

## HTML5 and JavaScript: Project Specification

This project we submit will be a Contacts Book application, which will have the following characteristics:

1. Add contact details:
  - First and Last name.
  - Sex.
  - Birthday.
  - Marital Status.
  - Personal and Work email.
  - Mobile and Work number.
  - Address, City and Post code.
  - Organisation.
  - Groups.
  - Contacts Picture.
  - Space for additional notes.
2. Operations through the graphical user-interface:
  - Being able to edit contact details.
  - Removing any contact details.
  - Clicking an email address will make a draft email will pop up.
  - Being able to sort contacts alphabetically and by groups.
3. Visible reminders
  - If a birthday is within a week, a reminder will show up below it.

### User-Interface Prototype

The image shows a web browser window with the address bar displaying 'localhost:63342/Contacts\_Assessment1/Contacts.html'. The main content is a blue-themed form titled 'Add New Contact'. The form contains the following fields and controls:

- First Name: Text input with placeholder 'Insert First name'.
- Last Name: Text input with placeholder 'Insert Last name'.
- Sex: Radio buttons for Male, Female, and Other.
- Birthday: Text input with placeholder 'dd/mm/yyyy'.
- Marital Status: Dropdown menu with 'Single' selected.
- Personal Email: Text input with placeholder 'Insert a valid email address'.
- Work Email: Text input with placeholder 'Insert a valid email address'.
- Mobile Number: Text input with placeholder 'Insert phone number'.
- Work Number: Text input with placeholder 'Insert phone number'.
- Address: Text input with placeholder 'Insert home address'.
- City: Text input with placeholder 'Insert City'.
- Postcode: Text input with placeholder 'Insert postcode'.
- Organisation: Text input with placeholder 'Insert organisation'.
- Group: Checkboxes for Family, Friends, Work, and Other.
- Contact Picture: A 'Choose file' button, a 'No file chosen' status, and a 'Submit' button.
- Notes: A large text area with placeholder 'Notes'.

At the bottom of the form, there is a silhouette icon for a contact picture, a 'Reset' button, and a 'Save' button.

## Design Documentation

### Contacts Type

- Name - string fields containing contact's name.
- Sex – a string field containing contact's gender.
- Birthday - a date field containing day, month and year.
- Marital Status - a string field containing contact's relationship status.
- Personal email - a string field containing contact's personal email address.
- Work email - a string field containing contact's work email address.
- Mobile number - a string field containing contact's mobile number.
- Work number – a string field containing contact's work number.
- Address – a string field containing contact's address.
- City – a string field containing contact's city.
- Post code – a string field containing contact's post code.
- Organisation – a string field containing contact's organisation.
- Groups - an array containing contact's groups.
- Notes - a string field containing contact's additional notes.
- Photo section - a storage section containing contact's photo.

### Methods available to Contacts instances are:

- getDate(): returns the date portion of a contact's date of birth.
- weekDue(): returns a string reminder when a contact's date of birth is a week away.

### Application Data and Functions

- groups[]: this will sort out the contacts into the groups the user has put them into.
- nameField, sexField, birthday Field, maritalField, pemailField, wemailField, mnumberField, wnumberField, addressField, cityField, postField and notesField,
- add(): add a new contact field.
- remove(): remove a contact field.
- edit(): edit a contact field.

### Project Aims

We expect to implement these features in the final application. In addition to these features, we will attempt to include the following features:

In “Add contact details”:

- Meetings.

In “Visible reminders”:

- An automatic template will be sent out wishing person a happy birthday.

In “Methods available to Contacts instances”:

- sendBirthday(): sends out a birthday wish and returns a boolean to show it's been sent.

In “Application Data and Functions”:

- getMeeting(): returns the date of the meeting.
- reminderMeeting(): if getMeeting() is today's date, return a boolean reminder.
- showLocation(): function which displays the location of the contact's address.
- exportContacts(): function which can export contact(s) either via email or notepad.
- addPicture(): function which allows the user to upload a contact picture.