

```
mongo --port 27017
an upcoming release.
We recommend you begin using "mongosh".
For installation instructions, see
https://docs.mongodb.com/mongodb-shell/install/

-----
The server generated these startup warnings when booting:
  2022-04-24T14:45:42.676+01:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
  2022-04-24T14:45:42.677+01:00: This server is bound to localhost. Remote systems will be unable to connect to this server. Start the server with --bind_ip <address> to specify which IP addresses it should serve responses from, or with --bind_ip_all to bind to all interfaces. If this behavior is desired, start the server with --bind_ip 127.0.0.1 to disable this warning
  2022-04-24T14:45:42.677+01:00: Soft limits for open file descriptors too low
  2022-04-24T14:45:42.677+01:00:           currentValue: 10240
  2022-04-24T14:45:42.677+01:00:           recommendedMinimum: 64000
-----

Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enablefreeMonitoring()
To permanently disable this reminder, run the following command: db.disablefreeMonitoring()

-----
>
```

그림 7-2 인터랙티브 몽고DB 세션

인터랙티브 세션에서 다음 명령을 연속적으로 실행해서 데이터베이스에 저장된 모든 사용자를 추출해보자.

```
> use planner
> db.users.find({})
```

추출 결과는 [그림 7-3]과 같다. 모든 사용자 정보가 반환되었으며 사용자 패스워드에는 해싱된 값이 저장되어 있다.

```
mongo --port 27017
> use planner
switched to db planner
> db.users.find({})
[ { "_id" : ObjectId("62655d4b52b6386b8b1b5fb"), "email" : "reader@packt.com", "password" : "$2b$12$Jcc5VXty397UDGeg3bdq8encodqNvi
f8ngVj06P1IU1NFIjONGP/m", "events" : [ ] } ]
>
```

그림 7-3 쿼리를 사용해 모든 사용자 정보를 추출한 결과

지금까지 패스워드를 안전하게 저장하는 컴포넌트를 성공적으로 구현했다. 이어서 JWT를 생성하고 검증하는 컴포넌트를 만들어보자.