TABLE PS_SPO_EVE
366 (
367     SPO_ID NUMBER (20) NOT NULL ,
368     AMOUNT NUMBER (2,10) NOT NULL ,
369     EVE_ID NUMBER (20) NOT NULL
370 )

```

```

373 COMMENT ON COLUMN PS_SPO_EVE.AMOUNT IS 'AMOUNT OF SPONSORED MONEY/COST FOR SEMINAR/EXHIBITION '
374 ;
375
376 ALTER TABLE PS_SPO_EVE
377   ADD CONSTRAINT PS_SPO_EVE_PK PRIMARY KEY ( EVE_ID, SPO_ID ) ;
378
379 -- STORE SPONSOR INFO
380 CREATE TABLE PS_SPON
381 (
382   SPO_ID NUMBER (20) NOT NULL ,
383   O_NAME VARCHAR2 (20) ,
384   FNAME VARCHAR2 (20) ,
385   LNAME VARCHAR2 (20) ,
386   STYPE VARCHAR2 (1) NOT NULL
387 )
388 ;
389
390 ALTER TABLE PS_SPON
391   ADD
392     CHECK (STYPE IN ('I', 'O'))
393 ;
394
395 COMMENT ON COLUMN PS_SPON.SPO_ID IS 'SPONSOR ID'
396 ;
397
398 COMMENT ON COLUMN PS_SPON.O_NAME IS 'ORGINAZATION NAME'
399 ;
400
401 COMMENT ON COLUMN PS_SPON.FNAME IS 'FIRST NAME'
402 ;
403
404 COMMENT ON COLUMN PS_SPON.LNAME IS 'LAST NAME'
405 ;
406
407
408 COMMENT ON COLUMN PS_SPON.STYPE IS 'Sponsor Type'
409 ;
410
411 ALTER TABLE PS_SPON
412   ADD CONSTRAINT PS_SPON_PK PRIMARY KEY ( SPO_ID ) ;
413
414 CREATE TABLE PS_STUDY_ROOM
415 (
416   ROOM_ID NUMBER (10) NOT NULL ,
417   CAPACITY NUMBER (4) NOT NULL ,
418   RE_ID NUMBER (10) NOT NULL
419 )
420 ;
421
422 COMMENT ON COLUMN PS_STUDY_ROOM.ROOM_ID IS 'STUDY ROOM ID'
423 ;
424
425 COMMENT ON COLUMN PS_STUDY_ROOM.CAPACITY IS 'CAPACITY OF THE ROOM'
426 ;
427 CREATE UNIQUE INDEX STUDY_ROOM__IDX ON PS_STUDY_ROOM
428 (
429   RE_ID ASC
430 )
431 ;
432
433 ALTER TABLE PS_STUDY_ROOM
434   ADD CONSTRAINT STUDY_ROOM_PK PRIMARY KEY ( ROOM_ID ) ;

```

```
434 ADD CONSTRAINT STUDY_ROOM_PK PRIMARY KEY ( ROOM_ID ) ;
435
436 CREATE TABLE PS_TOPIC
437 (
438   TOP_ID NUMBER (20) NOT NULL ,
439   TNAME VARCHAR2 (20) NOT NULL ,
440   TOP_TY VARCHAR2 (1) NOT NULL ,
441   B_ID NUMBER NOT NULL ,
442   RE_ID NUMBER NOT NULL ,
443   EVE_ID NUMBER NOT NULL
444 )
445 ;
446
447 ALTER TABLE PS_TOPIC
448   ADD
449     CHECK (TOP_TY IN ('B', 'E', 'R'))
450 ;
451
452 COMMENT ON COLUMN PS_TOPIC.TOP_ID IS 'TOPIC ID'
453 ;
454
455 COMMENT ON COLUMN PS_TOPIC.TNAME IS 'TOPIC NAME'
456 ;
457
458 COMMENT ON COLUMN PS_TOPIC.TOP_TY IS 'TOPIC TYPE'
459 ;
460 CREATE UNIQUE INDEX PS_TOPIC__IDX ON PS_TOPIC
461 (
462   B_ID ASC
463 )
464 ;
```

```

465 CREATE UNIQUE INDEX PS_TOPIC__IDXv1 ON PS_TOPIC
466 (
467     RE_ID ASC
468 )
469 ;
470 CREATE UNIQUE INDEX PS_TOPIC__IDXv2 ON PS_TOPIC
471 (
472     EVE_ID ASC
473 )
474 ;
475
476 ALTER TABLE PS_TOPIC
477     ADD CONSTRAINT PS_TOPIC_PK PRIMARY KEY ( TOP_ID ) ;
478
479 ALTER TABLE PS_A_BOOK
480     ADD CONSTRAINT PS_A_BOOK_PS_AUTHOR_FK FOREIGN KEY
481 (
482     A_ID
483 )
484 REFERENCES PS_AUTHOR
485 (
486     A_ID
487 )
488 ;
489
490 ALTER TABLE PS_A_BOOK
491     ADD CONSTRAINT PS_A_BOOK_PS_BOOK_FK FOREIGN KEY
492 (
493     B_ID
494 )

```

```

495     REFERENCES PS_BOOK
496 (
497     B_ID
498 )
499 ;
500
501 ALTER TABLE PS_BOOK
502     ADD CONSTRAINT PS_BOOK_PS_TOPIC_FK FOREIGN KEY
503 (
504     TOP_ID
505 )
506 REFERENCES PS_TOPIC
507 (
508     TOP_ID
509 )
510 ;
511
512 ALTER TABLE PS_CBOOK
513     ADD CONSTRAINT PS_CBOOK_PS_BOOK_FK FOREIGN KEY
514 (
515     B_ID
516 )
517 REFERENCES PS_BOOK
518 (
519     B_ID
520 )
521 ;
522
523 ALTER TABLE PS_CBOOK
524     ADD CONSTRAINT PS_CBOOK_PS_RENTAL_FK FOREIGN KEY
525

```

```

525 (
526   REN_ID
527 )
528 REFERENCES PS_RENTAL
529 (
530   REN_ID
531 )
532 ;
533
534 ALTER TABLE PS_CLIENT
535   ADD CONSTRAINT PS_CLIENT_PS_SEVE_FK FOREIGN KEY
536   (
537     RE_ID
538   )
539 REFERENCES PS_SEVE
540   (
541     RE_ID
542   )
543 ;
544
545 ALTER TABLE PS_EVE
546   ADD CONSTRAINT PS_EVE_PS_TOPIC_FK FOREIGN KEY
547   (
548     TOP_ID
549   )
550 REFERENCES PS_TOPIC
551   (
552     TOP_ID
553   )
554 ;
555
556 ALTER TABLE PS_EXH
557   ADD CONSTRAINT PS_EXH_PS_EVE_FK FOREIGN KEY
558   (
559     EVE_ID
560   )
561 REFERENCES PS_EVE
562   (
563     EVE_ID
564   )
565 ;
566
567 ALTER TABLE PS_INV
568   ADD CONSTRAINT PS_INV_PS_CLIENT_FK FOREIGN KEY
569   (
570     C_ID
571   )
572 REFERENCES PS_CLIENT
573   (
574     C_ID
575   )
576 ;
577
578 ALTER TABLE PS_INV
579   ADD CONSTRAINT PS_INV_PS_EVE_FK FOREIGN KEY
580   (
581     EVE_ID
582   )
583 REFERENCES PS_EVE
584   (
585     EVE_ID
586   )

```

```

589 ALTER TABLE PS_PAY
590   ADD CONSTRAINT PS_PAY_PS_INVOICE_FK FOREIGN KEY
591   (
592     IN_ID
593   )
594   REFERENCES PS_INVOICE
595   (
596     IN_ID
597   )
598 ;
599
600 ALTER TABLE PS_RENTAL
601   ADD CONSTRAINT PS_RENTAL_PS_CLIENT_FK FOREIGN KEY
602   (
603     C_ID
604   )
605   REFERENCES PS_CLIENT
606   (
607     C_ID
608   )
609 ;
610
611 ALTER TABLE PS_RENTAL
612   ADD CONSTRAINT PS_RENTAL_PS_INVOICE_FK FOREIGN KEY
613   (
614     IN_ID
615   )
616   REFERENCES PS_INVOICE
617   (
618     IN_ID
619   )
620
621 ALTER TABLE PS_SE
622   ADD CONSTRAINT PS_SE_PS_TOPIC_FK FOREIGN KEY
623   (
624     TOP_ID
625   )
626   REFERENCES PS_TOPIC
627   (
628     TOP_ID
629   )
630
631 ;
632
633 ALTER TABLE PS_SE
634   ADD CONSTRAINT PS_SE_PS_EVE_FK FOREIGN KEY
635   (
636     EVE_ID
637   )
638   REFERENCES PS_EVE
639   (
640     EVE_ID
641   )
642 ;
643
644 ALTER TABLE PS_SPO_EVE
645   ADD CONSTRAINT PS_SPO_EVE_PS_SE_FK FOREIGN KEY
646   (
647     EVE_ID
648   )
649   REFERENCES PS_SE
650   (
651     EVE_ID
652

```

```

655 ALTER TABLE PS_SPO_EVE
656   ADD CONSTRAINT PS_SPO_EVE_PS_SPON_FK FOREIGN KEY
657   (
658     SPO_ID
659   )
660   REFERENCES PS_SPON
661   (
662     SPO_ID
663   )
664 ;
665
666 ALTER TABLE PS_STUDY_ROOM
667   ADD CONSTRAINT PS_STUDY_ROOM_PS_SESEVE_FK FOREIGN KEY
668   (
669     RE_ID
670   )
671   REFERENCES PS_SESEVE
672   (
673     RE_ID
674   )
675 ;
676
677 CREATE OR REPLACE TRIGGER ARC_FKArc_1_PS_SE
678 BEFORE INSERT OR UPDATE OF EVE_ID
679 ON PS_SE
680 FOR EACH ROW
681 DECLARE
682   d VARCHAR2 (6);
683 BEGIN
684   SELECT A.EVE_TYPE INTO d
685
686   FROM PS_EVE A
687   WHERE A.EVE_ID = :new.EVE_ID;
688   IF (d IS NULL OR d <> 'SE') THEN
689     raise_application_error(-20223,'FK PS_SE_PS_EVE_FK in Table PS_SE violates Arc constraint on Table PS_EVE
- discriminator column EVE_TYPE doesn''t have value ''SE'''');
690   END IF;
691   EXCEPTION
692   WHEN NO_DATA_FOUND THEN
693     NULL;
694   WHEN OTHERS THEN
695     RAISE;
696 END;
697 /
698
699 CREATE OR REPLACE TRIGGER ARC_FKArc_1_PS_EXH
700 BEFORE INSERT OR UPDATE OF EVE_ID
701 ON PS_EXH
702 FOR EACH ROW
703 DECLARE
704   d VARCHAR2 (6);
705 BEGIN
706   SELECT A.EVE_TYPE INTO d
707   FROM PS_EVE A
708   WHERE A.EVE_ID = :new.EVE_ID;
709   IF (d IS NULL OR d <> 'PS_EXH') THEN
710     raise_application_error(-20223,'FK PS_EXH_PS_EVE_FK in Table PS_EXH violates Arc constraint on Table
PS_EVE - discriminator column EVE_TYPE doesn''t have value ''PS_EXH'''');
711   END IF;
712   EXCEPTION
713   WHEN NO_DATA_FOUND THEN
714     NULL;

```

```

713      NULL;
714      WHEN OTHERS THEN
715          RAISE;
716  END;
717 /
718
719
720
721 -- Oracle SQL Developer Data Modeler 概要报告:
722 --
723 -- CREATE TABLE                      17
724 -- CREATE INDEX                       10
725 -- ALTER TABLE                        43
726 -- CREATE VIEW                         0
727 -- ALTER VIEW                          0
728 -- CREATE PACKAGE                     0
729 -- CREATE PACKAGE BODY                0
730 -- CREATE PROCEDURE                   0
731 -- CREATE FUNCTION                    0
732 -- CREATE TRIGGER                     2
733 -- ALTER TRIGGER                     0
734 -- CREATE COLLECTION TYPE            0
735 -- CREATE STRUCTURED TYPE           0
736 -- CREATE STRUCTURED TYPE BODY       0
737 -- CREATE CLUSTER                     0
738 -- CREATE CONTEXT                     0
739 -- CREATE DATABASE                   0
740 -- CREATE DIMENSION                  0
741 -- CREATE DIRECTORY                  0
742 -- CREATE DISK GROUP                 0
743 -- CREATE ROLE                        0
744 -- ALTER TRIGGER                     0
745 -- CREATE COLLECTION TYPE            0
746 -- CREATE STRUCTURED TYPE           0
747 -- CREATE STRUCTURED TYPE BODY       0
748 -- CREATE CLUSTER                     0
749 -- CREATE CONTEXT                     0
750 -- CREATE DATABASE                   0
751 -- CREATE DIMENSION                  0
752 -- CREATE DIRECTORY                  0
753 -- CREATE DISK GROUP                 0
754 -- CREATE ROLE                        0
755 -- CREATE ROLLBACK SEGMENT           0
756 -- CREATE SEQUENCE                   0
757 -- CREATE MATERIALIZED VIEW         0
758 -- CREATE MATERIALIZED VIEW LOG     0
759 -- CREATE SYNONYM                    0
760 -- CREATE TABLESPACE                 0
761 -- CREATE USER                        0
762 -- DROP TABLESPACE                  0
763 -- DROP DATABASE                    0
764 -- REDACTION POLICY                 0
765 -- ORDS DROP SCHEMA                 0
766 -- ORDS ENABLE SCHEMA                0
767 -- ORDS ENABLE OBJECT                0
768 --
769 -- ERRORS                           0
770 -- WARNINGS                          0

```

6. Record Count Per Table:

Query result Script output DBMS output Explain Plan SQL history

trash info Download ▾ Execution time: 0.019 seconds

	TABLE_NAME	RECORD_COUNT
1	PS_CLIENT	30
2	PS_BOOK	30
3	PS_EVE	30
4	PS_RENTAL	30
5	PS_INVOICE	30
6	PS_PAY	30

7. Screenshots of Web Application:

Admin Dashboard:

The screenshot shows the Admin Dashboard of the LibraryPlus application. The sidebar on the left contains a navigation menu with links for Dashboard, Books, Study Rooms, Events, Clients, Rentals, Invoices, and Profile. The main content area is titled "Admin Dashboard" and includes the following information:

- Total Books:** 45 (12 checked out)
- Study Rooms:** 30 (8 currently booked)
- Upcoming Events:** 30 (Events scheduled)
- Active Clients:** 30 (Registered authors & clients)
- Revenue Today:** \$155.00 (Total payments received today)
- Pending Revenue:** \$85.00 (Awaiting payment)

Below these metrics are two charts:

- Weekly Activity:** A bar chart showing the number of books, rooms, and events for each day of the week. The x-axis lists the days from Sun to Sat, and the y-axis ranges from 0 to 2.
- Recent Activity:** A list of recent events and rentals, including:
 - New event: Kerala Culture & History - 2357 minutes ago
 - New event: Software Development Best Practices - 917 minutes ago
 - Priya Natarajan checked out "Tamil Classical Literature" - 11 days ago
 - Raghav Menon checked out "Historical Kerala" - 12 days ago
 - Aishwarya Mehta checked out "Modern Software Engineering" - 13 days ago

LibraryPlus

Welcome, Admin User

Admin Dashboard

- [Dashboard](#)
- [Books](#)
- [Study Rooms](#)
- [Events](#)
- [Clients](#)
- [Rentals](#)
- [Invoices](#)
- [Profile](#)

Total Books

45
12 checked out

Study Rooms

30
8 currently booked

Upcoming Events

30
Events scheduled

Active Clients

30
Registered authors & clients

Revenue Today

\$155.00
Total payments received today

Pending Revenue

\$85.00
Awaiting payment

Weekly Activity
Books, rooms, and events this week

Day	Count
Sun	1.0
Mon	1.0
Tue	1.0
Wed	1.0
Thu	2.0
Fri	2.0
Sat	2.0

Recent Activity
Latest rentals, bookings, clients and events

- New event: Kerala Culture & History
-18 minutes ago
- New event: Software Development Best Practices
-43 minutes ago
- Priya Natarajan checked out "Tamil Classical Literature"
11 days ago
- Raghav Menon checked out "Historical Kerala"
12 days ago
- Aishwarya Mehta checked out "Modern Software Engineering"
13 days ago

Books Dashboard:

LibraryPlus

Welcome, Admin User

Books

- [Dashboard](#)
- [**Books**](#)
- [Study Rooms](#)
- [Events](#)
- [Clients](#)
- [Rentals](#)
- [Invoices](#)
- [Profile](#)

OPERATING SYSTEM CONCEPTS

Abraham Silberschatz
Peter Baer Galvin
Greg Gagne

Ninth Edition

Historical Kerala

by Raghav Menon
History 2020

Tamil Classical Literature

by Priya Natarajan
Literature 2018

Search books by title or author...
▼ Topics
▼ Availability

Room reservations:

Study Rooms

05/14/2025

Select client for booking | 09:00 to 11:00

Room Name	Capacity	Available	Action
Quiet Haven	4 people	Available	Book Room
Tech Hub	6 people	Available	Book Room
Focus Zone	8 people	Available	Book Room
Collaborative Corner	6 people	Available	Book Room
Cozy Nook	2 people	Available	Book Room
Conference Central	10 people	Available	Book Room
Research Retreat	Reference Materials, Ergonomic Chairs	Available	
Group Study Garden	Natural Light, Plants	Available	
Brainstorm Bay	Wall-to-Wall Whiteboards, Flexible Layout	Available	Book Room

Library Events:

Library Events

+ Create Event

Search events...

Event Name	Category	Date	Location	Attendees	Action
Software Development Best Practices	Technology	May 14, 2025	Main Hall	7	Manage Event
Kerala Culture & History	History	May 15, 2025	Cultural Center	7	Manage Event
Tamil Literature Appreciation	Literature	May 15, 2025	Reading Room	7	Manage Event
"Through Ages: A Historical Tour	History				
Pakistan Cultural Heritage	Culture				

Invoices:

The screenshot shows the LibraryPlus software interface. The top navigation bar includes a logo, a search bar, and a welcome message for "Admin User". The left sidebar has a "Invoices" button highlighted in blue, indicating the current active section. The main content area displays a table of invoices with columns for Invoice ID, Date, Client, Type, Amount, and Status. Each row shows details for a specific invoice, such as INV-0001 for Aishwarya Mehta on April 30, 2025, for a book worth \$120.00, marked as Paid.

Invoice ID	Date	Client	Type	Amount	Status
INV-0001	Apr 30, 2025	Aishwarya Mehta Book: Introduction to React		\$120.00	Paid
INV-0002	May 01, 2025	Raghav Menon Room: Study Room A		\$110.00	Pending
INV-0003	May 02, 2025	Priya Natarajan Book: Database Systems		\$115.00	Paid
INV-0004	May 03, 2025	Neeraj Kumar Book: JavaScript Fundamentals		\$120.00	Overdue
INV-0005	May 04, 2025	Lakshmi Das Room: Conference Room B		\$125.00	Paid

Create New Invoice:

Create New Invoice

X

Enter the details for the new invoice.

Client

Select client

Type

Book Rental

Book Rental

Select rental

Date

May 14th, 2025



Amount (₹)

0.00

Status

Pending

Cancel

Create Invoice

Create New Invoice:

Invoices

Invoice ID	Date	Client	Type	Amount	Status
INV-0001	Apr 30, 2025	Aishwarya Mehta Book: Introduction to React		\$120.00	Paid
INV-0002	May 01, 2025	Raghav Menon Room: Study Room A		\$110.00	Pending
INV-0003	May 02, 2025	Priya Natarajan Book: Database Systems		\$115.00	Paid
INV-0004	May 03, 2025	Neeraj Kumar Book: JavaScript Fundamentals		\$120.00	Overdue
INV-0005	May 04, 2025	Lakshmi Das Room: Conference Room B		\$125.00	Paid

Invoice created
Invoice for 115 has been created

Client Management:

Client Management

Aishwarya Mehta aishi@mail.com 9876543201 Member since December 31, 2023	Raghav Menon raghav2@mail.com 9876543202 Member since January 14, 2024	Priya Natarajan priya3@mail.com 9876543203 Member since January 31, 2024
Neeraj Kumar neeraj4@mail.com 9876543204 Member since February 14, 2024	Lakshmi Das lakshmi5@mail.com 9876543205 Member since February 29, 2024	Anuj Bhatt anuj6@mail.com 9876543206 Member since March 14, 2024
Kavya Rao kavya7@mail.com 9876543207 Member since March 31, 2024	Rohit Shah rohit8@mail.com 9876543208 Member since April 14, 2024	Sneha Iyer sneha9@mail.com 9876543209 Member since April 30, 2024
Vikram Patel	Meera Jain	Nisha Kohli

View Client Statement:

The screenshot shows the LibraryPlus Admin Panel with the 'Clients' tab selected. A modal window is open for 'Aishwarya Mehta', displaying her client details and an activity summary. The activity summary shows 1 book borrowed, 1 room booking, and total payments of \$120.00. Below this, a specific book rental record is shown for 'book1', borrowed on April 30, 2025, and returned on May 5, 2025, with a status of 'PAID'. The background shows other client profiles like Neeraj Kumar, Kavya Rao, Priya Natarajan, Anuj Bhatt, Sneha Iyer, Vikram Patel, Meera Jain, and Nisha Kohli.

The screenshot shows the LibraryPlus Admin Panel with the 'Clients' tab selected. An 'Add New Client' modal is open, prompting the user to fill in details for a new client. The fields include Name (sid), Email (sid@gmail.com), Phone (12384190), ID Number (2132), and Address (247). The 'Add Client' button is visible at the bottom right of the modal. The background shows the same list of clients as the previous screenshot.

LibraryPlus

Welcome, Admin User

Admin Panel

- Dashboard
- Books
- Study Rooms
- Events
- Clients**
- Rentals
- Invoices
- Profile

Client Management

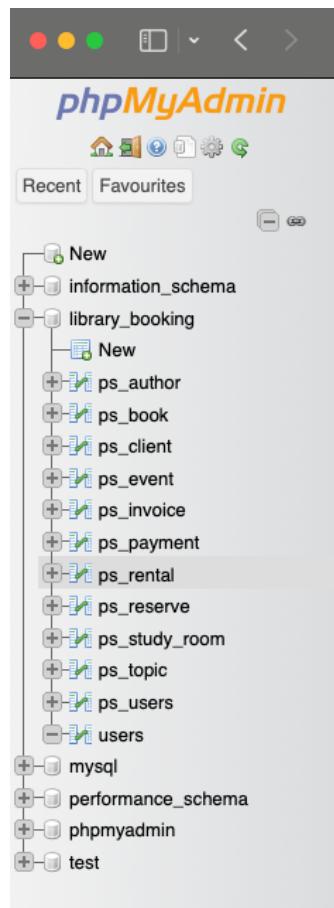
+ Add Client

Search clients by name, email, or phone...

Aishwarya Mehta aishi@mail.com 9876543201 Member since December 31, 2023	Raghav Menon raghav2@mail.com 9876543202 Member since January 14, 2024	Priya Natarajan priva3@mail.com 9876543203 Member since January 31, 2024
View Details	View Details	View Details
Neeraj Kumar neeraj4@mail.com 9876543204 Member since February 14, 2024	Lakshmi Das lakshmi5@mail.com 9876543205 Member since February 29, 2024	Anuj Bhatt anuj6@mail.com 9876543206 Member since March 14, 2024
View Details	View Details	View Details
Kavya Rao kavya7@mail.com 9876543207 Member since March 31, 2024	Rohit Shah rohit8@mail.com 9876543208 Member since April 14, 2024	Sneha Iyer sneha9@mail.com 9876543209 Member since April 30, 2024
View Details	View Details	View Details
Vikram Patel	Meera Jain	Nisha Kohli

Client added
sid has been successfully added to the system.

phpMyAdmin:



Author:

SELECT * FROM `ps_author`

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

	<input type="button" value="1"/>	>	>>	<input type="checkbox"/> Show all	Number of rows: 25	Filter rows:	Search this table	Sort by key: None			
Extra options											
	<input type="button" value="←"/>	<input type="button" value="→"/>	<input type="button" value="▼"/>	a_id	a_fname	a_lname	em_add	add_c	add_s	add_ci	add_str
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	1	Aishwarya	Mehta	aish1@mail.com	India	MH	Mumbai	Andheri
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	2	Raghav	Menon	raghav2@mail.com	India	KL	Cochin	MG Road
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	3	Priya	Natarajan	priya3@mail.com	India	TN	Chennai	Adyar
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	4	Neeraj	Kumar	neeraj4@mail.com	India	DL	Delhi	Saket
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	5	Lakshmi	Das	lakshmi5@mail.com	India	WB	Kolkata	Salt Lake
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	6	Anuj	Bhatt	anuj6@mail.com	India	RJ	Jaipur	Malviya Nagar
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	7	Kavya	Rao	kavya7@mail.com	India	KA	Bangalore	Koramangala
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	8	Rohit	Shah	rohit8@mail.com	India	GJ	Ahmedabad	CG Road
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	9	Sneha	Iyer	sneha9@mail.com	India	TN	Chennai	T Nagar
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	10	Vikram	Patel	vikram10@mail.com	India	MH	Pune	Baner
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	11	Meera	Jain	meera11@mail.com	India	KA	Mysore	Vijayanagar
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	12	Nisha	Kohli	nisha12@mail.com	India	PB	Ludhiana	Civil Lines
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	13	Ramesh	Tiwari	ramesh13@mail.com	India	UP	Lucknow	Hazratganj
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	14	Divya	Singh	divya14@mail.com	India	MP	Indore	Palasia
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	15	Harsha	Ghosh	harsha15@mail.com	India	WB	Kolkata	New Alipore
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	16	Amit	Chopra	amit16@mail.com	India	DL	Delhi	Dwarka
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	17	Bhavna	Roy	bhavna17@mail.com	India	BR	Patna	Kankarbagh
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	18	Sunil	Das	sunil18@mail.com	India	OD	Bhubaneswar	Jaydev Vihar
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	19	Geeta	Kumar	geeta19@mail.com	India	UP	Noida	Sector 18
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	20	Anjali	Malik	anjali20@mail.com	India	HR	Gurgaon	DLF Phase 3
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	21	Naveen	Reddy	naveen21@mail.com	India	AP	Vijayawada	Labbipet
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	22	Chitra	Bose	chitra22@mail.com	India	WB	Howrah	Shibpur
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	23	Tushar	Joshi	tushar23@mail.com	India	MH	Nagpur	Dharampeth
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	24	Lavanya	Shekhar	lavanya24@mail.com	India	KA	Hubli	Vidyanagar
<input checked="" type="checkbox"/>	<input type="button" value="Console"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	25	Sandeep	Verma	sandeep25@mail.com	India	RJ	Jodhpur	Paota

Books:

SELECT * FROM `ps_book`

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

1 < > >> | Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

	b_id	bname	top_id
<input type="checkbox"/>	1	Book A	1
<input type="checkbox"/>	2	Book 2	NULL
<input type="checkbox"/>	3	Book 3	NULL
<input type="checkbox"/>	4	Book 4	NULL
<input type="checkbox"/>	5	Book 5	NULL
<input type="checkbox"/>	6	Book 6	NULL
<input type="checkbox"/>	7	Book 7	NULL
<input type="checkbox"/>	8	Book 8	NULL
<input type="checkbox"/>	9	Book 9	NULL
<input type="checkbox"/>	10	Book 10	NULL
<input type="checkbox"/>	11	Book 11	NULL
<input type="checkbox"/>	12	Book 12	NULL
<input type="checkbox"/>	13	Book 13	NULL
<input type="checkbox"/>	14	Book 14	NULL
<input type="checkbox"/>	15	Book 15	NULL
<input type="checkbox"/>	16	Book 16	NULL
<input type="checkbox"/>	17	Book 17	NULL
<input type="checkbox"/>	18	Book 18	NULL
<input type="checkbox"/>	19	Book 19	NULL
<input type="checkbox"/>	20	Book 20	NULL
<input type="checkbox"/>	21	Book 21	NULL
<input type="checkbox"/>	22	Book 22	NULL
<input type="checkbox"/>	23	Book 23	NULL
<input type="checkbox"/>	24	Book 24	NULL
<input type="checkbox"/>	25	Book 25	NULL

Client:

SELECT * FROM `ps_client`

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

1 > >> | Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Extra options

		c_id	fname	lname	phone_num	email	id_ty	re_id
<input type="checkbox"/>	Edit Copy Delete	1	Sneha	Patel	9876500001	sneha1@example.com	PAN	1
<input type="checkbox"/>	Edit Copy Delete	2	Arjun	Verma	9876500002	arjun2@example.com	AAD	2
<input type="checkbox"/>	Edit Copy Delete	3	Divya	Iyer	9876500003	divya3@example.com	DL	3
<input type="checkbox"/>	Edit Copy Delete	4	Karthik	Nair	9876500004	karthik4@example.com	VOT	4
<input type="checkbox"/>	Edit Copy Delete	5	Riya	Sharma	9876500005	riya5@example.com	PAN	5
<input type="checkbox"/>	Edit Copy Delete	6	Ajay	Pillai	9876500006	ajay6@example.com	AAD	6
<input type="checkbox"/>	Edit Copy Delete	7	Megha	Bansal	9876500007	megha7@example.com	PAN	7
<input type="checkbox"/>	Edit Copy Delete	8	Tarun	Joshi	9876500008	tarun8@example.com	DL	8
<input type="checkbox"/>	Edit Copy Delete	9	Latha	Krishnan	9876500009	latha9@example.com	VOT	9
<input type="checkbox"/>	Edit Copy Delete	10	Nikhil	Desai	9876500010	nikhil10@example.com	PAN	10
<input type="checkbox"/>	Edit Copy Delete	11	Sara	Kapoor	9876500011	sara11@example.com	AAD	11
<input type="checkbox"/>	Edit Copy Delete	12	Manoj	Shetty	9876500012	manoj12@example.com	PAN	12
<input type="checkbox"/>	Edit Copy Delete	13	Ishita	Rao	9876500013	ishita13@example.com	DL	13
<input type="checkbox"/>	Edit Copy Delete	14	Kunal	Jain	9876500014	kunal14@example.com	VOT	14
<input type="checkbox"/>	Edit Copy Delete	15	Radhika	Sen	9876500015	radhika15@example.com	PAN	15
<input type="checkbox"/>	Edit Copy Delete	16	Yash	Bhatt	9876500016	yash16@example.com	AAD	16
<input type="checkbox"/>	Edit Copy Delete	17	Aditi	Mehra	9876500017	aditi17@example.com	PAN	17
<input type="checkbox"/>	Edit Copy Delete	18	Ravi	Mishra	9876500018	ravi18@example.com	DL	18
<input type="checkbox"/>	Edit Copy Delete	19	Tanvi	Gupta	9876500019	tanvi19@example.com	VOT	19
<input type="checkbox"/>	Edit Copy Delete	20	Nitin	Chawla	9876500020	nitin20@example.com	PAN	20
<input type="checkbox"/>	Edit Copy Delete	21	Priya	Dixit	9876500021	priya21@example.com	AAD	21
<input type="checkbox"/>	Edit Copy Delete	22	Aman	Goel	9876500022	aman22@example.com	PAN	22
<input type="checkbox"/>	Edit Copy Delete	23	Neha	Malhotra	9876500023	neha23@example.com	DL	23
<input type="checkbox"/>	Edit Copy Delete	24	Abhishek	Singh	9876500024	abhishek24@example.com	VOT	24
<input type="checkbox"/>	Edit Copy Delete	25	Shruti	Agarwal	9876500025	shruti25@example.com	PAN	25

Event:

Showing rows 0 - 24 (31 total, Query took 0.0001 seconds.)

```
SELECT * FROM `ps_event`
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

1 < > >> | Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

	eve_id	name	s_date	eve_type	e_date	top_id
<input type="checkbox"/>	1	Event 1	2025-05-01	Seminar	2025-05-01	1
<input type="checkbox"/>	2	Event 2	2025-05-02	Workshop	2025-05-02	2
<input type="checkbox"/>	3	Event 3	2025-05-03	Seminar	2025-05-03	3
<input type="checkbox"/>	4	Event 4	2025-05-04	Seminar	2025-05-04	4
<input type="checkbox"/>	5	Event 5	2025-05-05	Workshop	2025-05-05	5
<input type="checkbox"/>	6	Event 6	2025-05-06	Seminar	2025-05-06	6
<input type="checkbox"/>	7	Event 7	2025-05-07	Seminar	2025-05-07	7
<input type="checkbox"/>	8	Event 8	2025-05-08	Seminar	2025-05-08	8
<input type="checkbox"/>	9	Event 9	2025-05-09	Workshop	2025-05-09	9
<input type="checkbox"/>	10	Event 10	2025-05-10	Seminar	2025-05-10	10
<input type="checkbox"/>	11	Event 11	2025-05-11	Workshop	2025-05-11	11
<input type="checkbox"/>	12	Event 12	2025-05-12	Seminar	2025-05-12	12
<input type="checkbox"/>	13	Event 13	2025-05-13	Seminar	2025-05-13	13
<input type="checkbox"/>	14	Event 14	2025-05-14	Seminar	2025-05-14	14
<input type="checkbox"/>	15	Event 15	2025-05-15	Seminar	2025-05-15	15
<input type="checkbox"/>	16	Event 16	2025-05-16	Workshop	2025-05-16	16
<input type="checkbox"/>	17	Event 17	2025-05-17	Seminar	2025-05-17	17
<input type="checkbox"/>	18	Event 18	2025-05-18	Seminar	2025-05-18	18
<input type="checkbox"/>	19	Event 19	2025-05-19	Seminar	2025-05-19	19
<input type="checkbox"/>	20	Event 20	2025-05-20	Workshop	2025-05-20	20
<input type="checkbox"/>	21	Event 21	2025-05-21	Seminar	2025-05-21	21
<input type="checkbox"/>	22	Event 22	2025-05-22	Seminar	2025-05-22	22
<input type="checkbox"/>	23	Event 23	2025-05-23	Workshop	2025-05-23	23
<input checked="" type="checkbox"/>	24	Event 24	2025-05-24	Seminar	2025-05-24	24

Invoices:

Showing rows 0 - 24 (30 total, Query took 0.0001 seconds.)

```
SELECT * FROM `ps_invoice`
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

1 > >> | Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Extra options

		in_id	in_date	amount	ren_id
<input type="checkbox"/>	Edit Copy Delete	1	2025-05-01	120.00	1
<input type="checkbox"/>	Edit Copy Delete	2	2025-05-02	110.00	2
<input type="checkbox"/>	Edit Copy Delete	3	2025-05-03	115.00	3
<input type="checkbox"/>	Edit Copy Delete	4	2025-05-04	120.00	4
<input type="checkbox"/>	Edit Copy Delete	5	2025-05-05	125.00	5
<input type="checkbox"/>	Edit Copy Delete	6	2025-05-06	130.00	6
<input type="checkbox"/>	Edit Copy Delete	7	2025-05-07	135.00	7
<input type="checkbox"/>	Edit Copy Delete	8	2025-05-08	140.00	8
<input type="checkbox"/>	Edit Copy Delete	9	2025-05-09	145.00	9
<input type="checkbox"/>	Edit Copy Delete	10	2025-05-10	150.00	10
<input type="checkbox"/>	Edit Copy Delete	11	2025-05-11	155.00	11
<input type="checkbox"/>	Edit Copy Delete	12	2025-05-12	160.00	12
<input type="checkbox"/>	Edit Copy Delete	13	2025-05-13	165.00	13
<input type="checkbox"/>	Edit Copy Delete	14	2025-05-14	170.00	14
<input type="checkbox"/>	Edit Copy Delete	15	2025-05-15	175.00	15
<input type="checkbox"/>	Edit Copy Delete	16	2025-05-16	180.00	16
<input type="checkbox"/>	Edit Copy Delete	17	2025-05-17	185.00	17
<input type="checkbox"/>	Edit Copy Delete	18	2025-05-18	190.00	18
<input type="checkbox"/>	Edit Copy Delete	19	2025-05-19	195.00	19
<input type="checkbox"/>	Edit Copy Delete	20	2025-05-20	200.00	20
<input type="checkbox"/>	Edit Copy Delete	21	2025-05-21	205.00	21
<input type="checkbox"/>	Edit Copy Delete	22	2025-05-22	210.00	22
<input type="checkbox"/>	Edit Copy Delete	23	2025-05-23	215.00	23
<input checked="" type="checkbox"/>	Console Copy Delete	24	2025-05-24	220.00	24

Payments:

Showing rows 0 - 24 (30 total, Query took 0.0001 seconds.)

```
SELECT * FROM `ps_payment`
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

	1	>	>>	<input type="checkbox"/> Show all	Number of rows:	25	Filter rows:	Search this table	9
Extra options									
	←	→	▼	pay_id	pdate	met	cardn	p_amount	in_id
<input type="checkbox"/>				1	2025-05-01	Card	1111-2222-3333-4444	120.00	1
<input type="checkbox"/>				2	2025-05-02	Card	1111-2222-3333-1002	110.00	2
<input type="checkbox"/>				3	2025-05-03	Card	1111-2222-3333-1003	115.00	3
<input type="checkbox"/>				4	2025-05-04	Card	1111-2222-3333-1004	120.00	4
<input type="checkbox"/>				5	2025-05-05	Card	1111-2222-3333-1005	125.00	5
<input type="checkbox"/>				6	2025-05-06	Card	1111-2222-3333-1006	130.00	6
<input type="checkbox"/>				7	2025-05-07	Card	1111-2222-3333-1007	135.00	7
<input type="checkbox"/>				8	2025-05-08	Card	1111-2222-3333-1008	140.00	8
<input type="checkbox"/>				9	2025-05-09	Card	1111-2222-3333-1009	145.00	9
<input type="checkbox"/>				10	2025-05-10	Card	1111-2222-3333-1010	150.00	10
<input type="checkbox"/>				11	2025-05-11	Card	1111-2222-3333-1011	155.00	11
<input type="checkbox"/>				12	2025-05-12	Card	1111-2222-3333-1012	160.00	12
<input type="checkbox"/>				13	2025-05-13	Card	1111-2222-3333-1013	165.00	13
<input type="checkbox"/>				14	2025-05-14	Card	1111-2222-3333-1014	170.00	14
<input type="checkbox"/>				15	2025-05-15	Card	1111-2222-3333-1015	175.00	15
<input type="checkbox"/>				16	2025-05-16	Card	1111-2222-3333-1016	180.00	16
<input type="checkbox"/>				17	2025-05-17	Card	1111-2222-3333-1017	185.00	17
<input type="checkbox"/>				18	2025-05-18	Card	1111-2222-3333-1018	190.00	18
<input type="checkbox"/>				19	2025-05-19	Card	1111-2222-3333-1019	195.00	19
<input type="checkbox"/>				20	2025-05-20	Card	1111-2222-3333-1020	200.00	20
<input type="checkbox"/>				21	2025-05-21	Card	1111-2222-3333-1021	205.00	21
<input type="checkbox"/>				22	2025-05-22	Card	1111-2222-3333-1022	210.00	22
<input type="checkbox"/>				23	2025-05-23	Card	1111-2222-3333-1023	215.00	23
<input checked="" type="checkbox"/>				24	2025-05-24	Card	1111-2222-3333-1024	220.00	24

Rental:

Showing rows 0 - 24 (30 total, Query took 0.0001 seconds.)

```
SELECT * FROM `ps_rental`
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

	ren_id	bdate	er_date	ar_date	rstatus	c_id	in_id
<input type="checkbox"/>	1	2025-05-01	2025-05-07	2025-05-06	Returned	1	1
<input type="checkbox"/>	2	2025-05-02	2025-05-08	2025-05-07	Returned	2	2
<input type="checkbox"/>	3	2025-05-03	2025-05-09	2025-05-08	Returned	3	3
<input type="checkbox"/>	4	2025-05-04	2025-05-10	2025-05-09	Returned	4	4
<input type="checkbox"/>	5	2025-05-05	2025-05-11	2025-05-10	Returned	5	5
<input type="checkbox"/>	6	2025-05-06	2025-05-12	2025-05-11	Returned	6	6
<input type="checkbox"/>	7	2025-05-07	2025-05-13	2025-05-12	Returned	7	7
<input type="checkbox"/>	8	2025-05-08	2025-05-14	2025-05-13	Returned	8	8
<input type="checkbox"/>	9	2025-05-09	2025-05-15	2025-05-14	Returned	9	9
<input type="checkbox"/>	10	2025-05-10	2025-05-16	2025-05-15	Returned	10	10
<input type="checkbox"/>	11	2025-05-11	2025-05-17	2025-05-16	Returned	11	11
<input type="checkbox"/>	12	2025-05-12	2025-05-18	2025-05-17	Returned	12	12
<input type="checkbox"/>	13	2025-05-13	2025-05-19	2025-05-18	Returned	13	13
<input type="checkbox"/>	14	2025-05-14	2025-05-20	2025-05-19	Returned	14	14
<input type="checkbox"/>	15	2025-05-15	2025-05-21	2025-05-20	Returned	15	15
<input type="checkbox"/>	16	2025-05-16	2025-05-22	2025-05-21	Returned	16	16
<input type="checkbox"/>	17	2025-05-17	2025-05-23	2025-05-22	Returned	17	17
<input type="checkbox"/>	18	2025-05-18	2025-05-24	2025-05-23	Returned	18	18
<input type="checkbox"/>	19	2025-05-19	2025-05-25	2025-05-24	Returned	19	19
<input type="checkbox"/>	20	2025-05-20	2025-05-26	2025-05-25	Returned	20	20
<input type="checkbox"/>	21	2025-05-21	2025-05-27	2025-05-26	Returned	21	21
<input type="checkbox"/>	22	2025-05-22	2025-05-28	2025-05-27	Returned	22	22
<input type="checkbox"/>	23	2025-05-23	2025-05-29	2025-05-28	Returned	23	23
<input checked="" type="checkbox"/>	Console it	2025-05-24	2025-05-30	2025-05-29	Returned	24	24

Reserve:

Showing rows 0 - 24 (30 total, Query took 0.0001 seconds.)

SELECT * FROM `ps_reserve`

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

	re_id	stime	etime	date	gsize	top_id
<input type="checkbox"/>	1	2025-05-01 09:00:00	2025-05-01 10:00:00	2025-05-01	2	1
<input type="checkbox"/>	2	2025-05-02 10:00:00	2025-05-02 11:00:00	2025-05-02	4	2
<input type="checkbox"/>	3	2025-05-03 11:00:00	2025-05-03 12:00:00	2025-05-03	3	3
<input type="checkbox"/>	4	2025-05-04 12:00:00	2025-05-04 13:00:00	2025-05-04	5	4
<input type="checkbox"/>	5	2025-05-05 09:30:00	2025-05-05 10:30:00	2025-05-05	6	5
<input type="checkbox"/>	6	2025-05-06 14:00:00	2025-05-06 15:00:00	2025-05-06	2	6
<input type="checkbox"/>	7	2025-05-07 15:00:00	2025-05-07 16:00:00	2025-05-07	3	7
<input type="checkbox"/>	8	2025-05-08 16:00:00	2025-05-08 17:00:00	2025-05-08	1	8
<input type="checkbox"/>	9	2025-05-09 17:00:00	2025-05-09 18:00:00	2025-05-09	8	9
<input type="checkbox"/>	10	2025-05-10 18:00:00	2025-05-10 19:00:00	2025-05-10	10	10
<input type="checkbox"/>	11	2025-05-11 09:00:00	2025-05-11 10:00:00	2025-05-11	2	11
<input type="checkbox"/>	12	2025-05-12 10:00:00	2025-05-12 11:00:00	2025-05-12	4	12
<input type="checkbox"/>	13	2025-05-13 11:00:00	2025-05-13 12:00:00	2025-05-13	3	13
<input type="checkbox"/>	14	2025-05-14 12:00:00	2025-05-14 13:00:00	2025-05-14	5	14
<input type="checkbox"/>	15	2025-05-15 09:30:00	2025-05-15 10:30:00	2025-05-15	6	15
<input type="checkbox"/>	16	2025-05-16 14:00:00	2025-05-16 15:00:00	2025-05-16	2	16
<input type="checkbox"/>	17	2025-05-17 15:00:00	2025-05-17 16:00:00	2025-05-17	3	17
<input type="checkbox"/>	18	2025-05-18 16:00:00	2025-05-18 17:00:00	2025-05-18	1	18
<input type="checkbox"/>	19	2025-05-19 17:00:00	2025-05-19 18:00:00	2025-05-19	8	19
<input type="checkbox"/>	20	2025-05-20 18:00:00	2025-05-20 19:00:00	2025-05-20	10	20
<input type="checkbox"/>	21	2025-05-21 09:00:00	2025-05-21 10:00:00	2025-05-21	2	21
<input type="checkbox"/>	22	2025-05-22 10:00:00	2025-05-22 11:00:00	2025-05-22	4	22
<input type="checkbox"/>	23	2025-05-23 11:00:00	2025-05-23 12:00:00	2025-05-23	3	23
<input checked="" type="checkbox"/>	24	2025-05-24 12:00:00	2025-05-24 13:00:00	2025-05-24	5	24

Study Room:

Showing rows 0 - 24 (30 total, Query took 0.0001 seconds.)

SELECT * FROM `ps_study_room`

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

1 > >> | Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

	room_id	capacity	re_id
<input type="checkbox"/>	101	4	1
<input type="checkbox"/>	102	6	2
<input type="checkbox"/>	103	8	3
<input type="checkbox"/>	104	6	4
<input type="checkbox"/>	105	2	5
<input type="checkbox"/>	106	10	6
<input type="checkbox"/>	107	4	7
<input type="checkbox"/>	108	6	8
<input type="checkbox"/>	109	8	9
<input type="checkbox"/>	110	6	10
<input type="checkbox"/>	111	2	11
<input type="checkbox"/>	112	10	12
<input type="checkbox"/>	113	4	13
<input type="checkbox"/>	114	6	14
<input type="checkbox"/>	115	8	15
<input type="checkbox"/>	116	6	16
<input type="checkbox"/>	117	2	17
<input type="checkbox"/>	118	10	18
<input type="checkbox"/>	119	4	19
<input type="checkbox"/>	120	6	20
<input type="checkbox"/>	121	8	21
<input type="checkbox"/>	122	6	22
<input type="checkbox"/>	123	2	23
<input type="checkbox"/>	124	10	24
<input checked="" type="checkbox"/>	125	4	25

8. Security Features Implemented:

Authentication Handling

The application has a proper login flow that ensures only valid users can access protected features. Login state is managed centrally using React Context (`AuthProvider`), which helps keep user information consistent across all components. Once a user logs in, their session details are securely stored in `localStorage`, allowing the app to remember the login state even on page refresh.

To prevent unauthorized access, routes such as dashboard, profile, etc., are guarded. If a user tries to access them without logging in, they are automatically redirected to the login page. This check is enforced globally using route guards and conditional rendering.

Authorization and Role Management

The app uses **role-based access control** to determine which users can access specific features. Each user has an associated role (like `admin` or `customer`), and based on this role, certain pages or actions are shown or hidden.

This is implemented both in the routing level (by checking roles before rendering a page) and inside components (by conditionally rendering UI elements based on permissions). Unauthorized users attempting to access restricted areas are redirected to appropriate fallback pages.

Frontend Input Validation

User inputs are handled using controlled components to ensure form integrity. Fields like login credentials and profile updates include HTML validation rules like `required`, which prevent submission of incomplete forms.

For password changes, the form includes fields for the new password and a confirmation field. Before submission, the app checks that both fields match to avoid mistakes. This helps in maintaining a basic level of client-side validation before the data is sent to the server.

This approach ensures that only authorized users can access and interact with the app securely. While the frontend checks provide a strong user experience, they also lay a foundation for good security practices, which can be further enhanced on the backend.

9. SQL Queries:

Q1) Join with at least 3 tables

A1) SQL Query:

```
SELECT C.C_ID AS Client_ID, C.C_NAME AS Client_Name, B.BNAME  
AS Book_Title, R.R_DATE AS Rental_Date  
FROM PL_CLIENT C  
JOIN PL_RENTAL R ON C.C_ID = R.C_ID  
JOIN PL_BOOK B ON R.B_ID = B.B_ID  
ORDER BY R.R_DATE DESC;
```

A2) Result:

List of clients with the books they rented and the corresponding rental date.

The screenshot shows the LibraryPlus Admin Panel interface. The left sidebar has a dark blue header "LibraryPlus" and a "Admin Panel" section with links: Dashboard, Books, Study Rooms, Events, Clients, **Rentals** (which is selected and highlighted in blue), Invoices, and Profile. The main content area is titled "Book Rentals" and contains a search bar "Search rentals...". Below it is a table with columns: ID, Book, Client, Borrow Date, Due Date, Return Date, Status, and Actions. The table lists 10 rental records. Each record includes a small book icon next to the book title. The "Status" column for all entries shows "Returned". The "Actions" column contains a "Returned" button and an "Edit" link for each row.

ID	Book	Client	Borrow Date	Due Date	Return Date	Status	Actions
1	Introduction to React	Sneha Patel	2025-05-01	2025-05-07	2025-05-06	Returned	Edit
2	Database Systems	Arjun Verma	2025-05-02	2025-05-08	2025-05-07	Returned	Edit
3	JavaScript Fundamentals	Divya Iyer	2025-05-03	2025-05-09	2025-05-08	Returned	Edit
4	Modern Web Development	Karthik Nair	2025-05-04	2025-05-10	2025-05-09	Returned	Edit
5	Python for Data Science	Riya Sharma	2025-05-05	2025-05-11	2025-05-10	Returned	Edit
6	Artificial Intelligence Basics	Ajay Pillai	2025-05-06	2025-05-12	2025-05-11	Returned	Edit
7	Machine Learning Applications	Megha Bansal	2025-05-07	2025-05-13	2025-05-12	Returned	Edit
8	Cloud Computing Essentials	Tarun Joshi	2025-05-08	2025-05-14	2025-05-13	Returned	Edit
9	DevOps Principles	Latha Krishnan	2025-05-09	2025-05-15	2025-05-14	Returned	Edit
10	Mobile Application Development	Nikhil Desai	2025-05-10	2025-05-16	2025-05-15	Returned	Edit

A3) Business Purpose:

To find which books are being rented by which customers and when, for tracking usage trends.

The screenshot shows a user profile for 'Aishwarya Mehta'. On the left, there's a circular profile picture placeholder. To the right, the name 'Aishwarya Mehta' is displayed in bold. Below the name, there are four data points: an envelope icon followed by 'aish1@mail.com', a phone icon followed by '9876543201', an ID card icon followed by 'ID: PAN12345A', and a calendar icon followed by 'Member since December 31, 2023'. At the bottom of the profile section, there are three tabs: 'Book Rentals' (which is selected and highlighted in blue), 'Room Bookings', and 'Payments'. To the right of the profile, a light blue box titled 'Activity Summary' contains three items: 'Books borrowed: 1', 'Room bookings: 1', and 'Total payments: \$120.00'. Below the activity summary, there's a card for a book rental. The card has 'Book ID: book1' at the top. Underneath, it says 'Borrowed: April 30, 2025' and 'Returned: May 5, 2025'. To the right of the return date, the word 'PAID' is written in green capital letters. Below the card, the text 'Due: May 6, 2025' is visible.

Q2) Multi-row subquery

A1) SQL Query:

```
SELECT C_ID, C_NAME
FROM PL_CLIENT
WHERE C_ID IN (
    SELECT C_ID FROM PL_RENTAL WHERE R_STATUS = 'Pending'
);
```

A2) Result:

All clients who currently have one or more pending rentals.



Raghav Menon

✉ raghav2@mail.com
📞 9876543202
👤 ID: PAN12346B
📅 Member since January 14, 2024

Activity Summary

Books borrowed:	1
Room bookings:	1
Total payments:	\$110.00

[Book Rentals](#) [Room Bookings](#) [Payments](#)

Room: room105 PENDING
 May 16, 2025: 14:00 - 16:00
 Purpose: Kerala History Research



Raghav Menon

✉ raghav2@mail.com
📞 9876543202
👤 ID: PAN12346B
📅 Member since January 14, 2024

Activity Summary

Books borrowed:	1
Room bookings:	1
Total payments:	\$110.00

[Book Rentals](#) [Room Bookings](#) [Payments](#)

Book ID: book2 Not Paid
 Borrowed: May 1, 2025 Due: May 7, 2025
 Expected: June 6, 2025

Q3) Correlated Subquery

A1) SQL Query:

```
SELECT C.C_ID, C.C_NAME,
       (SELECT COUNT(*) FROM PL_RENTAL R WHERE R.C_ID =
C.C_ID) AS Total_Rentals
FROM PL_CLIENT C
ORDER BY Total_Rentals DESC;
```

A2) Result:

Each client with the total number of rentals they've made.

[Full information about this client](#)

**Sneha Iyer**

✉️ sneha9@mail.com

📞 9876543209

👤 ID: PAN123531

📅 Member since April 30, 2024

Activity Summary

Books borrowed:	6
Room bookings:	7
Total payments:	\$145.00

[Book Rentals](#) [Room Bookings](#) [Payments](#)

Q4) SET Operator Query

A1) SQL Query:

```
SELECT C_ID FROM PL_RENTAL  
UNION  
SELECT C_ID FROM PL_RESERVE;
```

A2) Result:

All clients who have either rented books or reserved study rooms (no duplicates).

Study Rooms

📅 05/14/2025 ⏺

👤 Select client for booking ⏺ ⏰ 09:00

Quiet Haven

Capacity: 4 people

📅 2025-05-14

⌚ 9:00 - 17:00 (Available All Day)

Reserved by Sneha Patel Available from 11 AM

Q5) Inline View or WITH Clause

A1) SQL Query:

```
WITH ClientRentals AS (
    SELECT C_ID, COUNT(*) AS Rental_Count
    FROM PL_RENTAL
    GROUP BY C_ID
)
SELECT C.C_NAME, CR.Rental_Count
FROM ClientRentals CR
JOIN PL_CLIENT C ON C.C_ID = CR.C_ID
ORDER BY CR.Rental_Count DESC;
```

Q6) TOP-N / BOTTOM-N Query

A1) SQL Query (TOP 3 Events by Expense):

```
SELECT E.EVE_ID, E.EVE_TYPE, SUM(EX.EXPENSE) AS
Total_Expense
FROM PL_EVENT E
JOIN PL_EXP EX ON E.EVE_ID = EX.EVE_ID
GROUP BY E.EVE_ID, E.EVE_TYPE
ORDER BY Total_Expense DESC
LIMIT 3;
```

A2) Result:

Manage Event ×

View and manage event details and attendees

Event Details Edit Event

Coding for Beginners
Introductory coding workshop focused on beginners by Kavya Rao
Technology May 17, 2025 Registration fees \$ 250

Attendees List

Neeraj Kumar neeraj4@mail.com • 9876543204	Attending
Lakshmi Das lakshmi5@mail.com • 9876543205	Attending
Aishwarya Mehta aish1@mail.com • 9876543201	Attending
Raghav Menon raghav2@mail.com • 9876543202	Maybe

Close

10. Lessons Learned:

Importance of Managing Auth Flow Properly:

Implementing the login system using React Context taught me how crucial centralized state management is for authentication. It simplifies access control and makes the app more maintainable, especially when dealing with protected routes.

Role-Based Access is Not Just a Backend Concern:

Initially, I thought access control was only the backend's job. But while working on this project, I realized how important it is to enforce role-based access even in the frontend to enhance user experience and minimize exposure of sensitive features.

Storing User Data Securely in LocalStorage:

Using `localStorage` for session persistence worked well for keeping the user logged in. But I also learned about the potential risks and the importance of not storing sensitive information like passwords or tokens insecurely.

Simple Validation Can Prevent Big Issues:

Even basic form validations on the frontend (like `required` fields or password confirmation) greatly reduce user errors and unnecessary backend requests. It reminded me that small details in form handling can improve both security and usability.

Protecting Routes is a Must-Have, Not an Afterthought:

At first, I overlooked route protection, assuming users wouldn't navigate manually. But after implementing redirections for unauthenticated users, I saw how easily security holes can appear if route protection is not handled properly.

Planning User Roles Early Saves Headaches Later:

Defining roles (`admin`, `customer`) early in the development helped structure the logic clearly. It prevented confusion and made the component-level access control straightforward.

Code Readability and Separation Matters:

Separating authentication logic into a dedicated `AuthProvider` and keeping API/login handling clean made the codebase easier to work with and scale. This project reinforced the value of clean code structure, especially in React.

EXTRA CREDIT:

Graph on Dashboard:

To enhance the user experience and give more value to logged-in users, a graphical chart was added to the dashboard. This visual component provides a quick overview of relevant data, making the interface more interactive and informative.

Purpose of the Graph:

The chart is designed to summarize key metrics at a glance — such as booking trends, user activity, or event statistics — depending on the role of the user (admin or customer). For example:

- **Admins** may see data like total event bookings over time or user registrations.
- **Customers** may view their past bookings or participation stats.

Technical Implementation:

- The graph is built using a popular charting library like **Chart.js** or **Recharts**, integrated directly into the dashboard component.
- Data is fetched dynamically based on the logged-in user's role and ID.
- The graph uses line charts, bar charts, or pie charts depending on the type of data being shown.

Features Covered:

- Dynamic updates based on user role
- Responsive design for different screen sizes
- Interactive tooltips and hover effects for better readability
- Clean visual styling to match the rest of the app