

Program zajęć i zadania dla technika informatyka

| | |
|---|---|
| Zajęcia/ zadanie 1: Wprowadzenie do organizacji i zasad pracy | |
| Opis: | Zapoznanie z organizacją przyjmującą, zasadami funkcjonowania i kulturą pracy. Omówienie zasad BHP, ochrony danych i bezpieczeństwa informacji w miejscu praktyk. Poznanie opiekuna/mentora oraz ustalenie sposobu komunikacji i raportowania postępów. Przygotowanie stanowiska pracy oraz konfiguracja dostępu do narzędzi i zasobów. Omówienie celu praktyk i sposobu realizacji zadań w całym okresie mobilności. |
| Zajęcia / zadanie 2: Konfiguracja środowiska i uruchomienie projektu React.js | |
| Opis: | Konfiguracja środowiska programistycznego do pracy nad aplikacją webową. Utworzenie i uruchomienie projektu w React.js oraz wstępne uporządkowanie struktury. Przygotowanie repozytorium projektu i ustalenie podstawowych zasad pracy z kodem. Wstępne testy działania aplikacji i weryfikacja poprawności konfiguracji. |
| Zajęcia / zadanie 3: React.js – komponenty i struktura aplikacji | |
| Opis: | Tworzenie komponentów w React.js oraz praca z ich podstawowymi właściwościami. Organizacja struktury aplikacji, porządkowanie folderów i podział na moduły. Budowa prostych widoków i łączenie ich w spójną strukturę aplikacji. Utrwalanie dobrych praktyk czytelnego kodu i pracy w projekcie. |
| Zajęcia / zadanie 4: React.js – formularze i elementy interakcji | |
| Opis: | Tworzenie formularzy i obsługa danych wprowadzanych przez użytkownika. Implementacja podstawowych elementów interakcji w interfejsie aplikacji. Walidacja prostych danych wejściowych i testowanie działania formularzy. Dopasowanie formularzy do potrzeb aplikacji oraz weryfikacja poprawności działania. |
| Zajęcia / zadanie 5: React.js – dopracowanie funkcjonalności i przygotowanie do pracy z danymi | |
| Opis: | Dopracowanie logiki aplikacji i porządkowanie kodu źródłowego. Przygotowanie elementów aplikacji pod integrację z warstwą danych. Testowanie działania komponentów i usuwanie błędów w podstawowych funkcjach. Zwiększenie spójności działania aplikacji przed etapem pracy z bazą danych. |

| | |
|---|--|
| Zajęcia / zadanie 6: Projektowanie struktury bazy danych MongoDB | |
| Opis: | Posługiwanie się podstawowymi pojęciami dotyczącymi baz danych NoSQL. Projektowanie struktury danych w MongoDB obejmującej kolekcje, dokumenty i pola. Dobór struktury danych do wymagań aplikacji webowej. Przygotowanie założeń do późniejszej obsługi operacji na danych. |

| | |
|---|---|
| Zajęcia / zadanie 7: MongoDB – tworzenie kolekcji i przygotowanie danych | |
| Opis: | Tworzenie kolekcji oraz przygotowanie przykładowych danych testowych. Weryfikacja spójności zaprojektowanej struktury danych w praktyce. Porządkowanie danych i przygotowanie ich do wykorzystania w aplikacji. Testowanie podstawowych operacji na danych w środowisku roboczym. |

| | |
|---|--|
| Zajęcia / zadanie 8: MongoDB – operacje CRUD | |
| Opis: | Wykonywanie podstawowych operacji CRUD w MongoDB. Dodawanie, odczyt, aktualizacja i usuwanie danych w kolekcjach i dokumentach. Testowanie poprawności operacji na danych w typowych scenariuszach aplikacji. Utrwalanie zasad poprawnej pracy na danych i kontroli zmian. |

| | |
|---|--|
| Zajęcia / zadanie 9: MongoDB – utrwalanie operacji CRUD i porządkowanie danych | |
| Opis: | Doskonalenie operacji CRUD na danych w kontekście potrzeb aplikacji. Uspójnianie i porządkowanie struktury danych oraz wprowadzanie korekt projektu bazy. Testowanie scenariuszy użycia związanych z pracą na danych. Przygotowanie bazy danych do wykorzystania w dalszych etapach pracy projektowej. |

| | |
|---|--|
| Zajęcia / zadanie 10: Bootstrap – układ strony i podstawowe komponenty UI. | |
| Opis: | Projektowanie układu aplikacji oraz budowanie responsywnego layoutu w Bootstrap. Stosowanie podstawowych komponentów Bootstrap w interfejsie użytkownika. Przygotowanie spójnej struktury widoków z zachowaniem zasad czytelności. Testowanie podstawowego działania interfejsu w różnych rozdzielczościach. |

| | |
|---|---|
| Zajęcia / zadanie 11: Bootstrap – formularze i nawigacja | |
| Opis: | Implementacja elementów nawigacyjnych z wykorzystaniem Bootstrap. Tworzenie i stylowanie formularzy oraz podstawowych elementów UI. Dbanie o spójność wizualną i ergonomię interfejsu użytkownika. Testowanie działania komponentów interfejsu w aplikacji. |

| | |
|---|--|
| Zajęcia / zadanie 12: Bootstrap – dopracowanie responsywności i spójności interfejsu | |
|---|--|

| | |
|--|---|
| Opis: | Testowanie responsywności UI i wprowadzanie korekt układu. Porządkowanie i ujednolicanie elementów wizualnych aplikacji. Dopracowanie szczegółów interfejsu w celu poprawy czytelności. Weryfikacja spójności UI w przygotowywanym rozwiązaniu. |
| Zajęcia / zadanie 13: Podstawowe zabezpieczenia aplikacji webowej – wdrożenie | |
| Opis: | Wdrażanie podstawowych ustawień bezpieczeństwa w aplikacji webowej. Stosowanie dobrych praktyk ograniczających typowe ryzyka w aplikacjach webowych. Poprawna konfiguracja kluczowych elementów po stronie aplikacji i środowiska. Wstępna weryfikacja bezpieczeństwa na poziomie podstawowych scenariuszy. |
| Zajęcia / zadanie 14: Podstawowe zabezpieczenia – weryfikacja i korekty konfiguracji. | |
| Opis: | Testowanie działania wprowadzonych zabezpieczeń w praktyce. Korekta ustawień i konfiguracji w celu usunięcia wykrytych nieprawidłowości. Utrwalanie dobrych praktyk bezpiecznego przygotowania aplikacji do wdrożenia. Przygotowanie aplikacji do uruchomienia w środowisku produkcyjnym. |
| Zajęcia / zadanie 15: Wdrażanie aplikacji na serwer z wykorzystaniem AWS – uruchomienie i testy | |
| Opis: | Przygotowanie aplikacji do uruchomienia w środowisku produkcyjnym. Wdrożenie aplikacji na serwer z wykorzystaniem podstawowych usług AWS. Konfiguracja niezbędnych zasobów i podstawowe ustawienia bezpieczeństwa w chmurze. Weryfikacja poprawności działania wdrożonego rozwiązania oraz testy końcowe. Podstawy utrzymania wdrożenia i kontrola działania aplikacji po uruchomieniu. |

Program zajęć i zadania dla technika programisty

| | |
|--|---|
| Zajęcia/ zadanie 1: Wprowadzenie do organizacji i zasad pracy | |
| Opis: | Zapoznanie z organizacją przyjmującą, zasadami funkcjonowania i kulturą pracy. Omówienie zasad BHP, ochrony danych i bezpieczeństwa informacji w miejscu praktyk. Poznanie opiekuna/mentora oraz ustalenie sposobu komunikacji i raportowania postępów. Przygotowanie stanowiska pracy oraz konfiguracja dostępu do narzędzi i zasobów. Omówienie celu praktyk i sposobu realizacji zadań w całym okresie mobilności. |
| Zajęcia / zadanie 2: Konfiguracja środowiska i uruchomienie projektu React Native | |
| Opis: | Konfiguracja środowiska programistycznego do pracy nad aplikacją mobilną. Utworzenie i uruchomienie projektu w React Native oraz wstępne uporządkowanie struktury. Przygotowanie repozytorium projektu i ustalenie podstawowych zasad pracy z kodem. Wstępne testy działania aplikacji i weryfikacja poprawności konfiguracji. |
| Zajęcia / zadanie 3: React Native – komponenty i struktura aplikacji mobilnej | |
| Opis: | Tworzenie komponentów w React Native oraz praca z ich podstawowymi właściwościami. Organizacja struktury aplikacji, porządkowanie folderów i podział na moduły. Budowa prostych ekranów i łączenie ich w spójną strukturę aplikacji. Utrwalanie dobrych praktyk czytelnego kodu i pracy w projekcie. |
| Zajęcia / zadanie 4: React Native – formularze i obsługa danych użytkownika | |
| Opis: | Tworzenie formularzy i obsługa danych wprowadzanych przez użytkownika. Implementacja podstawowych elementów interakcji w aplikacji mobilnej. Walidacja prostych danych wejściowych i testowanie działania formularzy. Dopasowanie formularzy do potrzeb aplikacji oraz weryfikacja poprawności działania. |
| Zajęcia / zadanie 5: React Native – nawigacja i dopracowanie funkcjonalności | |
| Opis: | Implementacja podstawowej nawigacji pomiędzy ekranami aplikacji mobilnej. Dopracowanie logiki aplikacji i porządkowanie kodu źródłowego. Testowanie działania komponentów i usuwanie błędów w podstawowych funkcjach. Zwiększenie spójności działania aplikacji przed etapem testowania i wdrożenia. |
| Zajęcia / zadanie 6: Git i GitHub – podstawy pracy zespołowej w projekcie | |

| | |
|-------|--|
| Opis: | Omówienie zasad pracy zespołowej z wykorzystaniem systemu kontroli wersji Git i platformy GitHub. Przygotowanie repozytorium i ustalenie zasad wprowadzania zmian. Wykonywanie podstawowych operacji w Git (commit, branch, merge). Organizacja pracy nad zadaniami w zespole z wykorzystaniem GitHub. |
|-------|--|

| | |
|---|--|
| Zajęcia / zadanie 7: Git i GitHub – współdzielenie kodu i rozwiązywanie konfliktów | |
| Opis: | Praca na gałęziach i przygotowywanie zmian do integracji. Wymiana zmian w zespole oraz współpraca nad wspólnymi elementami projektu. Rozwiązywanie podstawowych konfliktów przy scalaniu kodu. Utrwalanie zasad czytelnych commitów i porządku w repozytorium. |

| | |
|--|---|
| Zajęcia / zadanie 8: GitHub - przegląd zmian i organizacja pracy (code review w praktyce) | |
| Opis: | Przygotowanie zmian do weryfikacji zespołowej. Udział w przeglądzie kodu i omawianie wprowadzonych rozwiązań. Wprowadzanie korekt na podstawie informacji zwrotnej. Utrwalanie zasad pracy zespołowej i odpowiedzialności za jakość kodu. |

| | |
|--|---|
| Zajęcia / zadanie 9: Testy jednostkowe w Jest – przygotowanie środowiska i pierwsze testy | |
| Opis: | Zapoznanie z ideą testów jednostkowych i rolą testowania w procesie wytwarzania oprogramowania. Konfiguracja podstawowego środowiska testowego w Jest. Tworzenie prostych testów jednostkowych dla wybranych elementów projektu. Uruchamianie testów i interpretacja wyników. |

| | |
|---|---|
| Zajęcia / zadanie 10: Jest – rozwijanie testów i analiza wyników | |
| Opis: | Rozwijanie zestawu testów jednostkowych dla kolejnych elementów aplikacji. Analiza błędów wykrytych przez testy oraz wprowadzanie poprawek w kodzie. Utrwalanie dobrych praktyk pisania czytelnych testów. Weryfikacja stabilności aplikacji poprzez regularne uruchamianie testów. |

| | |
|--|---|
| Zajęcia / zadanie 11: Dobre praktyki programistyczne – clean code w projekcie | |
| Opis: | Stosowanie podstawowych zasad clean code w bieżącej pracy nad aplikacją. Porządkowanie kodu, poprawa czytelności i spójności nazewnictwa. Eliminowanie powtórzeń i usprawnianie struktury modułów. Przygotowanie kodu do przeglądu i dalszego rozwijania w zespole. |

| | |
|---|---|
| Zajęcia / zadanie 12: Code review i współpraca – doskonalenie jakości rozwiązania interfejsu | |
| Opis: | Udział w code review i omawianie proponowanych rozwiązań w zespole. Uzasadnianie zmian oraz przyjmowanie informacji |

| | |
|--|--|
| | zwrotnej. Wprowadzanie poprawek po przeglądzie kodu. Utrwalanie standardów jakości i komunikacji w pracy programistycznej. |
| Zajęcia / zadanie 13: CI/CD – wprowadzenie i przygotowanie podstawowego procesu | |
| Opis: | Zapoznanie z ideą automatyzacji budowania i wdrażania aplikacji. Przygotowanie podstawowego procesu CI/CD dla projektu, obejmującego automatyczne kroki build i testy. Konfiguracja podstawowych ustawień procesu wdrożeniowego. Weryfikacja poprawności działania automatyzacji na przykładach. |
| Zajęcia / zadanie 14: CI/CD – uruchomienie procesu, testy i korekty | |
| Opis: | Uruchamianie procesu CI/CD w praktyce i analiza wyników etapów automatycznych. Korekty konfiguracji w przypadku błędów w budowaniu lub testach. Utrwalanie zasad publikowania kolejnych wersji aplikacji. Przygotowanie rozwiązania do finalnego wdrożenia. |
| Zajęcia / zadanie 15: CI/CD – wdrożenie wersji aplikacji i podsumowanie | |
| Opis: | Wdrożenie wersji aplikacji zgodnie z przygotowanym procesem CI/CD. Weryfikacja poprawności działania opublikowanej wersji oraz sprawdzenie rezultatów końcowych. Omówienie przebiegu prac i osiągniętych efektów uczenia się z mentorem. Podsumowanie realizacji praktyk i rekomendacje do dalszego rozwoju. |