

```

margin: 0; padding: 0;
}

#chat-text p span.person {
  color: #636cc8;
}

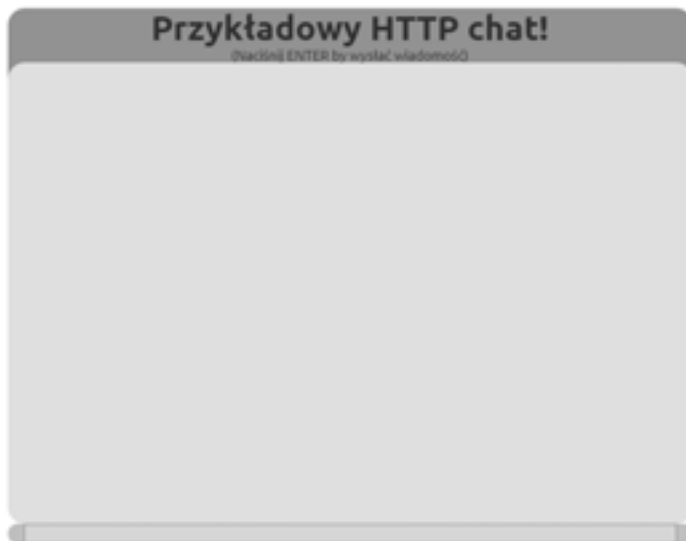
#chat-text p span.text {
  color: blue;
}

#chat-input-window {
  background-color: #c1b8ff;
  text-align: center;
  border-radius: 1em;
}

#chat-input {
  width: 95%;
  background-color: #d3d4ff;
  border: 1px solid #8d88ff;
}

```

Uzyskany wygląd strony, bądź co bądź bardzo minimalistyczny, został pokazany na rysunku 10.



**Rysunek 10.** Wygląd strony czatu wg definicji w pliku HTML i CSS

Jeśli chodzi o skrypt obsługujący interfejs użytkownika, ma on dwie główne funkcje:

- **Obsługa tekstowego pola wiadomości** – jeśli użytkownik naciśnie klawisz ENTER, mając zaznaczone pole wiadomości, pobierz wiadomość z pola, wyczyść pole tekstowe, a następnie w tle<sup>290</sup> prześlij do serwera żądanie z wiadomością, wskazując zasób /chat jako adresata.
- **Aktywne odpytywanie serwera o nowe wiadomości** – co 1000 ms wyślij w tle żądanie z identyfikatorem najnowszej znanej wiadomości do serwera, wskazując zasób /messages jako adresata. Wszystkie otrzymane wiadomości wyświetl w oknie z wiadomościami.

Opis samej mechaniki operowania na drzewie DOM (*Document Object Model*), czyli strukturze strony WWW, wykracza poza zakres niniejszej książki. Niemniej jednak w tym przypadku kod dodający nowe wiadomości oraz podpinający obsługę zdarzeń (naciśnięcia klawiszy) pod pole tekstowe jest bardzo prosty, więc jego analiza nie powinna sprawić trudności.

Sam kod przedstawia się następująco:

#### httpchat\_main.js:

```
// Funkcja do uruchomienia po załadowaniu strony.
$(function() {
  // Podepnij funkcję wysyłającą tekst do chat-input i ustaw focus na
  // pole
  // tekstowe.
  $('#chat-input')
    .focus()
    .keypress(function(ev) {
      if (ev.which !== 13) { // Klawisz ENTER.
        return;
      }
      ev.preventDefault();

      // Pobierz zawartość i wyczyść pole tekstowe.
      var text = $(this).val();
      $(this).val('');

      // Wyślij tekst do serwera.
      var text_json = JSON.stringify({
        "text": text
      });
      $.ajax({
```

---

<sup>290</sup> Popularnie metoda ta nazywana jest AJAX (*Asynchronous JavaScript and XML*), choć obecnie format XML jest wykorzystywany rzadziej niż dużo prostszy JSON (metoda serializacji przesyłanych danych jest oczywiście dowolna i zależy jedynie od wyboru programisty).