Color Coded

Chris, Zach, Ethan, Catherine

Tools used









pg- promise / Express / EJS / EJS-lint



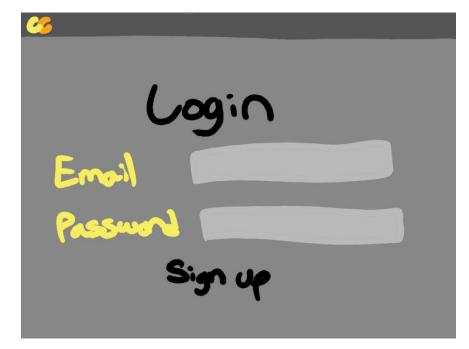


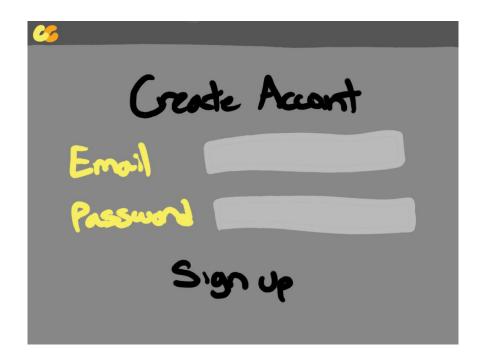




Visual Design









C:\Users\Lengl\Documents\Code\CSCI_3308\csci-3308-summer22-003\views\pages\index.ejs - Notepad++ File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ? ୭୯|୦୯|୫୧|୩୦|୭୩|୭୬୩|୭୬୦|୬୭୮||୭୯୭||୭୯୭||୭୯୭||୭୯ server js ⊠ package json ⊠ .gitignore ⊠ fun js ⊠ index.ejs

about.html

register.html

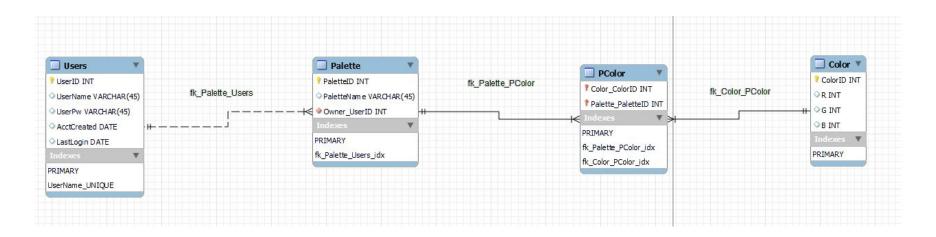
create.sql

style.css

login.html var express = require('express'); var app = express(); <div class="col" name="savedColor 2" id="circle"> var bodyParser = require('body-parser'); <div class="circle-txt" id="savedColorHex 2"></div> app.use(bodyParser.json()); app.use(bodyParser.urlencoded({ extended: true })); <div class="row" id="savedColors"> <div class="col" name="savedColor 3" id="circle"> <div class="circle-txt" id="savedColorHex 3"></div> <div class="col" name="savedColor 4" id="circle"> app.set('view engine', 'ejs'); <div class="circle-txt" id="savedColorHex 4"></div> var pgp = require('pg-promise')(); <div class="col" name="savedColor 5" id="circle"> const dbConfig = { <div class="circle-txt" id="savedColorHex 5"></div> <div class="modal-footer"> <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Close</button</pre> var db = pgp(dbConfig); <button type="submit" id="save color btn" class="btn btn-primary">Save Color</button> app.use(express.static(dirname + '/')); app.get('/views/pages/index.html', function(reg, res) { console.log('home html'); <input type="hidden" id="UserID" name="UserID"> res.render('pages/index.ejs',{ <label for="customRangel" class="form-label" class="text-center">Brightness</label> <input type="range" class="form-range" id="colorSlider" min="0%" max="100%" oninput="initColorPicker(th</pre> app.get("/", function(req, res) { console.log('home ejs');
res.render('pages/index.ejs',{ app.post('/load', function(req, res) { var id = req.body.User ID; <% if (input && input[0] && input[0].userid) { %> var query = "SELECT UserID, Colors FROM Colors WHERE UserId = '" + id + "';"; console.log('loading'); console.log('<%= input[0].userid %>'); db.any(query) .then(function (info) { <% if (input[0].colors && input[0].colors[0]) { %> console.log("sample found, updating preview");
console.log("length is:");
console.log('<%= input[0].colors.length %>'); console.log("adding color:");
console.log('savedColor ' + '<%= i %>'); document.getElementsByName('savedColor' + '<%= i %>')[0].style.backgroundColor = '<%= input[app.get('/test', function(req, res) { document.getElementById('savedColorHex ' + '<%= i %>').innerHTML = '<%= input[0].colors[i] %> return task.batch([task.any(insert), </body> length: 8,450 lines: 248 Ln:209 Col:31 Pos:7,000 Unix (LF)

er Text Markup Language file

Database Model



Challenges

- Planning do to the nature of our course being in such a short time frame, our group had problems with meeting and planning our project. But we have persevered through.
- Jira no one in our group has used Jira before, thus it was a learning experience for everyone. Much of our collaboration was done through Discord.
- Mapping coordinates of image to mouse click with resized windows.
- The brightness slider was hard because of the constant values being sent. Ultimately needed one line of code for this problem.
- Stretch goals we wanted to do "harmonies" but it was a stretch goal that we were not able to achieve within the timeframe of this course. We couldn't figure out the correct algorithm to pick out the other color values based on the users choice.