

**Team Name:** APCS (Apple Peeler Corer Slicer)

**Members:**

NAMES

- 1) Matthew Skogen
- 2) Ryan Davis
- 3) Asher Farr
- 4) Chad Di Lauro
- 5) John Henry Fitzgerald
- 6) Kamiar Coffey

GITHUB USERNAME

- <https://github.com/mskogen>  
<https://github.com/ryan-s-davis>  
<https://github.com/BillFlan>  
<https://github.com/chaddilauro>  
<https://github.com/jhff24>  
<https://github.com/kamiarcoffey>

## User Acceptance Test Plans

### 1. Registration Functionality

- Test that a new user can be registered and that the correct information is populated in the database
  - Name: TestUser
  - Email: [Test.Email@Test.com](mailto:Test.Email@Test.com)
  - Verify: Name and Email stored inside of **users** database

### 2. Login Functionality

- Test that a user can log in after an account had already been registered for them
  - Login using the registered user from test case 1
  - Verify: The user is logged back in based on the generated cookie in the **sessions** database

### 3. User Preferences Stored

- Test that user preferences were stored in the **users** database
  - Pass Held: Ikon Pass
  - Snow Surface Type: Corn
  - Arrival Time: No Rush
  - Preferred Weather: Cold
  - Verify: pass = "Ikon"
  - Verify: preferred\_snowType: "Corn"
  - Verify: preferred\_travelTime: "No Rush"
  - Verify: preferred\_temperature: "Cold"

## Automated Test Cases

### 1) Testing Registration:

**Register**

Email address

Test.Email@test.com

Username

TestEmail

Password

.....

Confirm Password

.....|

**Password must contain the following**

- ✓ at least one **lowercase** letter
- ✓ at least one **capital** letter
- ✓ at least one **symbol**
- ✓ at least one **number**
- ✓ at least one **8 characters**
- ✓ passwords **match**

**Register**

## Users Database with populated information for new user:

The screenshot shows the MongoDB Compass interface for a local database named 'BlizzardKickerDev.users'. The left sidebar displays the database structure with collections: registrations, resorts, sessions, and users. The main panel shows the 'users' collection with two documents. The first document is a JSON object with fields: \_id, resort\_id\_list, email, username, password, \_\_v, preferred\_snowType, preferred\_temperature, preferred\_travelTime, pass, and resorts. The second document is a similar JSON object with different values. The interface includes a filter bar, an 'INSERT DOCUMENT' button, and a 'VIEW' toggle set to 'LIST'.

MongoDB Compass - localhost:27017/BlizzardKickerDev.users

Connect View Collection Help

My Cluster

localhost:27017 STANDALONE

MongoDB 4.0.4 Community

BlizzardKickerDev.users

DOCUMENTS 2 TOTAL SIZE 632B AVG. SIZE 316B INDEXES 3 TOTAL SIZE 104.0KB AVG. SIZE 34.7KB

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER

INSERT DOCUMENT VIEW LIST TABLE

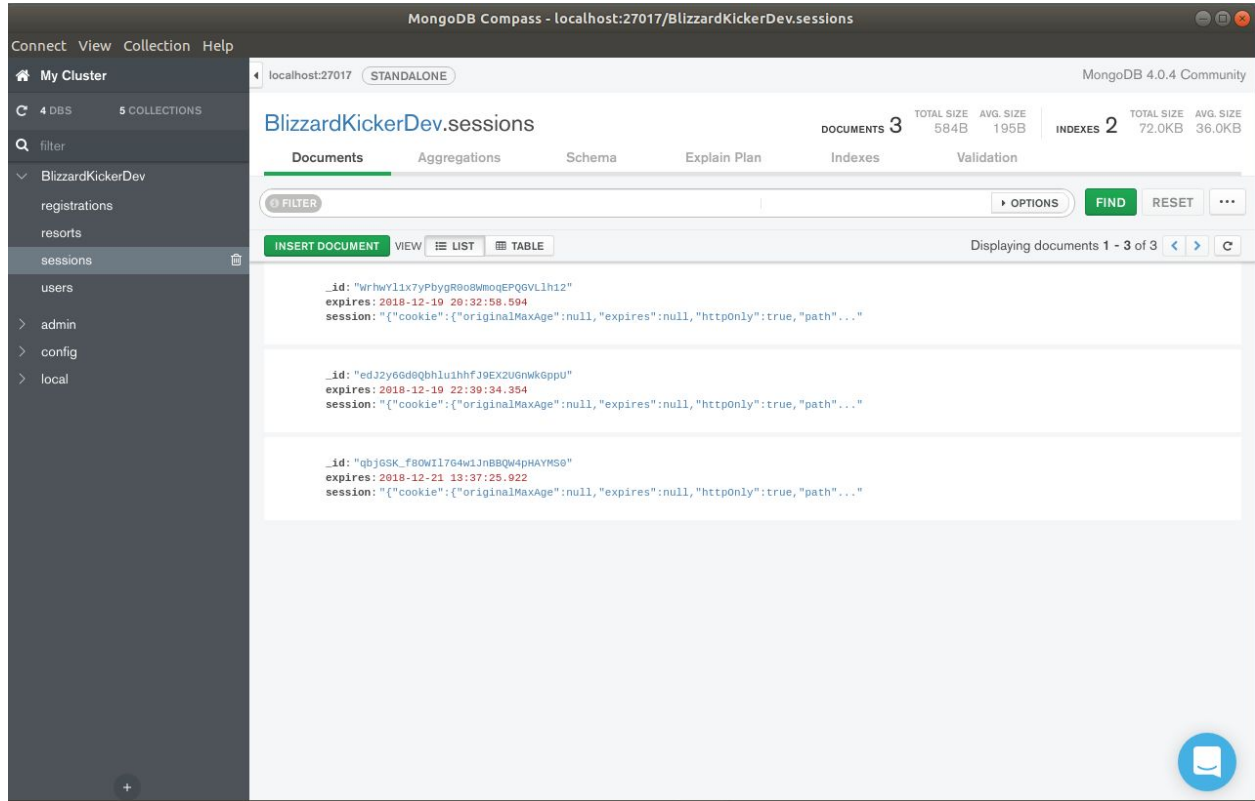
Displaying documents 1 - 2 of 2

```
{
  "_id": ObjectId("5c088457dc826011332dbc95"),
  "resort_id_list": Array,
  "email": "test_email@test.com",
  "username": "testyMcTestFace",
  "password": "$2a$10$8sQKGFNU0zLUcuIdKuU10Vc1J0ohM4YLv40wyzahuV296/rr/j0y",
  "__v": 0,
  "preferred_snowType": "Corn",
  "preferred_temperature": "Cold",
  "preferred_travelTime": "No Rush",
  "pass": Array,
  "resorts": Array
}
```

```
{
  "_id": ObjectId("5cedf9d04a87a12ecceda64b"),
  "resorts": Array,
  "pass": Array,
  "email": "Test.Email@test.com",
  "username": "TestEmail",
  "password": "$2a$10$ti/q.yqbI7auR3GN5BAyweIdQlHl3SEaCuzwsqmLcVPbIbm4CO.q",
  "__v": 0
}
```

## 2) Testing Login with generated cookie:

Session Database before login:



MongoDB Compass - localhost:27017/BlizzardKickerDev.sessions

My Cluster | localhost:27017 | STANDALONE | MongoDB 4.0.4 Community

4 DBS | 5 COLLECTIONS

filter

BlizzardKickerDev

- registrations
- resorts
- sessions
- users
- admin
- config
- local

BlizzardKickerDev.sessions

DOCUMENTS 3 | TOTAL SIZE 584B | AVG. SIZE 195B | INDEXES 2 | TOTAL SIZE 72.0KB | AVG. SIZE 36.0KB

Documents | Aggregations | Schema | Explain Plan | Indexes | Validation

FILTER | OPTIONS | FIND | RESET | ...

INSERT DOCUMENT | VIEW | LIST | TABLE | Displaying documents 1 - 3 of 3

```
{ "_id": "wrhwy1lx7yPbyGR808WmoeFPQGVlh12",
  "expires": "2018-12-19 20:32:58.594",
  "session": "{\"cookie\":{\"originalMaxAge\":null,\"expires\":null,\"httpOnly\":true,\"path\":\"\"}}"
```

```
{ "_id": "edJ2y6deQbhluihhfJ9EX2UGnwkGppu",
  "expires": "2018-12-19 22:39:34.354",
  "session": "{\"cookie\":{\"originalMaxAge\":null,\"expires\":null,\"httpOnly\":true,\"path\":\"\"}}"
```

```
{ "_id": "qbJ6SK_f80wI1764w1JnBBQW4pHAYMS0",
  "expires": "2018-12-21 13:37:25.922",
  "session": "{\"cookie\":{\"originalMaxAge\":null,\"expires\":null,\"httpOnly\":true,\"path\":\"\"}}"
```

Session Database after login occurs with generated cookie:

MongoDB Compass - localhost:27017/BlizzardKickerDev.sessions

Connect View Collection Help

My Cluster

localhost:27017 STANDALONE

MongoDB 4.0.4 Community

BlizzardKickerDev.sessions

DOCUMENTS 4 TOTAL SIZE 792B AVG. SIZE 198B INDEXES 2 TOTAL SIZE 72.0KB AVG. SIZE 36.0KB

Documents Aggregations Schema Explain Plan Indexes Validation

INSERT DOCUMENT VIEW LIST TABLE

Displaying documents 1 - 4 of 4

```
{ "_id": "wrhwy1lx7yPbygR8o8WmoqEPQGVl1h12",
  "expires": 2018-12-19 20:32:58.594,
  "session": {"cookie":{"originalMaxAge":null,"expires":null,"httpOnly":true,"path":...}}
}
```

```
{ "_id": "edJ2y6d8QbhluihhfJ9EX2UGnWk6ppU",
  "expires": 2018-12-19 22:39:34.354,
  "session": {"cookie":{"originalMaxAge":null,"expires":null,"httpOnly":true,"path":...}}
}
```

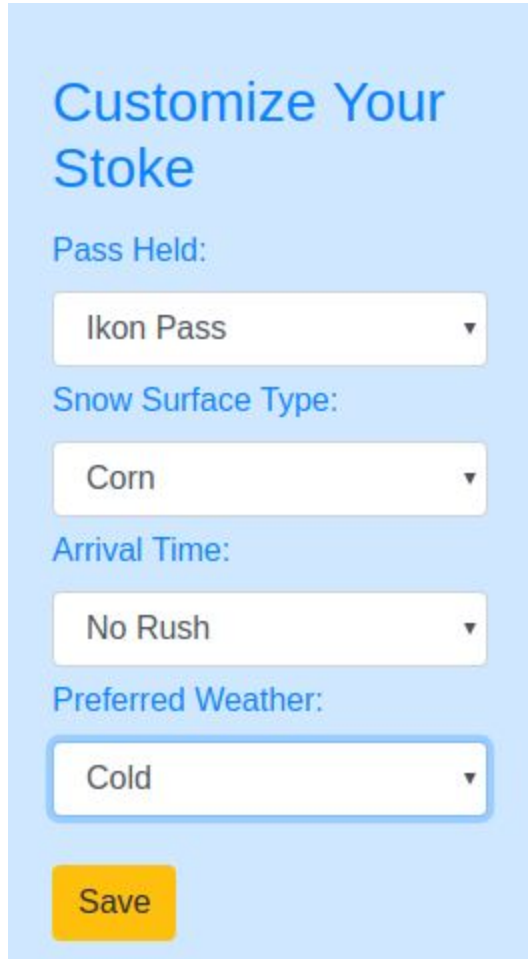
```
{ "_id": "qbJ6SK_f8OWI17G4w1JnBBQW4pHAYMS0",
  "expires": 2018-12-21 13:37:25.922,
  "session": {"cookie":{"originalMaxAge":null,"expires":null,"httpOnly":true,"path":...}}
}
```

```
1 { "_id": "cZqQrEWZVtn17AljRK1-D3zueHnWwH4",
2   "expires": 2018-12-23 22:33:22.372,
3   "session": {"cookie":{"originalMaxAge":null,"expires":null,"httpOnly":true,"path":"/"},
  "userId": "Test.Email@test.com", "userName": "TestEa"
}
```

CANCEL UPDATE

### 3) Testing updating user preferences

Preferences which are set:



The image shows a web form titled "Customize Your Stoke" on a light blue background. The form contains four dropdown menus, each with a label in blue text to its left. The first dropdown is labeled "Pass Held:" and shows "Ikon Pass". The second is labeled "Snow Surface Type:" and shows "Corn". The third is labeled "Arrival Time:" and shows "No Rush". The fourth is labeled "Preferred Weather:" and shows "Cold". At the bottom of the form is a yellow "Save" button.

Customize Your Stoke

Pass Held:

Ikon Pass ▼

Snow Surface Type:

Corn ▼

Arrival Time:

No Rush ▼

Preferred Weather:

Cold ▼

Save

User Database displaying updated user preferences being stored in the database:

MongoDB Compass - localhost:27017/BlizzardKickerDev.users

My Cluster | localhost:27017 | STANDALONE | MongoDB 4.0.4 Community

4 DBS | 5 COLLECTIONS

filter

BlizzardKickerDev

- registrations
- resorts
- sessions
- users

admin

config

local

BlizzardKickerDev.users

DOCUMENTS 2 | TOTAL SIZE 849B | AVG. SIZE 425B | INDEXES 3 | TOTAL SIZE 104.0KB | AVG. SIZE 34.7KB

Documents | Aggregations | Schema | Explain Plan | Indexes | Validation

FILTER | OPTIONS | FIND | RESET | ...

INSERT DOCUMENT | VIEW | LIST | TABLE

Displaying documents 1 - 2 of 2

```
{
  "_id": ObjectId("5c088457dc826011332dbc85"),
  "resort_id_list": Array,
  "email": "test_email@test.com",
  "username": "testyMcTestFace",
  "password": "$2a$10$8sQG6FPNU9zLUcuIdKuui0Vci39ohM4VLv40wyzahuV296/rr/j0y",
  "v": 0,
  "preferred_snowType": "Corn",
  "preferred_temperature": "Cold",
  "preferred_travelTime": "No Rush",
  "pass": Array,
  "resorts": Array
}
```

```
{
  "_id": ObjectId("5c0df9d04a97a12ecceda84b"),
  "resorts": Array,
  "pass": Array,
  "v": 0,
  "email": "Test.Email@test.com",
  "username": "TestEmail",
  "password": "$2a$10$tI/q.yqbI7aur3GnsBAyyeIdQlHEL3SEaCuzwsqmLcVPbIbm4CO.q",
  "preferred_snowType": "Corn",
  "preferred_temperature": "Cold",
  "preferred_travelTime": "No Rush"
}
```