

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <title>Search Database</title>

  <style>

    table { width: 100%; border-collapse: collapse; }

    table, th, td { border: 1px solid black; }

    th, td { padding: 8px; text-align: left; }

    th { cursor: pointer; }

  </style>

</head>

<body>

  <form method="GET" action="">

    <input type="text" name="name" placeholder="Search by Name" value="<?php echo
    htmlspecialchars($_GET['name'] ?? '', ENT_QUOTES); ?>">

    <input type="text" name="negative" placeholder="Negative Search (e.g., -example)"
    value="<?php echo htmlspecialchars($_GET['negative'] ?? '', ENT_QUOTES); ?>">

    <input type="text" name="min" placeholder="Min (ignore spaces)" value="<?php echo
    htmlspecialchars($_GET['min'] ?? '', ENT_QUOTES); ?>">

    <input type="text" name="max" placeholder="Max (ignore spaces)" value="<?php echo
    htmlspecialchars($_GET['max'] ?? '', ENT_QUOTES); ?>">

    <select name="sort">

      <option value="">Sort by Rate</option>

      <option value="asc" <?php echo (isset($_GET['sort']) && $_GET['sort'] == 'asc') ?
      'selected' : ''; ?>>Low to High</option>
```

```

        <option value="desc" <?php echo (isset($_GET['sort']) && $_GET['sort'] == 'desc') ?
'selected' : ''; ?>>High to Low</option>

    </select>

    <input type="text" name="category" placeholder="Search by Category" value="<?php
echo htmlspecialchars($_GET['category'] ?? '', ENT_QUOTES); ?>">

    <select name="type">

        <option value="">All Results</option>

        <option value="Default" <?php echo (isset($_GET['type']) && $_GET['type'] == 'Default')
? 'selected' : ''; ?>>Default</option>

        <option value="Custom Comments" <?php echo (isset($_GET['type']) && $_GET['type']
== 'Custom Comments') ? 'selected' : ''; ?>>Custom Comments</option>

        <option value="Package" <?php echo (isset($_GET['type']) && $_GET['type'] ==
'Package') ? 'selected' : ''; ?>>Package</option>

        <option value="Subscriptions" <?php echo (isset($_GET['type']) && $_GET['type'] ==
'Subscriptions') ? 'selected' : ''; ?>>Subscriptions</option>

    </select>

    <button type="submit">Search</button>

</form>

```

```

<?php

$dsn = 'sqlite:' . __DIR__ . '/db.db';

$options = [

    PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION,

    PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC,

];

try {

    $pdo = new PDO($dsn, null, null, $options);

```

```

} catch (PDOException $e) {

    die("Could not connect to the database: " . $e->getMessage());

}


// Prepare search filters

$name = $_GET['name'] ?? "";

$negative = $_GET['negative'] ?? ""; // For negative search

$min = preg_replace('/\s+/', "", $_GET['min'] ?? ""); // Remove spaces from min

$max = preg_replace('/\s+/', "", $_GET['max'] ?? ""); // Remove spaces from max

$sort = $_GET['sort'] ?? "";

$type = $_GET['type'] ?? "";

$category = $_GET['category'] ?? ""; // For category search

$page = isset($_GET['page']) ? (int)$_GET['page'] : 1;

$results_per_page = 200;

$offset = ($page - 1) * $results_per_page;


// Build the main query

$query = "SELECT \"provider name\", \"service\", \"name\", \"type\", \"rate\", \"min\", \"max\"
FROM merged_data WHERE 1=1";

$params = [];


// Add filters to the query

if ($name) {

    $words = explode(' ', $name);

    foreach ($words as $index => $word) {

        $query .= " AND \"name\" LIKE :word$index";
    }
}

```

```

        $params[":word$index"] = '%' . $word . '%';
    }
}

// Handle negative search
if ($negative) {
    $neg_words = explode(' ', $negative);
    foreach ($neg_words as $index => $neg_word) {
        $query .= " AND \"name\" NOT LIKE :neg_word$index";
        $params[":neg_word$index"] = '%' . ltrim($neg_word, '-') . '%';
    }
}

if ($min !== '') {
    $query .= " AND \"min\" = :min";
    $params[':min'] = $min;
}

if ($max !== '') {
    $query .= " AND \"max\" = :max";
    $params[':max'] = $max;
}

if ($type) {
    $query .= " AND \"type\" = :type";
    $params[':type'] = $type;
}

```

```
}
```

```
// Add category search filter
```

```
if ($category) {
```

```
    $query .= " AND \"category\" LIKE :category";
```

```
    $params[':category'] = '%' . $category . '%';
```

```
}
```

```
if ($sort) {
```

```
    $query .= " ORDER BY \"rate\" " . ($sort === 'asc' ? 'ASC' : 'DESC');
```

```
}
```

```
$query .= " LIMIT :limit OFFSET :offset";
```

```
$params[':limit'] = $results_per_page;
```

```
$params[':offset'] = $offset;
```

```
// Execute the query
```

```
$stmt = $pdo->prepare($query);
```

```
foreach ($params as $key => &$value) {
```

```
    $stmt->bindParam($key, $value);
```

```
}
```

```
$stmt->execute();
```

```
$results = $stmt->fetchAll();
```

```
// Get total count for pagination
```

```
$total_query = "SELECT COUNT(*) FROM merged_data WHERE 1=1";
```

```
$total_params = [];
```

```
if ($name) {
```

```
    foreach ($words as $index => $word) {
```

```
        $total_query .= " AND \"name\" LIKE :total_word$index";
```

```
        $total_params[":total_word$index"] = '%' . $word . '%';
```

```
    }
```

```
}
```

```
if ($negative) {
```

```
    foreach ($neg_words as $index => $neg_word) {
```

```
        $total_query .= " AND \"name\" NOT LIKE :neg_total_word$index";
```

```
        $total_params[":neg_total_word$index"] = '%' . ltrim($neg_word, '-') . '%';
```

```
    }
```

```
}
```

```
if ($min !== "") {
```

```
    $total_query .= " AND \"min\" = :min_total";
```

```
    $total_params[':min_total'] = $min;
```

```
}
```

```
if ($max !== "") {
```

```
    $total_query .= " AND \"max\" = :max_total";
```

```
    $total_params[':max_total'] = $max;
```

```
}
```

```
if ($type) {
```

```
    $total_query .= " AND \"type\" = :type_total";
```

```
    $total_params[':type_total'] = $type;
```

```
}
```

```

if ($category) {
    $total_query .= " AND \"category\" LIKE :category_total";
    $total_params[':category_total'] = '%' . $category . '%';
}

```

```

$total_stmt = $pdo->prepare($total_query);
foreach ($total_params as $key => &$value) {
    $total_stmt->bindParam($key, $value);
}
$total_stmt->execute();
$total_results = $total_stmt->fetchColumn();
$total_pages = ceil($total_results / $results_per_page);

```

?>

```

<table>
    <thead>
        <tr>
            <th>Provider Name</th>
            <th>Service</th>
            <th>Name</th>
            <th>Type</th>
            <th>Rate</th>
            <th>Min</th>
            <th>Max</th>
        </tr>

```

```

</thead>

<tbody>
    <?php if ($results): ?>
        <?php foreach ($results as $row): ?>
            <tr>
                <td><a href="http://<?php echo htmlspecialchars($row['provider name']);
?>/services"><?php echo htmlspecialchars($row['provider name']); ?></a></td>

                <td><?php echo htmlspecialchars($row['service']); ?></td>
                <td><?php echo htmlspecialchars($row['name']); ?></td>
                <td><?php echo htmlspecialchars($row['type']); ?></td>
                <td><?php echo htmlspecialchars($row['rate']); ?></td>
                <td><?php echo htmlspecialchars($row['min']); ?></td>
                <td><?php echo htmlspecialchars($row['max']); ?></td>
            </tr>

            <?php endforeach; ?>

        <?php else: ?>
            <tr>
                <td colspan="7">No results found</td>
            </tr>

        <?php endif; ?>
    </tbody>
</table>

<div>
    <?php for ($i = 1; $i <= $total_pages; $i++): ?>

        <a href="?<?php echo http_build_query(array_merge($_GET, ['page' => $i])); ?>"><?php
echo $i; ?></a>

```

```
<?php endfor; ?>
```

```
</div>
```

```
</body>
```

```
</html>
```