

PennyCgi/penny.sql (Page 1 of 1)

```

1: DROP TABLE IF EXISTS books;
2:
3: CREATE TABLE books (isbn TEXT PRIMARY KEY, author TEXT, title TEXT);
4:
5: INSERT INTO books (isbn, author, title)
6:   VALUES ('123', 'Kernighan', 'The Practice of Programming');
7: INSERT INTO books (isbn, author, title)
8:   VALUES ('234', 'Kernighan', 'The C Programming Language');
9: INSERT INTO books (isbn, author, title)
10:  VALUES ('345', 'Sedgewick', 'Algorithms in C');

```

PennyCgi/index.html (Page 1 of 1)

```

1: <!DOCTYPE html>
2:
3: <!-- ===== -->
4: <!-- index.html -->
5: <!-- Author: Bob Dondero -->
6: <!-- ===== -->
7:
8: <html>
9:
10:   <head>
11:     <title>Penny.com</title>
12:   </head>
13:
14:   <body>
15:     <hr>
16:     Welcome to <strong>Penny.com</strong>.
17:     <hr>
18:     <br>
19:     Click to <a href="/cgi-bin/searchform.py">begin</a>.<br>
20:     <br>
21:     <hr>
22:     Created by <a href="http://www.cs.princeton.edu/~rdondero">Bob
23:     Dondero</a>.
24:     <hr>
25:   </body>
26:
27: </html>

```

PennyCgi/cgi-bin/database.py (Page 1 of 1)

```

1: #!/usr/bin/env python
2:
3: #-----
4: # database.py
5: # Author: Bob Dondero
6: #-----
7:
8: import sqlite3
9: import contextlib
10:
11: #-----
12:
13: _DATABASE_URL = 'file:penny.sqlite?mode=ro'
14:
15: #-----
16:
17: def get_books(author):
18:     books = []
19:
20:     with sqlite3.connect(_DATABASE_URL, isolation_level=None,
21:         uri=True) as connection:
22:         with contextlib.closing(connection.cursor()) as cursor:
23:
24:             query_str = "SELECT isbn, author, title FROM books "
25:             query_str += "WHERE author LIKE ?"
26:             cursor.execute(query_str, [author+'%'])
27:
28:             table = cursor.fetchall()
29:             for row in table:
30:                 book = {'isbn': row[0], 'author': row[1],
31:                     'title': row[2]}
32:                 books.append(book)
33:
34:     return books
35:
36: #-----
37:
38: #-----
39:
40: def _test():
41:     books = get_books('ker')
42:     for book in books:
43:         print(book['isbn'])
44:         print(book['author'])
45:         print(book['title'])
46:         print()
47:
48:
49: if __name__ == '__main__':
50:     _test()

```

PennyCgi/cgi-bin/common.py (Page 1 of 1)

```

1: #!/usr/bin/env python
2:
3: #-----
4: # common.py
5: # Author: Bob Dondero
6: #-----
7:
8: import time
9:
10: #-----
11:
12: def print_header():
13:     if time.strftime('%p') == "AM":
14:         ampm = 'morning'
15:     else:
16:         ampm = 'afternoon'
17:     print('<hr>')
18:     print('Good ')
19:     print(ampm)
20:     print(' and welcome to <strong>Penny.com</strong>')
21:     print('<hr>')
22:
23: #-----
24:
25: def print_footer():
26:     print('<hr>')
27:     print('Today is ')
28:     print(time.asctime(time.localtime()))
29:     print('.<br>')
30:     print('Created by ')
31:     print(' <a href="https://www.cs.princeton.edu/~rdondero">')
32:     print('Bob Dondero</a>')
33:     print('<hr>')
34:
35: #-----
36:
37: # For testing:
38:
39: def _test():
40:     print_header()
41:     print()
42:     print()
43:     print_footer()
44:
45: if __name__ == '__main__':
46:     _test()

```

PennyCgi/cgi-bin/searchform.py (Page 1 of 1)

```

1: #!/usr/bin/env python
2:
3: #-----
4: # searchform.py
5: # Author: Bob Dondero
6: #-----
7:
8: import common
9:
10: #-----
11:
12: def main():
13:
14:     # Print HTTP headers.
15:
16:     print('Content-type: text/html; charset=utf-8')
17:     print()
18:
19:     # Print content.
20:
21:     print('<!DOCTYPE html>')
22:     print('<html>')
23:     print('<head>')
24:     print('<title>Penny.com</title>')
25:     print('</head>')
26:     print('<body>')
27:     common.print_header()
28:     print('<h1>Author Search</h1>')
29:     print('<form action="/cgi-bin/searchresults.py" method="get">')
30:     print('Please enter an author name:')
31:     print('<input type="text" name="author" autofocus>')
32:     print('<input type="submit" value="Go">')
33:     print('</form>')
34:     print('<br>')
35:     print('<br>')
36:     common.print_footer()
37:     print('</body>')
38:     print('</html>')
39:
40: #-----
41:
42: if __name__ == '__main__':
43:     main()

```

PennyCgi/cgi-bin/searchresults.py (Page 1 of 1)

```

1: #!/usr/bin/env python
2:
3: #-----
4: # searchresults.py
5: # Author: Bob Dondero
6: #-----
7:
8: import os
9: import html # html.escape() is used to thwart CSS attacks
10: import parseargs
11: import common
12: import database
13:
14: #-----
15:
16: def main():
17:
18:     args_str = os.environ.get('QUERY_STRING', '')
19:     args = parseargs.parse(args_str)
20:     author = args.get('author', '')
21:
22:     author = author.strip()
23:
24:     if author == '':
25:         author = '(None)'
26:         books = []
27:     else:
28:         books = database.get_books(author) # Exception handling omitted
29:
30:     # Print HTTP headers.
31:     print('Content-type: text/html; charset=utf-8')
32:     print()
33:
34:     # Print content.
35:     print('<!DOCTYPE html>')
36:     print('<html>')
37:     print('<head>')
38:     print('<title>Penny.com</title>')
39:     print('</head>')
40:     print('<body>')
41:     common.print_header()
42:     print('<h1>Author Search Results</h1>')
43:     print('<h2>Books by ' + html.escape(author) + ' :</h2>')
44:     if len(books) == 0:
45:         print('<None><br>')
46:     else:
47:         pattern = '%s: <strong>%s</strong>: %s<br>'
48:         for book in books:
49:             print(pattern %
50:                   (book['isbn'], book['author'], book['title']))
51:         print('<br>')
52:         print('<br>')
53:         print('Click here to do another ')
54:         print('<a href="/cgi-bin/searchform.py">author search</a>.')
55:         print('<br>')
56:         print('<br>')
57:         print('<br>')
58:         print('<br>')
59:         common.print_footer()
60:         print('</body>')
61:         print('</html>')
62:
63: #-----
64: if __name__ == '__main__':
65:     main()

```

PennyCgiState/cgi-bin/searchform.py (Page 1 of 1)

```

1: #!/usr/bin/env python
2:
3: #-----
4: # searchform.py
5: # Author: Bob Dondero
6: #-----
7:
8: import os
9: import html
10: import http.cookies
11: import common
12:
13: def main():
14:
15:     prev_author = ' (None)'
16:     cookie = http.cookies.SimpleCookie(os.environ.get('HTTP_COOKIE'))
17:     prev_author_morsel = cookie.get('prev_author')
18:     if prev_author_morsel is not None:
19:         prev_author = prev_author_morsel.value
20:
21:     # Print HTTP headers.
22:
23:     print('Content-type: text/html; charset=utf-8')
24:     print()
25:
26:     # Print content.
27:
28:     print('<!DOCTYPE html>')
29:     print('<html>')
30:     print('<head>')
31:     print('<title>Penny.com</title>')
32:     print('</head>')
33:     print('<body>')
34:     common.print_header()
35:     print('<h1>Author Search</h1>')
36:     print('<form action="/cgi-bin/searchresults.py" method="get">')
37:     print('Please enter an author name:')
38:     print('<input type="text" name="author" autofocus>')
39:     print('<input type="submit" value="Go">')
40:     print('</form>\n')
41:     print('<br>')
42:     print('<br>')
43:     print('<strong>Previous author search:</strong> ')
44:     # Escape to thwart XSS attacks.
45:     print(html.escape(prev_author))
46:     print('<br>')
47:     print('<br>')
48:     common.print_footer()
49:     print('</body>')
50:     print('</html>')
51:
52: #-----
53:
54: if __name__ == '__main__':
55:     main()

```

PennyCgiState/cgi-bin/searchresults.py (Page 1 of 1)

```

1: #!/usr/bin/env python
2:
3: #-----
4: # searchresults.py
5: # Author: Bob Dondero
6: #-----
7:
8: import os
9: import http.cookies
10: import html # html.escape() is used to thwart XSS attacks
11: import parseargs
12: import common
13: import database
14:
15: def main():
16:     args_str = os.environ.get('QUERY_STRING', '')
17:     args = parseargs.parse(args_str)
18:     author = args.get('author', '')
19:     author = author.strip()
20:
21:     if author == '':
22:         author = ' (None)'
23:         books = []
24:     else:
25:         books = database.get_books(author) # Exception handling omitted
26:
27:     cookie = http.cookies.SimpleCookie()
28:     cookie['prev_author'] = author
29:
30:     # Print HTTP header.
31:     print('Content-type: text/html; charset=utf-8')
32:     print(cookie.output())
33:     print()
34:
35:     # Print content.
36:     print('<!DOCTYPE html>')
37:     print('<html>')
38:     print('<head>')
39:     print('<title>Penny.com</title>')
40:     print('</head>')
41:     print('<body>')
42:     common.print_header()
43:     print('<h1>Author Search Results</h1>')
44:     print('<h2>Books by ' + html.escape(author) + ' :</h2>')
45:     if len(books) == 0:
46:         print(' (None)<br>')
47:     else:
48:         pattern = '%s: <strong>%s</strong>: %s<br>'
49:         for book in books:
50:             print(pattern %
51:                   (book['isbn'], book['author'], book['title']))
52:         print('<br>')
53:         print('<br>')
54:         print('Click here to do another ')
55:         print('<a href="/cgi-bin/searchform.py">author search</a>.')
56:         print('<br>')
57:         print('<br>')
58:         print('<br>')
59:         print('<br>')
60:         common.print_footer()
61:         print('</body>')
62:         print('</html>')
63:
64: #-----
65: if __name__ == '__main__':
66:     main()

```