1. PHP

```php
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    if (!isset($_POST["word"]) || !preg_match("/^[a-zA-Z]+$/", $_POST["word"])) {
        header("HTTP/1.1 400 Invalid Request");
        die();
    }

    $badword = strtolower($_POST["word"]);
    $lines = array();
    foreach (file("text.txt", FILE_IGNORE_NEW_LINES) as $line) {
        $badline = FALSE;
        $words = explode(" ", $line);
        foreach ($words as $word) {
            if (strtolower($word) == strtolower($badword)) {
                $badline = TRUE;
            }
        }
        if (!$badline) {
            $lines[] = $line;
        }
    }
    file_put_contents("text.txt", implode("\n", $lines));
}
?>

<form action="" method="post">
    <div>Word to remove:
        <input type="text" name="word" />
        <input type="submit" value="Submit" />
    </div>
</form>

<p>Current file text:</p>
<pre><?= file_get_contents("text.txt") ?></pre>
```

2. JavaScript / JSON

```javascript
window.onload = function() {
    fetchWord();
};

function fetchWord() {
    var part = document.getElementById("part").value;
    var ajax = new XMLHttpRequest();
    ajax.onload = wordHasArrived;
    ajax.open("GET", "word.php?part=" + part, true);
    ajax.send();
}

function wordHasArrived() {
    var json = JSON.parse(this.responseText);
    document.getElementById("word").innerHTML = json.word + " (" + json.part + ")";
    document.getElementById("choices").innerHTML = "";
    for (var i = 0; i < json.choices.length; i++) {
        var button = document.createElement("button");
        button.innerHTML = json.choices[i].definition;
        button.className = json.choices[i].correct ? "correct" : "incorrect";
        button.onclick = showResult;
        button.style.display = "block";
        document.getElementById("choices").appendChild(button);
    }
}

function showResult() {
    alert("You are " + this.className);
    fetchWord();
}
```

3. Regular Expressions

a) `/^([0-9]{3}-?)?[0-9]{3}-?[0-9]{4}$/`

b) `^[A-Z]{3} ?[0-9]{3}$`

c) `/^[a-zA-Z][a-zA-Z0-9-_]{5,17}$/`

4. SQL

```
-- all actors who were in a Romantic Comedy with Woody Allen
-- that came out in 1999 or later

SELECT a2.first_name, a2.last_name, m.name
FROM actors a1
JOIN roles r1 ON r1.actor_id = a1.id
JOIN movies m ON m.id = r1.movie_id
JOIN roles r2 ON r2.movie_id = m.id
JOIN actors a2 ON a2.id = r2.actor_id
JOIN movies_genres mg1 ON mg1.movie_id = m.id
JOIN movies_genres mg2 ON mg2.movie_id = m.id
WHERE a1.first_name = "Woody"
 AND a1.last_name = "Allen"
 AND a1.id != a2.id
 AND mg1.genre = "Romance"
 AND mg2.genre = "Comedy"
 AND m.year >= 1999
ORDER BY m.name, a2.last_name, a2.first_name;
```