BioAuth ATM System - Implementation Code Snippets

Face ID Registration:

```
<script>
            // reference to the current media stream
            var mediaStream = null;

            // Prefer camera resolution nearest to 1280x720.
            var constraints = {
                audio: false,
                video: {
                    width: { ideal: 640 },
                    height: { ideal: 480 },
                    facingMode: "environment"
                }
            };

            async function getMediaStream(constraints) {
                try {
                    mediaStream = await
navigator.mediaDevices.getUserMedia(constraints);
                    let video = document.getElementById('cam');
                    video.srcObject = mediaStream;
                    video.onloadedmetadata = (event) => {
                        video.play();
                    };
                } catch (err) {
                    console.error(err.message);
                }
            };

            async function switchCamera(cameraMode) {
                try {
                    // stop the current video stream
                    if (mediaStream != null && mediaStream.active) {
                        var tracks = mediaStream.getVideoTracks();
                        tracks.forEach(track => {
                            track.stop();
                        })
                    }
                    // set the video source to null
                    document.getElementById('cam').srcObject = null;

                    // change "facingMode"
                    constraints.video.facingMode = cameraMode;

                    // get new media stream
                    await getMediaStream(constraints);
                } catch (err) {
                    console.error(err.message);
                    alert(err.message);
                }
            }

            function takePicture() {
                let canvas = document.getElementById('canvas');
                let video = document.getElementById('cam');
                let photo = document.getElementById('photo');
                let context = canvas.getContext('2d');

                const height = video.videoHeight;
                const width = video.videoWidth;

                if (width && height) {
                    canvas.width = width;
                    canvas.height = height;
                    context.drawImage(video, 0, 0, width, height);
                    var data = canvas.toDataURL('image/png');
                    photo.setAttribute('src', data);

                    document.getElementById("loadingOverlay").style.display =
"flex";
                    $.ajax({
                        url: "{{ url_for('auth.register_face') }}",
                        type: 'POST',
                        data: JSON.stringify({ email: registeredEmail,
imgData: data }),
                        contentType: "application/json",
                        success: function (response) {
                            console.log(response);

                            const nextStepId = "SuccessfulRegister";
                            const steps = document.querySelectorAll(".step");

                            steps.forEach(function (step) {
                                step.classList.remove("active");
                            });
                            document.getElementById(nextStepId).classList.add
("active");
                            document.getElementById("loadingOverlay").style.d
isplay = "none";

                        },
                        error: function (jqXHR, textStatus, errorThrown) {
                            console.error("Error:", textStatus, errorThrown);
                            document.getElementById("loadingOverlay").style.d
isplay = "none";
                        }
                    });

                } else {
                    clearphoto();
                }
            }
            function clearPhoto() {
                let canvas = document.getElementById('canvas');
                let photo = document.getElementById('photo');
                let context = canvas.getContext('2d');

                context.fillStyle = "#AAA";
                context.fillRect(0, 0, canvas.width, canvas.height);

                var data = canvas.toDataURL('image/png');
                photo.setAttribute('src', data);
            }

            document.getElementById('loginFaceBtn').onclick = (event) => {
                switchCamera("user");
            }

            document.getElementById('snapBtn').onclick = (event) => {
                takePicture();
                event.preventDefault();
            }

            clearPhoto();
    </script>
```

Fingerprint Registration:

We used the Digital Persona 4500 fingerprint reader.

```
 <script>
        var registeredEmail = null;
        let resultImg = document.getElementById('resultImg');

        function fromBase64Url(s) {
            return ((s.length % 4 === 2) ? s + "==" :
                (s.length % 4 === 3) ? s + "=" : s)
                .replace(/-/g, "+")
                .replace(/_/g, "/");
        }
        window.addEventListener('DOMContentLoaded', function () {
            // Check if the device has been connected
            let reader = new fpController({
                debug: true,
                version: 1
            });
```

```
      let trigger = document.getElementById('startReading');
      let result = document.getElementById('result');

      if (trigger)
          trigger.addEventListener('click', (e) => {
              reader.startReading();
          });

            // Adding event listener to capture onAcquisitionStarted
event

            reader.reader.on("SamplesAcquired", (event) => {
                console.log('Fingerprint sample acquired');

                console.log(event);
                var samples = event.samples[0];
                console.log(samples);

                sendSamplesToServer(samples);
            });


        });
        function sendSamplesToServer(samples) {
  var base64Str = "data:image/png;base64," + fromBase64Url(samples);
      let resultImg = document.getElementById('resultImg');
           console.log(base64Str);

            resultImg.src = base64Str;

            console.log(registeredEmail);
          document.getElementById("loadingOverlay").style.display =
"flex";

            $.ajax({
                type: "POST",
                url: "{{ url_for('auth.register_finger') }}",
                data: JSON.stringify({ email: registeredEmail, img:
base64Str }),
                contentType: "application/json",
                success: function (response) {
                    console.log("Success", response);

          const nextStepId = "SuccessfulRegister";
      const steps = document.querySelectorAll(".step");

                steps.forEach(function (step) {
                        step.classList.remove("active");
                    });
  document.getElementById(nextStepId).classList.add("active");
                    document.getElementById("loadingOverlay").style.dis
play = "none";
                },
  error: function (xhr, status, error) {
```

```
                  console.error("failed", xhr, status, error);
                    alert("Fingerprint registeration failed");
                          document.getElementById("loadingOverlay").style.dis
      play = "none";
                      }
                  });
              }
   </script>
```

PIN Registration:

```
<script>
        $(document).ready(function () {
            $('.tick-btn').click(async function () {
                var pin = $('#pinInput').val();
                var userId = 1;
                try {
                 document.getElementById("loadingOverlay").style.display =
"flex";
                    await new Promise((resolve, reject) => {
                        $.ajax({
                            type: "POST",
                            url: "{{ url_for('auth.register_pin') }}",
                            contentType: "application/json",
                            data: JSON.stringify({ pin: pin, user_id: userId,
email : registeredEmail }),
                            success: function (response) {
                                console.log(response.message);
                                resolve(response);
                            },
                 error: function (xhr, status, error) {
      console.error("Registration failed", xhr.responseText);
                                alert(JSON.parse(xhr.responseText).error);
                                window.location.reload();
                                reject(new Error("Registration failed"));
                            }
                        });
                    });
                  document.getElementById("loadingOverlay").style.display =
"none";
                    console.log("This code runs after a successful AJAX
call.");
                } catch (error) {
                    console.error("An error occurred:", error.message);
                }
            });

        });
    </script>
```