

## CS3308 Milestone 5

Team Number: 103-4

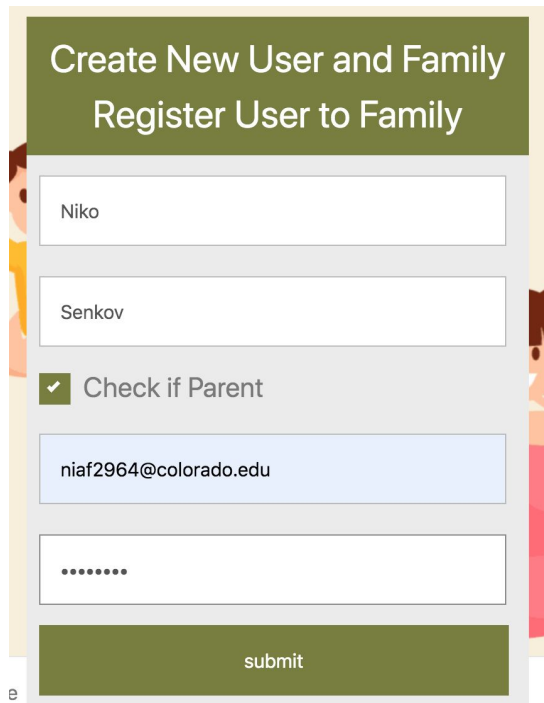
Team Members: James, Joe, Kara, Niko, Aphoto, Henry

### Test: User Registration

Test Result: **Passed**

**Functionality:** Upon registration, our server registers user information and credentials.  
If the user designates a 'parent' role, a new family is registered as well.

#### Front-end Request:



A registration form titled "Create New User and Family" and "Register User to Family". It contains input fields for first name ("Niko"), last name ("Senkov"), email ("niaf2964@colorado.edu"), and password (masked with dots). There is a checkbox labeled "Check if Parent" which is checked. A "submit" button is at the bottom.

#### Back-end Response:

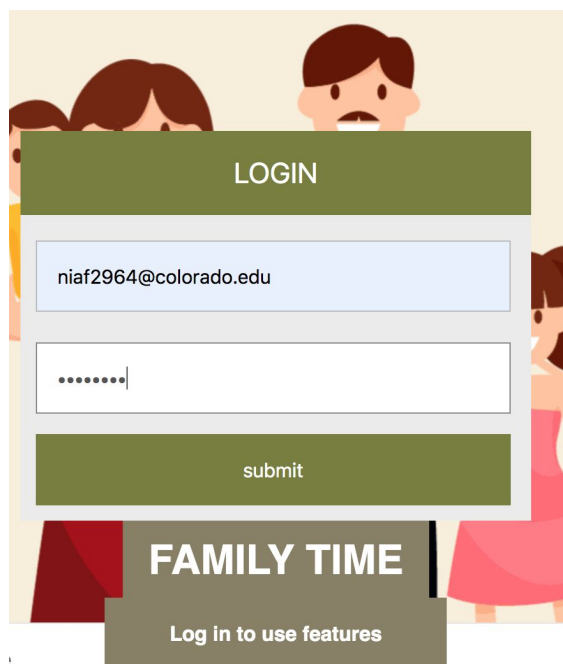
App running on port 4000  
GET /family.css 404 4.667 ms - 149  
{ first\_name: 'Niko',  
last\_name: 'Senkov',  
email: 'niaf2964@colorado.edu',  
password: 'password',  
parent: true }  
user and family created!  
POST /createFamily 200 50.264 ms - 322

### Test: User Login

Test Result: **Passed**

**Functionality:** Upon user input and submission of credentials, our server verifies the user and login request.

#### Front-end Request:



A login form titled "LOGIN" with input fields for email ("niaf2964@colorado.edu") and password (masked with dots). A "submit" button is at the bottom. Below the login form is a "FAMILY TIME" section with a "Log in to use features" button.

#### Back-end Response:

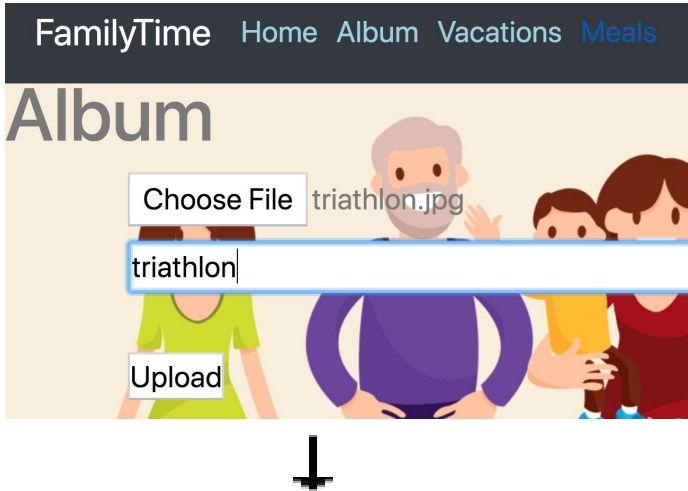
GET /family.css 404 0.428 ms - 149  
logged in  
POST /login 200 9.119 ms - 214

Test: Image Upload to Family Album

Test Result: Passed

**Functionality:** A logged in user may upload a photo to their family album. Our website allows a user to choose a file from their computer, and upload it to the site. Upon clicking the upload button, an HTTP request / response is carried out and the photo is stored in our database.

Front-end Request :



Back-end Response:

```
POST /getAllFamilyImages 200 1.983 ms - 22
family: 2
POST /getAllFamilyImages 200 9.378 ms - 22
IncomingMessage {
  ... omitting verbose HTTP request ...
}
Route { path: '/upload', stack: [ [Layer] ], methods: { post: true } }
family: 2
POST /upload 200 17.677 ms - 33
POST /getAllFamilyImages 200 1.252 ms - 22
url: http://localhost:4000/photos/2/triathlon.jpg family: 2
POST /addImageToDB 200 6.514 ms - 47
```

Front End Result of Image Upload :



```
GET /family.css 404 0.659 ms - 149
logged in
POST /login 200 9.297 ms - 214
family: 2
POST /getAllFamilyImages 200 1.434 ms - 91
GET /photos/2/triathlon.jpg 200 0.433 ms - 294675
```