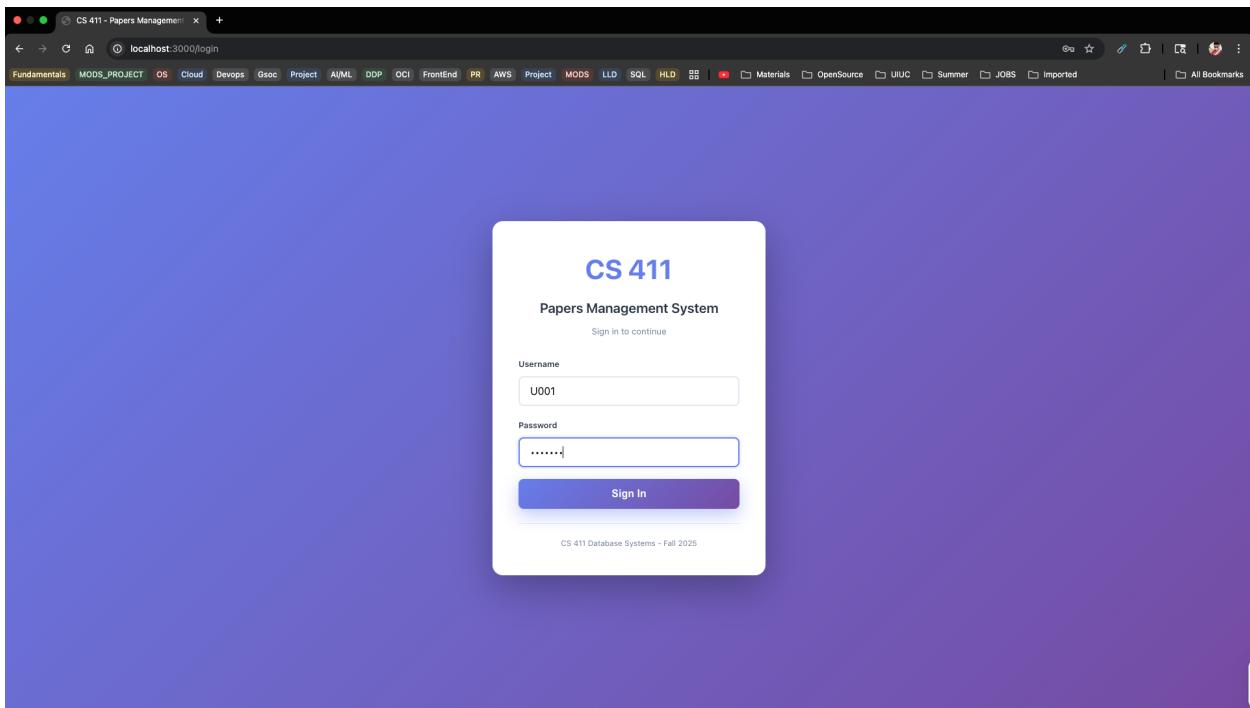
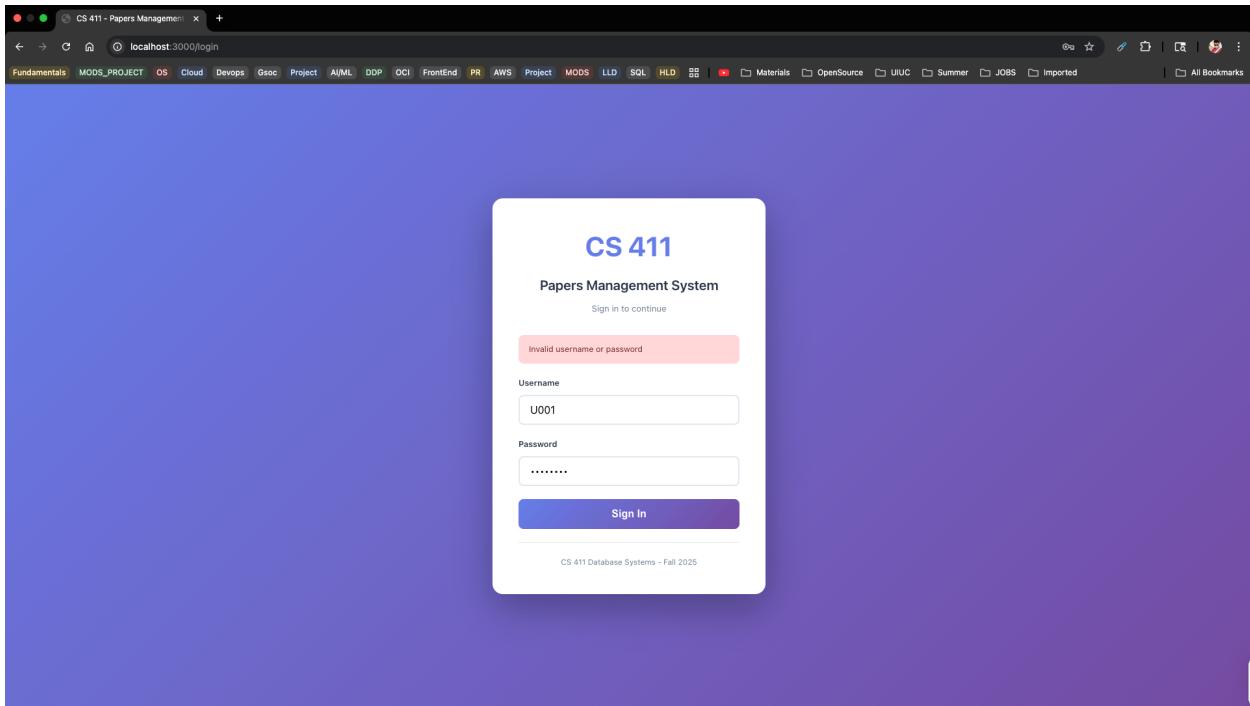


GRP-126: DEMO

Login Page:



User login credentials are validated against the database



Retrieve all papers

The screenshot shows the 'Papers Management System' dashboard. At the top, there are several tabs: 'Search Papers', 'All Papers', 'Query 1: User Papers', 'Query 2: Venues by Year', 'Query 3: Top Reviewers', and 'Query 4: User Paper Reviews'. A user 'Ada Lovelace' is logged in. Below the tabs, there is a 'Refresh Papers' button. The main area displays a grid of 11 paper cards:

- Learned Bloom Filters**: VLDB 2024, Under Review. Abstract: Reducing false positives. Date: 4/10/2025.
- Log-Structured Systems**: ICDE Workshops 2025, Under Review. Abstract: Write-optimized pipelines. Date: 4/2/2025.
- Storage Layout Showdown**: CIDR 2024, Draft. Abstract: Row vs column vs PAX. Date: 3/20/2025.
- Branchless Hash Join**: SIGMOD 2025, Draft. Abstract: Micro-architectural tweaks. Date: 3/15/2025.
- Vectorized Execution**: SIGMOD 2025, Under Review. Abstract: SIMD-aware operators. Date: 3/1/2025.
- CE on Text Data**: PVLDB 2025, Under Review. Abstract: Estimating text-centric schemas. Date: 2/25/2025.
- Adaptive Query Plans**: VLDB 2024, Draft. Abstract: Runtime re-optimization. Date: 2/10/2025.
- Page Cache Modeling**: CIDR 2024, Published. Abstract: Hot/cold separation. Date: 1/25/2025.
- Learned Indexes Revisited**: VLDB 2024, Under Review. Abstract: Comparison with B+Trees... Date: 1/3/2025.
- Join Hints Benchmark**: ICDE 2023, Published. Abstract: Pragmatic optimizer hints. Date: 1/1/2023.
- Hybrid CE with ML**: SIGMOD 2025, Published. Abstract: Rules + ML for cardinality. Date: 1/1/2025.
- Efficient Joins in Practice**: ICDE 2023, Published. Abstract: We study join orders... Date: 1/1/2023.

The screenshot shows the Network tab in the developer tools. It lists 11 requests for the 'papers' endpoint. The requests are as follows:

- [paper_id: "P012", paper_title: "Learned Bloom Filters", abstract: "Reducing false positives...", paper_url: "http://localhost:3000/paper/P012"]
- [paper_id: "P012", paper_title: "Learned Bloom Filters", abstract: "Reducing false positives...", paper_url: "http://localhost:3000/paper/P012"]
- [paper_id: "P006", paper_title: "Storage Layout Showdown", abstract: "Row vs column vs PAX...", paper_url: "http://localhost:3000/paper/P006"]
- [paper_id: "P018", paper_title: "Branchless Hash Join", abstract: "Micro-architectural tweaks...", paper_url: "http://localhost:3000/paper/P018"]
- [paper_id: "P009", paper_title: "Vectorized Execution", abstract: "SIMD-aware operators...", paper_url: "http://localhost:3000/paper/P009"]
- [paper_id: "P003", paper_title: "CE on Text Data", abstract: "Estimating text-centric schemas...", paper_url: "http://localhost:3000/paper/P003"]
- [paper_id: "P003", paper_title: "Adaptive Query Plans", abstract: "Runtime re-optimization...", paper_url: "http://localhost:3000/paper/P003"]
- [paper_id: "P011", paper_title: "Page Cache Modeling", abstract: "Hot/cold separati...", paper_url: "http://localhost:3000/paper/P011"]
- [paper_id: "P002", paper_title: "Learned Indexes Revisited", abstract: "Comparison with B+Trees...", paper_url: "http://localhost:3000/paper/P002"]
- [paper_id: "P007", paper_title: "Join Hints Benchmark", abstract: "Pragmatic optimizer hints...", paper_url: "http://localhost:3000/paper/P007"]
- [paper_id: "P004", paper_title: "Hybrid CE with ML", abstract: "Rules + ML for cardinality...", paper_url: "http://localhost:3000/paper/P004"]
- [paper_id: "P001", paper_title: "Efficient Joins in Practice", abstract: "We study join orders...", paper_url: "http://localhost:3000/paper/P001"]

At the bottom, it shows 1 requests and 211 B transferred.

Query1

Papers Management System

Welcome, Ada Lovelace [Logout](#)

Search Papers All Papers [Query 1: User Papers](#) [Query 2: Venues by Year](#) [Query 3: Top Reviewers](#) [Query 4: User-Paper Reviews](#)

Year: User ID: [Run Query](#)

Get papers authored by a user, filtered by year, sorted by upload timestamp and review count (limit 10)

PAPER ID	PAPER TITLE	UPLOAD TIMESTAMP	STATUS	REVIEW COUNT
P008	Log-Structured Systems	2025-04-02T16:11:00.000Z	Under Review	1
P006	Storage Layout Showdown	2025-03-20T18:45:00.000Z	Draft	1
P003	Adaptive Query Plans	2025-02-10T21:12:00.000Z	Draft	1

Network Performance Memory Application Privacy and security Lighthouse Recorder Components Profiler

Filter 4 ms 10 ms 20 ms 25 ms 30 ms 35 ms 40 ms 45 ms 50 ms 55 ms 60 ms 65 ms 70 ms 75 ms 80 ms 85 ms 90 ms 95 ms 100 ms 105 ms

Name [Headers](#) [Payload](#) [Preview](#) [Response](#) [Initiator](#) [Timing](#)

[query?year=2024&user_id=U005](#)

Request Headers

- Access-Control-Allow-Origin: *
- Connection: keep-alive
- Date: Fri, 14 Nov 2025 05:58:11 GMT
- Elig: WtTaB-M2SVSDnzb7XY+DvSAwESjNY+
- Keep-Alive: timeout=5
- X-Powered-By: Express

Response Headers

- Content-Type: application/json; charset=utf-8
- Date: Fri, 14 Nov 2025 05:58:11 GMT
- Keep-Alive: timeout=5
- Server: cloudflare-nginx
- Vary: Accept-Encoding
- Via: 1.1 varnish
- Warning: 110 "Google Chrome"/v=142, "Net_A_Brand"/v=199
- X-Powered-By: Express

Request

- Accept: */*
- Accept-Encoding: gzip, deflate, br, zstd
- Accept-Language: en-US,en;q=0.9
- Connection: keep-alive
- Host: localhost:4000
- Referer: http://localhost:3000/
- Sec-Ch-Ua: Chakra/142.0.6964.100, "Google Chrome/142.0.6964.100", "Not_A_Brand/100"
- Sec-Ch-Ua-Mobile: 0
- Sec-Ch-Ua-Platform: "macOS"
- Sec-Fetch-Dest: empty
- Sec-Fetch-Mode: cors

1 requests | 211 B transferred | 424 B resources

FrontEnd: Invoked from home.jsx > runQuery1

```

.gitignore          U   73
package.json        U   74
README.md          U   75
test.html          U   76
vercel.json         U   77
cs_411_frontend
  node_modules
    src
      pages
        Home.css
        Home.jsx  M   91
        Login.css
        Login.jsx
        App.css
        App.jsx
        config.js

```

```

const runQuery1 = async () => {
  setLoading(true)
  setError('')
  try {
    const response = await fetch(
      `${API_BASE_URL}/api/advanced/query1?year=${query1Year}&user_id=${query1UserId}`
    )
    if (!response.ok) throw new Error('Query failed')
    const data = await response.json()
    setQuery1Results(data)
  } catch (err) {
    setError(err.message)
  } finally {
    setLoading(false)
  }
}

```

Backend:

The screenshot shows a code editor interface with the following details:

- File Structure:** The left sidebar shows a project structure for "CS411". It includes "cs_411_backend" and "cs_411_frontend" folders. "cs_411_backend" contains "api" and "advanced" subfolders, with "query1.js" selected. "query1.js" also lists other files like "query2.js", "query3.js", "query4.js", "index.js", etc.
- Code View:** The main area displays the content of "query1.js". The code is an asynchronous function handling a GET request. It checks if the method is allowed, sets CORS headers, and then tries to execute a database query. The query selects papers from the "Papers" table, joins them with the "Reviews" table, and counts reviews. It filters by user ID and upload timestamp, ordering by upload timestamp and review count, and limiting the results to 15. Finally, it handles the response with JSON.
- Editor UI:** The bottom of the editor shows status bars for "New update available", "Later", "Install Now", and "Review next file >".

```
cs_411_backend > api > advanced > JS query1.js < vercel.json

const { getConnection } = require('../_db');

module.exports = async (req, res) => {
  if (req.method !== 'GET') {
    return res.status(405).json({ error: 'Method not allowed' });
  }

  res.setHeader('Access-Control-Allow-Origin', '*');

  try {
    const connection = await getConnection();
    const { year, user_id } = req.query;

    if (!year || !user_id) {
      await connection.end();
      return res.status(400).json({ error: 'year and user_id are required' });
    }

    const yearStart = `${year}-01-01 00:00:00`;

    const query = `SELECT
      p.paper_id,
      p.paper_title,
      p.upload_timestamp,
      p.status,
      COUNT(r.review_id) as review_count
    FROM Authorship a
    INNER JOIN Papers p ON a.paper_id = p.paper_id
    LEFT JOIN Reviews r ON p.paper_id = r.paper_id
    WHERE a.user_id = ?
      AND p.upload_timestamp >= ?
      AND p.project_id IS NOT NULL
    GROUP BY p.paper_id, p.paper_title, p.upload_timestamp, p.status
    ORDER BY p.upload_timestamp DESC, review_count DESC
    LIMIT 15
  `;

    const [rows] = await connection.execute(query, [user_id, yearStart]);
    await connection.end();

    res.status(200).json(rows);
  } catch (error) {
    console.error('Query 1 error:', error);
    res.status(500).json({
      error: 'Query failed',
      message: error.message
    });
  }
};
```

Matches the result show on the UI:

<> Untitled Query

Run selected [Save](#) [Format](#) [Clear](#)

[Gemini settings](#) [Valid](#)

```

4   p.upload_timestamp,
5   p.paper_id,
6   COUNT(r.review_id) AS review_count
7   FROM Authorship a
8   INNER JOIN Paper p
9   ON a.paper_id = p.paper_id
10  LEFT JOIN Reviews r
11  ON p.paper_id = r.paper_id
12  WHERE p.paper_id = '1005'
13  AND p.upload_timestamp >= 2024
14  AND p.project_id IS NOT NULL
15  GROUP BY
16  p.paper_id,
17  p.paper_title,
18  p.upload_timestamp,
19  p.status
20  ORDER BY
21  p.upload_timestamp DESC,
22  review_count DESC
23  LIMIT 15;

```

Results

paper_id	paper_title	upload_timestamp	status	review_count
P008	Log-Structured Systems	2025-04-02 11:11:00	Under Review	1
P006	Storage Layout Showdown	2025-03-20 13:45:00	Draft	1
P003	Adaptive Query Plans	2025-02-10 15:12:00	Draft	1

Execution time: 6.1 ms [Export](#)

Rows per page: 20 ▾ 1 - 3 of 3 |< < > >|