# JobsAustralia.tech Final Schedule Plan

## Week 1

- Interviewed project supervisor.
- Confirmed requirements with project supervisor in regards to what was expected from the job matching website.
- All team members in the group listed their relevant skillsets.
- Team member roles were assigned.

## Week 2

- Researched technology, framework and cloud server to use.
- Discussed ideas on how to best implement the job match site including website design and features.

## Week 3

- Finalised technology, framework and cloud server to use. Our group decided to use Laravel for the development of the project. Brought a domain name and started uploading our test server to it.
- All team members in the group started setting up their development environments for this project.

## Week 4 (Sprint 1)

- All team members in the group set up their development environment.
- Delete account function was written.
- Create profile as an employer function was written.
- Wireframes for UI were designed and uploaded to the Google drive.
- Deployment to cloud service/server.
- Brought domain and configured DNS to host our website.
- Register an employee account function was written.
- Register an employer account function was written.
- Login function was written.
- Logout function was written.
- Testing was completed on the functions that were developed in this sprint.
- Tested application on the cloud/DNS host to ensure everything was set up/working correctly.
- Demonstrated end of sprint product to supervisor.

## Week 5 (Sprint 2)

- Create profile an employee function was written.
- Set up databases for the project.
- Wrote list of IT skills so we can determine what skills to list when registering or posting a job.
- Homepage for job seekers was created.
- Homepage for employers was created.
- Job posting in Laravel. Setting up required functions, etc. Was completed locally.
- Basic CSS for site.
- Edit Job seeker profile function was written.
- Tested functions developed in this sprint.
- Tested homepage compatibility.
- Demonstrated end of sprint product to supervisor.

## Week 6 (Sprint 3)

- View jobs as an employee (early works).
- Post job as an employer - HTML input form.
- Post job as an employer - PHP processing in Laravel.
- Post job as an employer - database engineering.
- HTML improvements for additional views.
- View and edit job seeker/employee profile function was written.
- Edit/view profile as an employer function was written.
- Testing of the functions that were developed in this sprint.
- Demonstrated end of sprint product to supervisor.

## Week 7 (Sprint 4)

- Job matchmaking algorithm for job seekers/employees.
- View posted job function was written.
- Job match filters.
- Job match (front end).
- Contact form for both sites was created.
- Support page for job seekers was created.
- Support page for employers was created.
- Edit job as an employer function was written.
- Testing of functions developed in this sprint.
- Demonstrated end of sprint product to supervisor.

## Week 8 (Sprint 5)

- Various APIs for applicant matchmaking on the employer side.
- Employer can view applicants for a job.
- Applicant matching on employer side.
- Improvements to matchmaking algorithm.
- Changed salary expression.
- Delete job as an employer function was written.
- Model security.
- Employees can submit/update resumes function was written.
- Apply for job as an employee function was written.
- Added location to employer.
- Testing of functions developed in this sprint.
- Demonstrated end of sprint product to supervisor.

## Week 9 (Sprint 6)

- Github integration (auto-fill skills on registration) was implemented.
- Github integration (reporting relevant repositories to Employers on Application).
- Logo design.
- Alphabetise and organise skills in views.
- Delete application as a job seeker function was written.
- Validation logic for salary (client side).
- Employer can view applicants resume function was written.
- Various bug fixes/improvements.
- Validation logic for salary (server side).
- Application page for employer was created.
- Improved the visibility of elements on the site for a non-white background.
- Reject application as an Employer (client side) function was written.
- Removed inline JavaScript.
- Auto fill location by IP address on registration was implemented.
- Demonstrated end of sprint product to supervisor.

## Week 10 (Sprint 7)

- Added additional fields, and ranking to matchmaking algorithm.
- Increased website performance (using Google Page Speed).
- Added additional skills (server and client side)
- Website design/GUI was updated.
- Notification email settings were implemented.
- Configured Laravel for outgoing emails (with Mailgun).
- Added filters for hours, term, and salary to job match page.
- Removed inline CSS.
- Job expiration function was written.
- Demonstrated end of sprint product to supervisor.

## Week 11 (Sprint 8)

- Cloud server and general security tests were conducted.
- Reject application as an employer (server side).
- Display expired notice on expired jobs.
- Notification emails were updated.
- Added ability for employer to converse via email with the job seeker who has applied for a job.
- Refactoring.
- Bug fixes.
- Documentation – Learning outcomes, user stories.

## Week 12

- Documentation – Final schedule plan, peer reviews, meeting minutes, technical solutions design, project charter, development guide, user manuals.
- End of project demo.