

Auth.php

```
controllers > Auth.php
1  <?php
2  require_once __DIR__ . '/../models/User.php';
3
4  /**
5   * Klassen Auth håndterer autentisering (innlogging, registrering og utlogging).
6   */
7  class Auth {
8
9      /**
10       * Håndterer innlogging av brukere.
11       */
12       public static function handleLogin() {
13           session_start();
14           $email = $_POST['email'] ?? '';
15           $password = $_POST['password'] ?? '';
16
17           // Verifiserer at e-post og passord stemmer med en bruker i databasen
18           $user = User::verifyCredentials($email, $password);
19
20           if ($user) {
21               // Hvis brukeren finnes - sett sesjonsdata og send til forsiden
22               $_SESSION['user_id'] = $user['id'];
23               $_SESSION['user_name'] = $user['name'];
24
25               // Redirect tilbake til prosjektets index (stamme-nivå)
26               header('Location: ../index.php');
27               exit;
28           } else {
29               // Hvis feil - lagre feilmelding i sesjonen og send tilbake til login-siden
30               $_SESSION['auth_error'] = 'Ugyldig e-post eller passord.';
31               header('Location: ../view/login.php');
32               exit;
33           }
34       }
35   }
```

```
/**
 * Håndterer registrering av nye brukere.
 * Oppretter ny bruker og logger automatisk inn hvis vellykket. */
public static function handleRegister() {
    session_start();
    $name = trim($_POST['name'] ?? '');
    $email = trim($_POST['email'] ?? '');
    $password = $_POST['password'] ?? '';

    // Forsøker å registrere brukeren
    $ok = User::register($name, $email, $password);

    if ($ok) {
        // Hvis registrering lykkes, logg inn automatisk
        $user = User::verifyCredentials($email, $password);
        $_SESSION['user_id'] = $user['id'];
        $_SESSION['user_name'] = $user['name'];

        header('Location: ../index.php');
        exit;
    } else {
        // Hvis noe feiler (f.eks. e-post finnes allerede)
        $_SESSION['auth_error'] = 'Kunne ikke registrere brukeren. E-post kan allerede være i bruk.';
        header('Location: ../register.php');
        exit;
    }
}
```

```
/**
 * Logger ut brukeren.
 * Stopper sesjonen og sender brukeren tilbake til login-siden.
 */
public static function logout() {
    session_start();
    session_unset();
    session_destroy();

    header('Location: ../view/login.php');
    exit;
}

/**
 * Enkel "dispatcher" som gjør at denne filen kan kalles direkte fra et HTML-skjema
 */
if (php_sapi_name() !== 'cli') {
    $action = $_GET['action'] ?? '';

    if ($action === 'login' && $_SERVER['REQUEST_METHOD'] === 'POST') {
        Auth::handleLogin();
    }

    if ($action === 'register' && $_SERVER['REQUEST_METHOD'] === 'POST') {
        Auth::handleRegister();
    }

    if ($action === 'logout') {
        Auth::logout();
    }
}
?>
```

Conversations.php

```
<?php
require_once 'models/Database.php';

class Conversation
{
    /**
     * Lagre en samtale (brukerinput + botens svar)
     */
    public static function saveMessage($userInput, $botResponse)
    {
        // Enkel validering
        if (empty($userInput) || empty($botResponse)) {
            return false;
        }

        try {
            $db = Database::connect();

            $stmt = $db->prepare("
                INSERT INTO conversations (user_input, bot_response)
                VALUES (:user_input, :bot_response)
            ");

            $stmt->execute([
                ':user_input' => $userInput,
                ':bot_response' => $botResponse
            ]);
            return true;
        } catch (Exception $e) {
            return false;
        }
    }
}
```

```

/**
 * Hent alle tidligere samtaler (nyeste først)
 */
public static function getAllMessages()
{
    $db = Database::connect();

    $stmt = $db->query("
        SELECT user_input, bot_response, created_at
        FROM conversations
        ORDER BY created_at DESC
    ");

    return $stmt->fetchAll(PDO::FETCH_ASSOC);
}

/**
 * Tøm hele samtaleloggen
 */
public static function clearAll()
{
    $db = Database::connect();
    $db->exec("DELETE FROM conversations");
}
}

?>

```

Database.php

```

<?php
class Database {

    // Databasekonfigurasjon
    private static $host = 'localhost';
    private static $db   = 'chatbot';
    private static $user = 'root';
    private static $pass = '';
    private static $charset = 'utf8mb4';

    /**
     * Oppretter og returnerer en PDO-tilkobling til databasen.
     * Bruker try/catch for å håndtere eventuelle tilkoblingsfeil på en sikker måte.
     *
     * @return PDO Returnerer et PDO-objekt ved vellykket tilkobling.
     */
    public static function connect() {

        // Data Source Name (DSN) - spesifiserer database-driver, vert og navn
        $dsn = "mysql:host=" . self::$host . ";dbname=" . self::$db . ";charset=" . self::$charset;

        try {
            //for de som bruker annen port enn standard 3306
            #$pdo = new PDO($dsn, self::$user, self::$pass);
            //for de som bruker port 3307
            $pdo = new PDO("mysql:host=localhost;port=3307;dbname=chatbot", "root", "");

            // Setter feilhåndtering til å kaste unntak
            $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

            return $pdo;
        } catch (PDOException $e) {
            // Unngå å vise detaljerte feilmeldinger til brukere av sikkerhetsårsaker
            die('Database connection failed.');
```

ChatBot.php

```

controllers > ChatBot.php
1  <?php
2  //Henter nødvendige modeller
3  require_once __DIR__ . '/../models/GeoCoder.php';
4  require_once __DIR__ . '/../models/WeatherService.php';
5  require_once __DIR__ . '/../models/Conversation.php';
6
7
8  class ChatBot {
9      public function respond($input) {
10         $place = trim($input);
11         if (!$place) return "Skriv inn et sted for å få værddata.";
12
13         // Hent koordinater
14         $coords = GeoCoder::getCoordinates($place);
15         if (!$coords) return "Beklager, jeg fant ikke stedet '$place'.";
16
17         // Hent værddata
18         $weather = WeatherService::getWeather($coords['lat'], $coords['lon']);
19         if (!$weather) return "Kunne ikke hente værddata for $place.";
20
21         // Lag svaret
22         $response = "Været i $place nå: {$weather['temperature']}°C, vind: {$weather['wind']} m/s, fuktighet: {$weather['humidity']}%";
23
24         // Lagre samtalen i databasen
25         Conversation::saveMessage($place, $response);
26
27         // Returner svaret til visningen
28         return $response;
29     }
30 }
31
```

User.php

```
models > User.php
1  <?php
2  require_once __DIR__ . '/Database.php';
3
4  class User {
5      public static function findByEmail($email) {
6          // $pdo = Database::connect();
7          $stmt = $pdo->prepare('
8              SELECT *
9              FROM users
10             WHERE email = :email LIMIT 1');
11          $stmt->execute([':email' => $email]);
12          return $stmt->fetch(PDO::FETCH_ASSOC); // fetch_obj
13      }
14
15      public static function register($name, $email, $password) {
16          // Grunnleggende validering
17          if (empty($email) || empty($password) || empty($name)) return false;
18
19          // Ser om e-posten allerede er registrert
20          if (self::findByEmail($email)) return false;
21
22          $hash = password_hash($password, PASSWORD_DEFAULT);
23
24          // sett inn ny bruker i databasen
25          $stmt = $pdo->prepare('INSERT INTO users (name, email, password_hash, created_at) VALUES (:name, :email, :password_hash, NOW())');
26          return $stmt->execute([
27              ':name' => $name,
28              ':email' => $email,
29              ':password_hash' => $hash
30          ]);
31      }
32
33      public static function verifyCredentials($email, $password) {
34          $user = self::findByEmail($email);
35          if (!$user) return false;
36          if (password_verify($password, $user['password_hash'])) return $user;
37          return false;
38      }
39  }
40
```

Index.php

```

<?php
session_start();

// Hvis ikke innlogget: send til login-side
if (empty($_SESSION['user_id'])) {
    header('Location: login.php');
    exit;
}

require_once __DIR__ . '/controllers/ChatBot.php';
$bot = new ChatBot();

$input = $_GET['sted'] ?? '';
$response = $input ? $bot->respond($input) : "Hei ! Skriv inn et sted, så forteller jeg deg været der.";
$conversations = Conversation::getAllMessages();
?>

<!DOCTYPE html>
<html lang="no">
<head>
    <meta charset="UTF-8">
    <title>Værassistent</title>
    <link rel="stylesheet" href="public/css/style.css">
</head>
<body>
    <a href="views/logout.php" class="logout-btn">Logg ut</a>

    <div class="chat">
        <h1>Værassistent</h1>
        <div class="messages">

            <!-- Knapp for å vise/skjule tidligere samtaler -->
            <button id="showPrev" type="button" onclick="toggleMessages()">📄 Vis tidligere samtaler</button>

            <!-- Tidligere samtaler ligger skjult til knappen trykkes -->
            <div id="previous" style="display:none; margin-top: 10px;">
                <?php foreach ($conversations as $conv): ?>
                    <div class="msg user"><?= htmlspecialchars($conv['user_input']) ?></div>
                    <div class="msg bot"><?= htmlspecialchars($conv['bot_response']) ?></div>
                <?php endforeach; ?>
            </div>

```

```

<body>
  <a href="views/logout.php" class="logout-btn">Logg ut</a>

  <div class="chat">
    <h1>Værassistent</h1>
    <div class="messages">

      <!-- Knapp for å vise/skjule tidligere samtaler -->
      <button id="showPrev" type="button" onclick="toggleMessages()"> Vis tidligere samtaler</button>

      <!-- Tidligere samtaler ligger skjult til knappen trykkes -->
      <div id="previous" style="display:none; margin-top: 10px;">
        <?php foreach ($conversations as $conv): ?>
          <div class="msg user"><?= htmlspecialchars($conv['user_input']) ?></div>
          <div class="msg bot"><?= htmlspecialchars($conv['bot_response']) ?></div>
        <?php endforeach; ?>
      </div>

      <!-- Nåværende melding -->
      <?php if ($input): ?>
        <div class="msg user"><?= htmlspecialchars($input) ?></div>
        <div class="msg bot"><?= htmlspecialchars($response) ?></div>
      <?php else: ?>
        <div class="msg bot"><?= htmlspecialchars($response) ?></div>
      <?php endif; ?>
    </div>

    <form method="get">
      <input type="text" name="sted" placeholder="Skriv inn sted..." autofocus>
      <button>Send</button>
    </form>
  </div>
  <script src="public/js/script.js"></script>
</body>
</html>

```