

Useampi blogi yhdelle käyttäjälle

Projektin kuvaus

Tällä hetkellä Tarinan Paikassa käyttäjä voi luoda yhden blogin per käyttäjätunnus. Tarkoitus on nostaa tämä raja viiteen per käyttäjätunnus. Samalla luodaan käyttäjän omalle sivulle valikko josta voi valita blogin jota käyttää (muokkaus yms.).

Blogirajoituksen poisto

Poistetaan ensin dashboard.php -tiedostosta blogien olemassaolo -tarkistus checkBlogExists -funktio.

Tiedosto dashboard.php päivitys.

Haetaan käyttäjän tiedot

```
$user_id = $_SESSION['user_id'] ?? null;

if (!$user_id) {
    echo "Virhe: käyttäjää ei tunnistettu.";
    exit;
}
```

Haetaan käyttäjän blogit

```
$stmt = $conn->prepare("SELECT blog_id, name, slug, description, created_at
                        FROM BLOG
                        WHERE user_id = ? AND deleted_at IS NULL
                        ORDER BY created_at DESC");

$stmt->bind_param("i", $user_id);
$stmt->execute();
$result = $stmt->get_result();
$userBlogs = $result->fetch_all(MYSQLI_ASSOC);
$stmt->close();

$blogCount = count($userBlogs);
$canCreateMore = $blogCount < 5;
```

Uusi blogi-painike

```
<?php if ($canCreateMore): ?>
  <a href="" alt="omat tiedot"
    hx-get="modals/create-blog-modal.php?modal_key=<?php echo
$_SESSION['modal_key']; ?>"
    hx-target="#modal-container"
    hx-trigger="click"
  >
    <section>
      
      <h3>Luo blogi!</h3>
    </section>
  </a>
<?php else: ?>
  <section class="disabled">
    
    <h3>Maksimi (5) blogia luotu</h3>
  </section>
<?php endif; ?>
```

Varmistetaan, onko käyttäjällä blogeja

```
<?php if ($blogCount > 0): ?> <!-- Jos blogeja on olemassa -->
<?php else: ?>
  <p>Et ole vielä luonut yhtään blogia. Aloita luomalla ensimmäinen
blogisi!</p>
<?php endif; ?>
```

Näytetään blogikortti ja sen toiminnot

```
<?php foreach ($userBlogs as $blog): ?>
<a href="blogit/<?= htmlspecialchars($blog['slug']); ?>" target="_blank">
  <button class="btn-view">Näytä
</a>

<button class="btn-article"
  hx-get="modals/create-article-modal.php?blog_id=<?= $blog['blog_id'];
?>&modal_key=<?= $_SESSION['modal_key']; ?>"
  hx-target="#modal-container"
  hx-trigger="click">
```

```

        Uusi julkaisu
    </button>

    <button class="btn-edit"

    hx-get="modals/edit-blog-modal.php?blog_id=<?= $blog['blog_id'];
    ?>&modal_key=<?= $_SESSION['modal_key']; ?>"
    hx-target="#modal-container"
    hx-trigger="click">
        Muokkaa blogia
    </button>

    <button class="btn-manage"

    hx-get="modals/edit-article-modal.php?blog_id=<?= $blog['blog_id'];
    ?>&modal_key=<?= $_SESSION['modal_key']; ?>"
    hx-target="#modal-container"
    hx-trigger="click">
        Muokkaa julkaisuja
    </button>

<?php endforeach; ?>

```

Blogikortin toiminnot

Näytä blogi:

Toimii entisenlaisena: linkki vie blogi-sivulle slugin perusteella.

Uusi julkaisu:

Painike vie create-article-modal.php -tiedostoon jossa blog_id:n perusteella lisätään julkaisu oikeaan blogiin.

Muokkaa blogia:

Painike vie edit-blog-modal.php -tiedostoon jossa blog_id:n perusteella lisätään julkaisu oikeaan blogiin.

Muokkaa julkaisuja:

Painike vie edit-article-modal.php -tiedostoon jossa blog_id:n perusteella lisätään julkaisu oikeaan blogiin.

Muutokset modal -tiedostoissa:

Tiedosto create-article-modal.php

Get blog_id

```
$blog_id = isset($_GET['blog_id']) ? (int) $_GET['blog_id'] : null;

if (!$blog_id) {
    http_response_code(400);
    echo "Virhe: blog_id puuttuu.";
    exit;
}
```

Haetaan sen blogin tiedot jonne uusi artikkeli halutaan lisätä

```
$blog = getBlogByIdAndUserId($conn, $blog_id, $user_id);
function getBlogByIdAndUserId(mysqli $conn, int $blog_id, int $user_id):
?array {
    $sql = "SELECT blog_id, name, slug, description, published, visibility,
views,
                created_at, updated_at, layout_id, style_id
FROM BLOG
WHERE blog_id = ? AND user_id = ? AND deleted_at IS NULL
LIMIT 1";

    if ($stmt = $conn->prepare($sql)) {
        $stmt->bind_param("ii", $blog_id, $user_id);
        $stmt->execute();
        $result = $stmt->get_result();
        $blog = $result->fetch_assoc();
        $stmt->close();
        return $blog ?: null;
    } else {
        error_log("Tietokantavirhe getBlogByIdAndUserId(): " . $conn->error);
        return null;
    }
}
```

Lisätään otsikkoon blogin nimi

```
<h3>Luo julkaisu blogiin <?= htmlspecialchars($blog['name']); ?></h3>
```

Varmistetaan että create-article-vf.php saa oikean blog_id:n

```
<input type="hidden" name="blog_id" value="<?= (int)$blog['blog_id']; ?>">
```

Tiedosto edit-blog-modal.php

Get blog_id

```
$blog_id = isset($_GET['blog_id']) ? (int) $_GET['blog_id'] : null;

if (!$blog_id) {
    http_response_code(400);
    echo "Virhe: blog_id puuttuu.";
    exit;
}
```

Haetaan sen blogin tiedot jonne uusi artikkeli halutaan lisätä

```
$blog = getBlogByIdAndUserId($conn, $blog_id, $user_id);
function getBlogByIdAndUserId(mysqli $conn, int $blog_id, int $user_id):
?array {
    $sql = "SELECT blog_id, name, slug, description, published, visibility,
views,
                created_at, updated_at, layout_id, style_id
FROM BLOG
WHERE blog_id = ? AND user_id = ? AND deleted_at IS NULL
LIMIT 1";

    if ($stmt = $conn->prepare($sql)) {
        $stmt->bind_param("ii", $blog_id, $user_id);
        $stmt->execute();
        $result = $stmt->get_result();
        $blog = $result->fetch_assoc();
        $stmt->close();
        return $blog ?: null;
    } else {
        error_log("Tietokantavirhe getBlogByIdAndUserId(): " . $conn->error);
        return null;
    }
}
```

Tiedosto edit-article-modal.php

Blogin artikkelien haku

```
// 14.10.25 --> Mahdollistetaan useamman blogin käyttö per käyttäjä
$blog_id = isset($_GET['blog_id']) ? (int) $_GET['blog_id'] : null;

// 14.10.25 Lisätty $blog_id jotta voidaan käsitellä oikeaa blogia.
$stmt = $conn->prepare("SELECT a.article_id, a.title, a.content, a.image_path,
a.deleted_at
                        FROM ARTICLE a
                        JOIN BLOG b
                        ON a.blog_id = b.blog_id
                        WHERE b.user_id = ? AND b.blog_id = ? AND b.deleted_at
IS NULL AND a.deleted_at IS NULL"
                        );
$stmt->bind_param("ii", $user_id, $blog_id);
$stmt->execute();
$result = $stmt->get_result();
$articles = $result->fetch_all(MYSQLI_ASSOC);
$stmt->close();
?>
```

Muutokset vf-tiedostoissa

Tiedosto edit-blog-vf.php

Lisätään blog_id:

```
$blog_id = trim($_POST['blog_id'] ?? '');
```

Funktio updateBlog muutos

Lisätään blog_id funktioon jotta saadaan päivitettyä oikea blog:

```
echo updateBlog($conn, $user_id, $blog_id, $blog_name, $blog_description,
$slug);

function updateBlog($conn, $user_id, $blog_id, $blog_name, $blog_description,
$slug) {
    $query = "UPDATE BLOG SET name = ?, description = ?, slug = ?, updated_at
= NOW()
            WHERE user_id = ? AND blog_id = ? AND deleted_at IS NULL";
```

```

        if ($stmt = $conn->prepare($query)) {
            $stmt->bind_param("ssii", $blog_name, $blog_description, $slug,
$user_id, $blog_id);
            if ($stmt->execute()) {
                $stmt->close();
                return "Blogin tiedot päivitetty onnistuneesti! Omaa blogia pääset
katsomaan kun päivität sivun.";
            } else {
                $stmt->close();
                return "Virhe päivityksessä: " . $conn->error;
            }
        } else {
            return "Virhe tietokantakyselyssä.";
        }
    }
}

```

Tiedosto delete-blog-vf.php

Funktio deleteBlog muutos

```

//14.10.25 Lisätty $blog_id jotta poistetaan oikea blogi
echo deleteBlog($conn, $user_id, $blog_id);

function deleteBlog($conn, $user_id, $blog_id) {
    $sql ="UPDATE BLOG SET deleted_at = NOW() WHERE user_id = ? AND blog_id =
? AND deleted_at IS NULL";

    if ($stmt = $conn->prepare($sql)) {
        $stmt->bind_param("ii", $user_id, $blog_id);
        if ($stmt->execute()){
            $stmt->close();
            return "Blogi on merkitty poistetuksi. Se poistetaan lopullisesti
viikon kuluttua.";
        } else {
            $stmt->close();
            return "Virhe poistossa: " . $conn->error;
        }
    } else {
        return "Virhe tietokantakyselyssä.";
    }
}

```